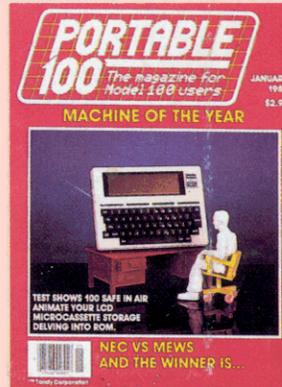
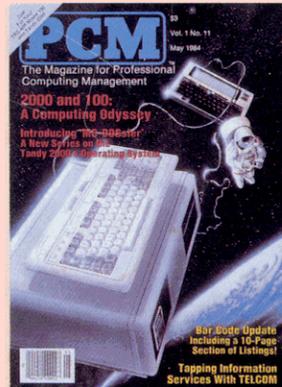
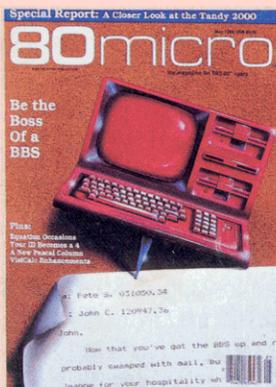
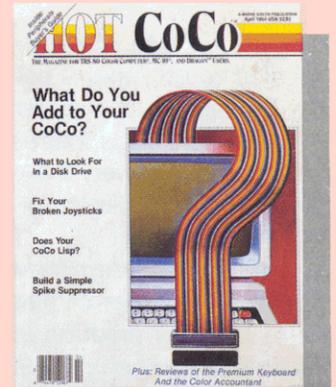
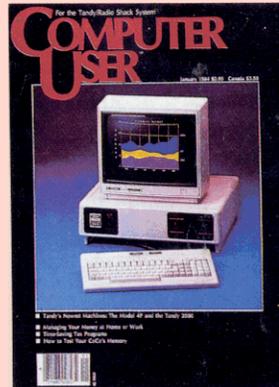
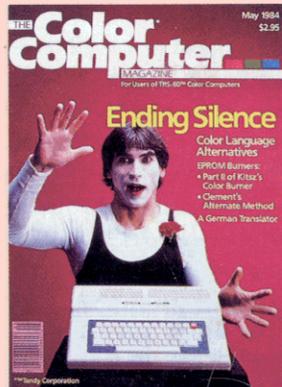


# Microcomputer News

# LAST ISSUE!



# COMPLETE YOUR SUBSCRIPTION WITH ANY ONE OF THESE GREAT TRS-80 ORIENTED MAGAZINES

SEE PAGE 2 FOR DETAILS

---

# THIS IS THE LAST ISSUE

## OF THE TRS-80 MICROCOMPUTER NEWS

***Here's how to complete your subscription with any one of the TRS-80 oriented computer magazines listed below.***

We regret to inform you that publication of the *TRS-80 Microcomputer News* is being discontinued with this issue.

Our reason for discontinuing publication is that there are currently eight magazines devoted exclusively to TRS-80 products, and doing a very fine job of covering the line. Therefore, continuing the publication of the *TRS-80 Microcomputer News* became redundant.

We are happy to recommend the magazines listed below. Forthcoming issues of these magazines will keep you informed of new products, practical applications and useful information about our TRS-80 line of computers and software. There will be no additional cost to you.

---

**1. ADVANCED COMPUTING** (formerly Two-Sixteen) is beginning its third year as an authoritative source of information for people who use TRS-80 Models II, 12, and 16 microcomputers in business, professional and scientific pursuits. *Advanced Computing* serves the particular needs of these specialized users. Back issues are available.

---

**2. COLOR COMPUTER** magazine tests and reviews all products of interest to color computerists, answers questions, provides support, features informative articles by recognized authors on home management, education and utility programs, exciting games and challenging hardware projects—a recognized authority.

---

**3. COMPUTER USER** is an all-in-one source for users of all models of the Radio Shack microcomputer line. Also included with your subscription will be access to P.O.D.E., our PROGRAM ON-LINE DATE EXCHANGE, FREE. "PODE" allows you the benefits of downloading program listings, electronic mail services, various data bases and much more.

---

**4. HOT COCO** is for color computer users. If you are just starting out, it will show you the first steps to learning how to make your color computer do what you want it to do, teach you BASIC programming, tell you what the best hardware and software buys are, provide useful programs for the home or office, and keep you entertained with games.

Next to this page is a postage-paid business reply card. Peel off the address label from this issue of the *TRS-80 Microcomputer News* and apply to label space on card. The upper right hand portion of the label will show how many issues you have left on your subscription. Check the box next to the magazine you want to receive (check only one box) to have your *TRS-80 Microcomputer News* subscription transferred (or extended if you already have a subscription); or if you paid for your subscription by check/money order and prefer a refund, please check that box. We must receive your choice by July 31, 1984. So, mail your card right away!

---

**5. 80 MICRO** is aimed at the user who wants to get more out of his machine. It includes tutorials on beginning and intermediate BASIC and Assembly programming, tips on making better use of commercial software, software and hardware reviews, buyer's guides, news, new product information and 12 to 20 programs each month the user can type in.

---

**6. PORTABLE 100** The Magazine for Model 100 Users, is the largest monthly magazine dedicated exclusively to the Model 100. It serves as the forum for applications and news releases in support of the 100. Additionally, *Portable 100* is the Systems Operator for the Model 100 SIG (Special Interest Group) on CompuServe.

---

**7. PCM** The Magazine for Professional Computing Management. For users of both the Model 100 and the Tandy 2000, combines articles with just-type-in-and-run programs, commentary, tutorials, hardware and software reviews, and games. A monthly feature is BASIC in bar code—ready to read into the M100's memory with a sweep of a bar code wand!

---

**8. THE RAINBOW** contains in each issue more than two dozen program listings, dozens of articles by nationally recognized authors, 20 departmental features by the leading color computer authorities and over 30 product reviews. *The Rainbow* has everything from games to business applications, home help, educational programs, and music.

# TRS-80<sup>®</sup> Microcomputer News

**TRS-80 Microcomputer News**  
Volume 6 Issue 6

JUNE, 1984

Editor Bruce Elliott  
Associate Editor Linda Miller  
Writers Carol Morton  
Maggie Sherrod-Stone  
Production Manager William Palmer  
Cover Art Director Barry King  
Art Production Richard Conner  
Production Clerks Diana Fair  
JoAnne White

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. Copyright 1984 by Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. All rights reserved.

Reproduction or use, without express written permission from Tandy Corporation, of any portion of TRS-80 Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than Ivan Sygoda's Profile article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

Single copies of TRS-80 Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 each, suggested retail price.

Back issues of Microcomputer News are available as Reprint sets.

Reprints	Cat. No.	Suggested Retail
Prior to Jan. 1981	26-2115	\$ 4.95
1981 Reprints	26-2240	9.95
1982 Reprints	26-2241	12.95

Retail prices shown in TRS-80 Microcomputer News may vary at individual stores and dealers. Reprints of other issues are not available.

© 1984 Tandy Corporation, Fort Worth, Texas 76102. The Company cannot be liable for pictorial or typographical errors. VisiCalc/Registered TM VisiCorp. Time Manager, Project Manager, Personnel Manager/TM Image Producers. ReformaTer/TM Microtech Exports. WESTLAW/TM West Publishing Co. DEC/Registered TM Digital Equipment Corp. IBM/Registered TM International Business Machines Corp. Apple/Registered TM Apple Computers. CompuServe/Registered TM CompuServe, Inc. Dow Jones News Retrieval/Registered TM Dow Jones & Co., Inc. Monty/TM Ritam Corp. ZORK/TM Infocom, Inc. Gammon Gambier/TM Dream Machine Collective. Scrabble/Registered TM Selchow & Richter Co. Monopoly/Registered TM Parker Brothers, Inc. Desktop/Plan/TM Marton, Inc. Electronic Broker/TM Electronic Broker, Inc. TeleVideo/Registered TM TeleVideo Systems, Inc. LDOS/TM Logical Systems, Inc. STOCKPAC/TM Standard & Poor's, Inc. PET/Registered TM Commodore Computers. AgrStar/TM AgriData Resources for its information and computing network. Poltergeist/© 1982 Metro-Goldwyn-Mayer Film Co. and SLM Entertainment LTD. Computer Discovery/TM Science Research Associates. XENIX and Multiplan/TM Microsoft Corp. UNIX/TM Bell Laboratories. CHAMPS/TM J.R. Dewey and M.L. Roush. Arcnet/TM Datapoint. CP/M, CP/M Plus, CBASIC and MP/M II/TM Digital Research. Z-80 Registered TM Zilog Corp. PFS. File and PFS. Report/TM Software Publishing. Planner/Calc/TM Comshare. Orchestra 90/TM Software Affair. 13 Ghosts/© Bryan Eggers and Larry Payne



## Contents:

<b>Bugs, Errors, and Fixes</b> .....	37
<b>Color Computer</b>	
Magnificent Micropainter by Maggie Sherrod-Stone .....	49
OS-9 Assembly Language Programming by Earl Bollinger .....	41
Programs	
Drawing With Joysticks by Will Yeckley .....	46
Indian Blanket by Eric M. White .....	47
Magic Color Square by Joe Ryan .....	47
Squares and Circles by Karen Saivini .....	45
Symmetry by Stefan Cernusca .....	48
Vertical Variation by Stefan Cernusca .....	48
<b>Computer Clubs</b> .....	20
<b>Computer Customer Service</b> .....	15
Basic Graphics	
<b>Data Bases</b>	
Bulletin Boards .....	12
CompuServe	
The Needs of Disabled Persons in the Computer Revolution	
Profile .....	7
Accessing Profile 16 Data from TRS-XENIX	
<b>Education</b>	
Pascal Programming on Network 3 .....	9
<b>Fort Worth Scene</b> .....	2
<b>General Interest</b>	
The Interactive Fiction Worlds of Infocom by Elizabeth Langosy .....	4
The Software Maintenance Program by Linda Miller .....	13
TRS-80 Microcomputer News Reprints Sale .....	17
<b>Magazines</b> .....	31
<b>Model I/III/4</b>	
Orchestra 90's Special Composer Commands by Bryan Eggers .....	18
Programs	
Amazing Random Mazes by Glen Cheatham .....	21
Pattern by Scott Shephard .....	21
Pixels in Circular Motion by John F. Flitton .....	21
Printer Graphics Generator by David Whitney .....	23
Weaver by Rejean Giasson .....	23
<b>Model II/12/16</b>	
Business Reports That "Show" and Tell with Business Graphics .....	32
Analysis Pak for Models II/12 by Carol Morton	
<b>Model 100</b>	
Programs	
Model 100 Screen Graphics .....	10
<b>Model 2000</b>	
Programs	
Color Change Routine for Model 2000 COBOL by Dennis Weiss .....	26
Tandy Model 2000 Character Editor by Paul Opitx .....	28
<b>Notes on Previous Newsletters</b> .....	36
<b>Peripherals</b>	
Programs	
Bit-Image Graphics for the LP VIII by Earl R. Kooi .....	35
<b>Pocket Computer</b>	
Programs	
Calendar Program for the PC-2 by H. B. Franklin .....	38
Italian Flag by Fabio Marzocca .....	40
Super Graph by Richard B. Drumm .....	39

**Prices shown in TRS-80 MICROCOMPUTER NEWS are in U.S. Funds.**

Programs published in the Microcomputer Computer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack cannot assume any liability for the accuracy either of the programs themselves or for the results produced by the programs.

Further, while TRS-80 Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such cannot be supported by the Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question.

---

# The Interactive Fiction Worlds of Infocom

Elizabeth Langosy  
Infocom, Inc.

It's a cold Friday evening in February 1938. Storm clouds are building in the sky and the lights of Los Angeles glow faintly in the distance. Or perhaps it's a gloomy afternoon in the Troll Room. Bloodstains and deep scratches mar the walls and the troll himself, nasty-looking axe in hand, blocks all exits. Or it may be another routine day of drudgery aboard the Stellar Patrol Ship Feinstein. Although the metal deck has been scrubbed to a high gleam, it's still not clean enough for the dreaded Ensign First Class Blather.

These are the interactive fiction worlds of Infocom, where the computer screen shows not gobbling critters or dive-bombing aliens, but text which forms mental images far more realistic and complex than computer graphics could possibly be. Infocom draws on the imagination, putting the player right at the scene of the crime or the lab room of the deserted planet. Or at the hapless mercy of the troll.

Interactive fiction is similar to reading a book only in that text, rather than pictures, is involved. In interactive fiction, you have the opportunity to influence the course of the story by responding to it in the form of English-language commands typed into the computer.

In this way, you are constantly interacting with other characters, objects, and situations in the story. Items discovered in the course of the story can be retrieved and held onto for later use. There are choices of paths to follow and places to visit. There are even decisions to be made about when to sleep or eat and, in some cases, disastrous results if you neglect to do either. Other characters inhabit the environment. They move independently through the game, responding to you only if you happen to be where they are and choose to interact with them. And in making these choices—whether to speak with Monica or concentrate on Mr. Linder, whether to take the flask or leave it on the shelf, whether to walk through the forest or enter the house—you go far beyond the realm of conventional fiction.

Innovative packaging, sets the mood of each Infocom game. The 1938 [Santa Ana Register](#) included with the mystery game "The Witness" puts you right into the atmosphere of pre-war California, with its society balls and sensational court cases. And the accompanying [National Detective Magazine](#) provides not only tips for greenhorn detectives, but comprehensive information on the logistics of the game itself. Here's where you find out how to command the computer and what sort of responses you can look forward to. The game packaging also provides an introduction to the rich plot and character development that can be expected in the Infocom game.

Infocom's interactive fiction games offer challenging, intelligent puzzles. Solving one of these puzzles is often a complex process, in which items must be obtained at various locations and then put to use. As in the world outside the computer, nothing is easy. There are unforeseen circumstances and unavoidable setbacks, as well as the satisfaction of carefully seeing your task through to completion.

Three Infocom games are now directly available from Radio Shack: "Zork I" (26-1951), for the TRS-80 Model III or Model 4 (in Model III mode), and "Planetfall" (26-5377) and "The Witness" (26-5376), for the Tandy Model 2000.



## ZORK I

In "Zork", you venture into a fantastical world of mazes, trolls and treasures. The game begins in an open field. There is a white house with a boarded front door and a small mailbox. To the north, a narrow path winds through a forest. You find a window slightly ajar at the back of the house and with great effort, open it just enough to climb through. You are now inside the kitchen of the white house and on your way to the underground caverns of "Zork". Ahead lurk creatures waiting to thwart you attempts to find the treasures of "Zork", as well as rivers to be crossed, passages to be followed and hardships to be overcome.

The rooms of the underground empire have their own special hazards. You might, for example, find yourself in the Loud Room.

*This is a large room with a ceiling which cannot be detected from the ground. There is a narrow passage from east to west and a stone stairway leading upward. The room is deafeningly loud with an undetermined rushing sound. The*

sound seems to reverberate from all of the walls, making it difficult even to think.

Other rooms have characteristics which are less innocuous, although equally intriguing.

*You are in a small room. Strange squeaky sounds may be heard coming from the passage at the north end. You may also escape to the east.*

The word "escape" should be enough to warn you that danger might lie in the vicinity. And it does—in the form of a vampire bat. The creatures to be found in "Zork", as in all other Infocom games, each have their own distinct characteristics.

*In the corner of the room on the ceiling is a large vampire bat who is obviously deranged and holding his nose.*

Or perhaps you might find yourself confronting the cyclops.

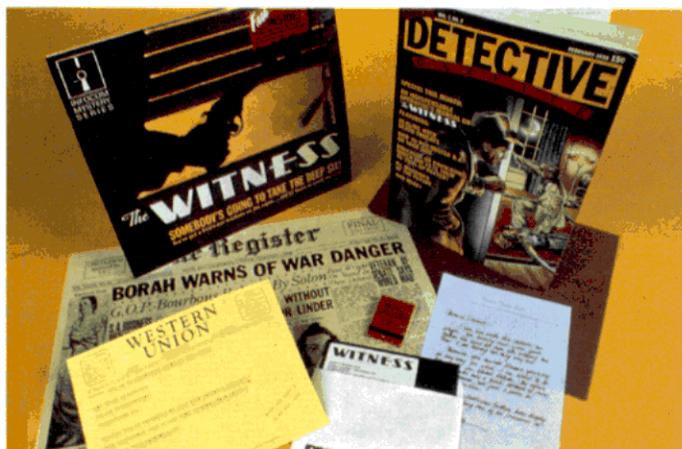
*A cyclops, who looks prepared to eat horses (much less adventurers) blocks the staircase. From his state of health, and the bloodstains on the walls, you gather that he is not very friendly, though he likes people.*

An adventurer who is not careful might prove to be a nice snack for the cyclops. There are, however, ways to overcome the dangers. One is by prudence—although no adventurer holds much stock in being too careful—and the other is by using the various implements which are found lying around the rooms of the white house and the underground empire. These items, such as a nasty knife or a clove of garlic, provide protection in unexpected encounters. The treasures are also found in the various chambers, but you must overcome the perils of the environment and the presence of the other inhabitants of the empire in order to obtain them.

The underground empire contains not only danger, but humor and beauty as well, such as in the engravings found on the wall of the Engravings Cave.

*The engravings were incised in the living rock of the cave wall by an unknown hand. They depict, in symbolic form, the beliefs of the ancient Zorkers. Skillfully interwoven with the bas reliefs are excerpts illustrating the major religious tenets of that time. Unfortunately, a later age seems to have considered them blasphemous and just as skillfully excised them.*

The adventure ends when you find all the treasures and safely return with them, leaving behind the great underground empire of "Zork".



## THE WITNESS

*Somewhere near Los Angeles. A cold Friday evening in February 1938. In this climate, cold is anywhere below about*

*fifty degrees. Storm clouds are swimming across the sky, their bottoms glowing faintly from the city lights in the distance. A search light pans slowly under the clouds, heralding another film premiere. The air seems expectant, waiting for the rain to begin, like a cat waiting for the ineffable moment to ambush.*

*The taxi has just dropped you off at the entrance to the Linder's driveway. The driver didn't seem to like venturing into this maze of twisty streets any more than you did. But the house windows are full of light, and radio music drifts toward you. Your favorite pistol, a snub-nosed Colt .32, is snug in its holster. You just picked up a match book off the curb. It might come in handy. Good thing you looked up the police file on Mrs. Linder's death. Her suicide note and the newspaper story told you all you know about the family. The long week is finished, except for this appointment. But why does an ominous feeling grip you?*

Thus the stage is set for Infocom's "The Witness", in which you investigate a threat on the life of Freeman Linder, wealthy philanthropist and husband of the late Virginia Linder. Virginia died recently of a gunshot wound, an apparent suicide—at least, according to her farewell note to her daughter Monica. The note is part of the colorful packaging for "The Witness", along with the Santa Ana Register (containing Virginia's obituary), the National Detective Magazine, a matchbook from the Brass Lantern Restaurant, and a telegram from Freeman Linder to you, the Chief Detective at Police Headquarters. It's in response to this telegram that you arrive at the Linder household at the start of the game.

The detailed packaging and atmospheric descriptions are both part of Infocom's interest in making you feel as involved in the game as possible. The author of "The Witness" is Stu Galley. His research included a 1937 Sears catalogue, a directory of American slang, and a history of advertising, all to ensure that the language, clothing, furnishings, and even brands of liquor were true to the time and place of his story. The characters in the game are also fully described, such as the ubiquitous butler, Phong, who opens the door of the Linder household:

*Phong's straight black hair and folded eyelids make him obviously Asian, but no definite nationality. His open, almost gentle face holds a quick smile and eyes that seem to miss nothing. He carries his stout body lightly, but you can see great strength under his light shirt and dark trousers. You guess his age at about fifty, but who knows how many lifetimes of experience he carries?*

Phong leads you into the living room, where Mr. Linder and Monica are talking. In period fashion, the room is decorated with a rattan davenport and club chair and bamboo print cushions. Monica doesn't seem too pleased with the intrusion, but Linder is eager to talk. At least, that's what you assume as you're led into his study. But it turns out that Linder is not quite as ready to provide information as you might like. As time passes, you realize you're under Linder's control, trapped in his office and searching for facts. Monica comes in and out, the cat dozes in the corner, Phong is presumably moving about inside the house. Is Linder trying to hide something? There's not much time to find out, because a sudden shot from the dark garden outside the study windows leaves the philanthropist slumped in his chair, quite dead. And you, unexpected witness to the crime, are now free to prowl about

the house, piecing together the solution to this intriguing mystery.

During the hour prior to his untimely demise, Linder showed you a threatening note received from a man named Stiles, alleged lover of Virginia Linder. This is the note which precipitated Linder's telegram. Although Linder didn't seem to have much to say about Stiles, the young man himself becomes involved in the case when he's brought into the house handcuffed to your trusty assistant, Sergeant Duffy. Duffy has shown up in the nick of time, worried about his boss and eager to offer any assistance, from analyzing fingerprints to giving advice.

Monica comes home from the movies, the coroner's van departs with the body, Duffy goes to and from the lab, Phong meditates in his room. As in any police investigation, there are clues to be discovered in various parts of the house and interrogations of sometimes unwilling suspects. Because each character has a distinct personality, you confront the very human problem of dealing with the quirks, deceits, and secret plans of four unique individuals.

The mystery must be solved within 12 hours. If you make an arrest, the case is tried by a jury. The jury then informs you whether the accused is guilty or whether there's not enough evidence for a conviction. If no arrest has been made within 12 hours, Police Chief Inspector Klutz arrives to take you off the case. However, there's no need for disgrace, because in this investigation, there's always the opportunity to try again.



## PLANETFALL

In "Planetfall", Infocom author Steve Meretzky has created a futuristic universe with its own language, games and scope of exploration. But the small human joys and frustrations are the same as those experienced back in the twentieth century.

The game packaging provides you with a humorous introduction to life aboard the Stellar Patrol Ship Feinstein, where you are a lowly Ensign Seventh Class. The most important item is a diary you have been keeping for the past few months. As the diary reveals, you're beginning to suspect a career in the Stellar Patrol might not provide the rewards you'd anticipated. After all, you're still an Ensign Seventh Class after two tours of duty and, worse yet, at the mercy of Ensign First Class Blather. Blather hands out hundreds of demerits at the least provocation, confiscates the equipment for the Ensigns Seventh Class Double Fanucci Tournament, rejects your request for astrophysics training and consigns you to the distasteful chore of cleaning out the grotch cages.

So it's almost a welcome change when you find yourself in an escape pod, fleeing from an explosion on the S.P.S. Feinstein. The patrol ships' mission was to effect a near galactic union, following the collapse of the Second Union and subsequent Dark Ages. The very motto of the Stellar Patrol—"Boldly Going Where Angels Fear to Tread"—indicates the near-barbarous state to which the galaxy has descended. And it's towards just such a forsaken corner that you find yourself hurtling in your escape pod. The pod lands with a thud and you step out onto the planet of Resida.

From the escape pod, it could be seen that Resida is composed of twin islands, both of which now appear to be totally deserted. You wander through rooms and down corridors. Tools acquired along the way help you evade future predicaments and in most cases several items need to be used in conjunction with each other to solve a problem. For example, you might come upon the machine shop.

*This room is probably some sort of machine shop filled with a variety of unusual machines. Doorways lead north, east, and west. Standing against the rear wall is a large dispensing machine with a spout. The dispenser is lined with brightly-colored buttons. The first four buttons, labelled "KUULINTS 1 - 4", are colored red, blue, green, and yellow. The next three buttons, labelled "KATALISTS 1 - 3", are colored gray, brown, and black. The last two buttons are both white. One of these is square and says "BAAS". The other white button is round and says "ASID".*

When you decide which button to push, and do so, some sort of chemical fluid pours out of the spout, spills all over the floor, and dries up.

Obviously, it's necessary to find a vessel in which to collect the fluid. Sooner or later there will probably be a use for it, although the future use might turn out to be for one of the fluids that was not collected.

Eventually, you meet a robot named Floyd. Funny and cute, with the personality of an inquisitive 8-year-old, Floyd is an irresistible companion. He also has the intelligence necessary to help you through some tight spots. He's affectionate enough to want to stay close and although he'll bound off to explore the surroundings, he'll always show up again.

Floyd doesn't seem to have much thought for danger, although he's given to chanting the death scene from "Carmen". If on first meeting Floyd, the suspicious player attempts an assault, this is the result:

*Floyd starts dashing around the room "Oh boy oh boy oh boy! I haven't played Chase and Tag for years! You be it! Nah, nah!"*

It is necessary for you to eat, drink, and sleep in order to survive. Warnings of hunger, thirst, or fatigue are given and disregard of such essential human needs can result in the announcement

\*\*\*\*You have died\*\*\*\*

Of course, it's also important to be sure that what you consume is food, not poisonous chemicals, and that your night's rest is taken in a secure place, free from man-eating grues.

Through your explorations of the twin islands, you learn the secret of the planet. There are systems to be repaired in order to bring Resida back to life and the neglect of any one of them means certain doom. When all the systems are properly restored, you've earned your ascent from underling to hero, reviving a long-dormant planet and bringing it once again into the Galactic Union.

# Accessing Profile 16 Data from TRS-XENIX

The Small Computer Company  
 P.O. Box 2910  
 Fort Worth, TX 76113-2910  
 By Ivan Sygoda, Director, Pentacle  
 Copyright 1984, Ivan Sygoda. All rights reserved.

In addition to its intelligent concept and clever design, Profile 16 is powerful because it makes good use of the sophisticated TRS-XENIX operating system on which it is based. This sophistication and power is directly available to you, providing you take advantage of it. Some of the most popular articles in this series have concerned customizing systems with operating commands or BASIC, and I think Profile 16 users will be just as eager to learn how to take advantage of XENIX in customizing their Profile applications. We'll examine some XENIX commands and procedures, which should prove helpful in this regard.

In the chapter on defining user menus, the Profile 16 manual describes some of the ways direct access to XENIX increases your ability to manipulate data and output. Each user menu selection that you define sends Instructions to the XENIX, exactly as if you had typed the command at the XENIX user prompt. Thus, learning to define Profile 16 user menus and learning to use XENIX more effectively go hand in hand. The goal is to move with confidence beyond the suggestions provided in the manual.

## PEP TALK

Yes, XENIX can be scary. It's big, complex and powerful. To make matters worse, the XENIX manual cautions that it is possible to zap everything if you make a mistake. True. But you might also get run over by a car any time you cross a street. That's no reason to stay at home. You simply look both ways before stepping off the curb. Equally common-sensical precautions apply when you experiment with XENIX: 1) always keep backups of your files; and 2) stay out of root (super-user) mode. (You'll know you're in root when the screen prompt is a "#" instead of a "\$".)

Also, it's a good idea to do a little background reading. Here are the names of a few books I found to be helpful in learning about XENIX: Rebecca Thomas and Jean Yates, [A User Guide to the Unix System](#) (Berkeley: Osborne/McGraw-Hill, 1982); S.R. Bourne, [The Unix System](#) (London: Addison-Wesley Publishing Company, 1982); Richard Gauthier, [Using the Unix System](#) (Reston, VA: Reston Publishing Co., 1981); and Mitchell Waite, Donald Martin and Stephen Prata, [Unix Primer Plus](#) (Indianapolis: Howard M. Sams & Co., 1983).

As you may know, XENIX (written by Microsoft) is Radio Shack's version of AT&T's UNIX operating system. Most of what you'll find in these books applies directly to XENIX. A

few commands may be different, depending on the version. If you're not sure whether or not a particular command works in your version, simply try it. XENIX will tell you if it "can't find" the command.

Finally, the on-line reference manual available with the XENIX Development System is one of the handiest sources of detailed information about commands. If you've installed the Development System, you can learn the ins and outs of the "sort" command by entering "man sort" at the user prompt. After a few moments, the first of many "pages" of information appears on the screen. The first rule for coping with all this is: Don't freak out! This special manual tells you more than you could possibly need or want to know the first time you use a command. That's why it's useful to have a specific task in mind when you first delve into XENIX. It gives focus to what is otherwise information overload.

## SCREEN DISPLAY

The first problem we're going to attack is how to display reports on the screen instead of sending them to the printer. If you read the previous articles on how to do this from Profile Plus and Profile III Plus using BASIC, you'll recall that in each case it took a whole program in BASIC. With XENIX, the same task can be accomplished with one command line! The trick is to isolate the building blocks involved and then assemble them in the proper order. But first another preamble of a general nature to help newcomers to XENIX get oriented.

Video displays (monitors, CRTs, screens) are relative newcomers to the world of computing. Most terminals were originally just printers—hardcopy devices—such as teletype machines. This suited the business world, which has used hardcopy to communicate information ever since the first travelling salesperson scratched the first invoice on a clay tablet. Today, however, it is feasible for everyone to have a terminal at his or her desk; we're beginning to reach for the keyboard instead of the file cabinet.

Operating systems such as XENIX, having started life on large computers, bear the imprint of this history. Take the "pr" command, which prints a file. The verb itself implies hardcopy, and many of the options available for this command are designed to format printed output—page length and width, presence or absence of headers and footers, etc. XENIX accommodates the screen display first by engaging in a bit of verbal acrobatics. One says that "pr" prints the file involved

"to the standard output." This standard output used to be the line printer. Now it is the display screen. What is involved is a process of adaptation, of modifying procedures invented for printer to fit the newer display screen environment.

However, although screens can be varied and many, they can only show one record at a time. If you want report-like formats directed to the screen, you have to make an extra **effort** which involves familiarizing yourself with a few XENIX commands and their options. End of preamble.

## BUILDING BLOCKS

We'll start with the example furnished in the Profile 16 User Manual in the chapter called "Defining User Menus." You should also read Appendix A, "Names of Programs and Files." I stress this because the appendices are ordinarily among the last sections you'd read in the course of setting up and using a Profile data base.

Using the tutorial file called "TClient," the manual instructs you to enter the following on a user menu command line:

```
/pf/report TClient -F Invoice -IA -P /tmp/invoice; more /tmp/invoice
```

Analysis: "/pf/report" is the pathname of the Profile 16 runtime program ("report") which generates output, usually printed output. "pf" is the name of the directory that contains "report" and most of the other programs which instruct Profile how to do its job. (Data and user-defined formats are in another directory called "profile." By the way, this directory is completely different from the similarly named ".profile" directory—pronounced "dot profile"—in which XENIX stores information about each user.)

Note: the slashes "/" are used in XENIX to separate the elements of a path name. The XENIX file structure has been described as a tree with branches, twigs and leaves. The leaves correspond to actual files; the branches and twigs are the various levels of nested directories. One leaf can have the same name as another leaf. They are different and distinct files, however, because they hang on different twigs. That is, they have different path names. In this sense, the path name is an integral part of any file's name. The initial slash is important; however, it's easy to forget when typing. XENIX find's "pf/report" (without the initial slash) only if you are working in the "parent" directory of "pf," that is, a directory called "appl," which contains Radio Shack's TRS-XENIX applications programs. Since Profile 16 user menus are in the user's "home" directory, it is necessary to move to the "pf" directory. This is the function of the initial slash. If you are not using a user menu, but entering a command directly at the XENIX command line, indicated by the "\$" prompt, you have to get yourself into the proper directory by typing "/appl/pf/report" instead of just "pf/report." You'll get used to all this. "TClient" is the file name of a data base. It must be separated from the preceding path name by a space. Remember that uppercase/lowercase distinctions are significant in XENIX file names. "TClient" is not the same as "tclient."

Following the file name, and separating each element with a space, are various options or "flags." In XENIX, options are indicated by a hyphen "-" immediately preceding the character that serves as the flag. "-F" introduces the name of the output format that the report generator uses to format your data. "-I" introduces the index (automatic A or B, or

demand indexes 0-9) according to which the report is sorted. "-P" indicates where the output is sent. XENIX treats output devices (printers, display screens, etc.) exactly the same way it treats data files. This stroke of structural genius on the part of XENIX's grandfathers at Bell Labs gives you great flexibility: instead of being sent to the printer, the report is sent to a file named "/tmp/invoice."

"/tmp" is a special directory where you can send files temporarily. The contents of the /tmp directory disappear when your system is shut down. This is a good idea because XENIX is a pack rat. It contentedly collects files until your hard disk is full. The name assigned to the report in the /tmp directory is "invoice."

So far, we've told Profile and XENIX to do the following: organize data from the TClient file according to the sort and select criteria specified in automatic index A; print it according to the format called "invoice;" and send it to a temporary file called "invoice," instead of to the line printer.

## IF SOME IS GOOD, MORE IS BETTER

The last part of the command line tells XENIX to send that temporary file to the display screen. This is a new command sequence that must therefore be separated from the previous command sequence by a semi-colon (";"). The semi-colon functions exactly the way the colon (":") does in a BASIC program. It allows you to place, on one line, commands that would otherwise have to be placed on separate lines.

The command that sends data to the screen in a user-friendly way—22 lines at a time—is "more." (The "cat" command also sends data to the screen, but scrolls it continuously until the end of the file is reached, too fast for ordinary mortals to read. Pressing **HOLD** stops the scrolling, but you need pretty sharp reflexes.) "/tmp/invoice" is the name of the file to be more'd. Finally, typing **ENTER** causes XENIX to execute the whole command sequence. This is enough for one session. Using the example given, send your own report formats to the screen. If you like accessing your data this way, it is probably worth the effort to create a report format designed specifically for screen display. It would be 80 columns wide, and each page would be 22 lines long (to match the "more" format). You can add "lpr/tmp/invoice" to the menu command and hardcopy the data as it appears on the display screen.

Experiment. You have nothing to lose but a /tmp file. Everything we've done is available with the XENIX runtime system delivered with your Model 16.

*PROFILE Editor's Note: Ivan Sygoda's column will be continued in the Small Computer Company's newsletter, Small Talk. Contact Gary Stix at (212)398-9290 for more information.*

*Pentacle is a New York City-based non-profit service organization specializing in administrative services for performing art groups.*

# Pascal Programming on Network 3

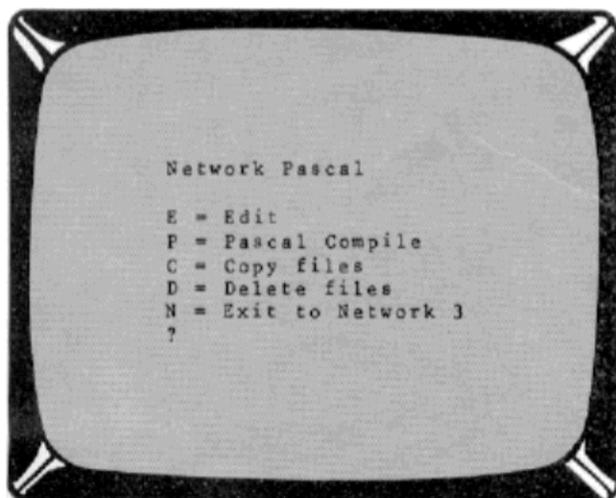
A new Pascal programming package, designed for use on a TRS-80 Network 3 system, will be available later this year through your local Radio Shack store or Computer Center. Pascal is a structured programming language commonly used in teaching. The standard college advanced placement test for computer programming now includes some questions on Pascal.

TRS-80 Network Pascal (26-2739) was designed for use by programming classes on a Network 3 system. The Network 3 system allows students, at up to 16 non-disk TRS-80 Model III or Model 4 student stations, to share the disk capabilities of a single, central "host" disk computer. With Network Pascal and with at least 48K of memory in the student stations, students can learn to write, edit, compile, and run their own Pascal programs, all at the student stations.

Network Pascal minimizes the time required for each student station to access the host disk. The student's program is stored in memory at the student station, which eliminates any need to re-load the program from disk each time editing or compiling is performed. During editing or compiling, the network is not used. The Network Pascal editor/compiler is a single program, with program overlays (extra portions of program code specific to the editor or the compiler). When you switch between editing and compiling, the only thing that needs to be loaded is the appropriate set of overlays. This means that only a small amount of time is required in switching between editing and compiling.

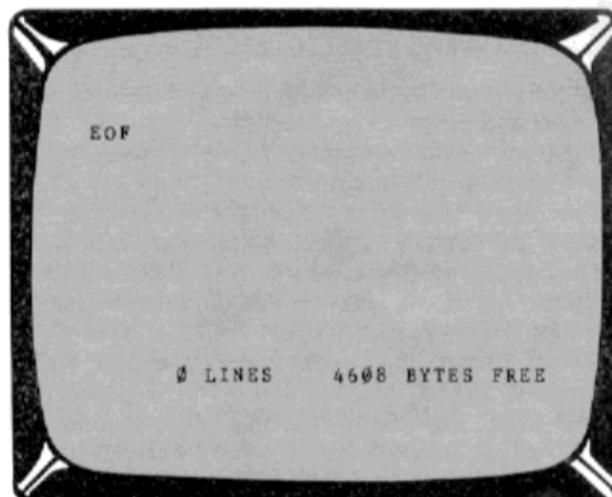
Let's take a look at how you'd use Network Pascal to enter, compile, and run a short Pascal program.

When the Network Pascal compiler/editor is loaded into the student station, a menu of options appears:



The menu allows you select operations easily.

E to Edit gets you into the Network Pascal editor program, where you can write a program and make any needed corrections. When the Edit option is selected and no program is currently in memory, an almost blank screen will appear:



On this screen, you can enter a Pascal program. To enter text, you press **(SHIFT) @** to insert a blank line, then you type the line (using the Pascal language syntax) and press **(ENTER)** at the end of the line.

Arrow keys may be used to move the cursor, and many other special key functions are available for inserting and deleting characters, indenting text, and performing other text editing functions. You aren't limited to the number of lines on the screen—Network Pascal can handle medium-sized programs (up to 100 or 150 lines), the program size that is most typical for teaching situations.

The short program below might be entered:

```
PROGRAM test;
BEGIN
WRITELN('* I am a Pascal wizard. ');
END.
```

Once you had entered the program, you could exit the Network Pascal editor by pressing the **(CLEAR)** key followed by **(C)**, then entering the word "exit". The Main Menu for Network Pascal would reappear.

The next step in using Network Pascal is to compile the program. The Network Pascal compiler takes the program text typed into memory (this program text is known as "source code") and translates it into a form that the computer can execute (the resulting program is the "object code"). At the same time as it translates the program, the Pascal com-



piler looks for errors in the source program and reports any errors it finds to the programmer.

To use the Pascal compiler, you'd simply type (P) at the Network Pascal menu, then respond (Y) when you see a message that asks if you want object code generated. As the program is compiled, a listing of the program is sent to the screen, along with error messages if any errors were found. (If errors are found, they can be easily corrected using the Network Pascal editor.)

Once a program has been compiled without errors, it can be saved to disk and then run. The C option at the Network Pascal Main Menu is used to copy a program to disk. Simply type (C) at the Main Menu, and prompts appear asking for specifics and for a filename. It is a good idea to copy both the source and object code in the student station's memory onto the host disk.

Once the program is stored on disk, you can exit the Network Pascal editor/compiler. You can then run the program using the RUN program stored on the Network Pascal disk. To run the program, you'd start out at the Network 3 prompt. Simply enter RUN followed by the filename you used when you stored the program's object code on disk. For the example program type: RUN TEST.

If you ran the little sample program TEST listed above, Pascal would first ask you to specify input and output files; then if you pressed (ENTER) at the input and output prompts, the output of the program would be displayed on the video screen (input isn't used by this program). In this case, the output is the message:

```
* I am a Pascal wizard.
```

## TEACHING MATERIALS IN NETWORK PASCAL

The Network Pascal package comes with a user's guide that includes:

- a "beginner's guide" to introduce you to the use of the program
- a Pascal language tutorial which demonstrates the fundamentals of Pascal programming
- a language reference guide with more advanced programming information
- an editor reference guide
- a system reference guide

16 student manuals, with the same content, are included for use by student programmers.

The Network Pascal package also includes the Pascal development system for stand-alone TRS-80 Model III and 4 disk systems. These diskettes for the stand-alone machines allow further flexibility in the classroom configuration.

For more information on the availability of Network Pascal, and how Network Pascal can be used in your school, contact your local Radio Shack store or Computer Center, or the Radio Shack Regional Educational Coordinator for your area.

## Model 100 Screen Graphics

This program will produce a changing graphic display on the Model 100 screen. Hit any key to clear the screen if it's getting too cluttered. Hit (BREAK) to stop it.

It's a nice one to leave running while you do something else . . . just remember to use your AC adapter! It doesn't take up much memory either.

```
10 CLEAR 12
   : CLS
20 S = VAL(RIGHT$(TIME$,2))
   : FOR I = 1 TO S
   : T = RND(1)
   : NEXT I
30 T = INT(RND(1)*2+1)
   : X1 = INT(RND(1)*239)
   : X2 = INT(RND(1)*239)
35 Y1 = INT(RND(1)*63)
   : Y2 = INT(RND(1)*63)
40 IF INKEY$ <> "" THEN CLS
45 ON T GOTO 50,60
50 LINE(X1,Y1)-(X2,Y2),0,BF
   : LINE(X1,Y1)-(X2,Y2),1,B
   : GOTO 30
60 LINE(X1,Y1)-(X2,Y2),1,BF
   : GOTO 30
70 CLS
   : GOTO 30
```



# The Needs of Disabled Persons in the Computer Revolution

*Editors Note: The CompuServe Consumer Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. The CompuServe Consumer Information Service and Executive Information Service are sold at Radio Shack stores nationwide and in Canada.*

The knowledge we garner from being able to hear and touch what is around us goes unappreciated by many people. Yet, on reflection, it is these sensory capabilities that are the basis for how we communicate with, learn about, and explore the universe around us.

For the handicapped, who do not possess one or more of such capabilities, the advent of the personal computer has been especially important—it has opened up many opportunities that were limited before or not available at all. The capabilities to access the news, take care of personal needs like shopping or banking or hold a simple discussion were not easily available to an individual with a disability.

From its inception, the CompuServe Consumer Information Service has provided many of the communications tools that are now invaluable to disabled subscribers.

Of these tools, The CB Simulator and the various Special Interest Groups have been the most influential. One comment by a user who is totally paralyzed speaks of the value of the Special Interest Groups and CB in the lives of the handicapped—"People hesitate to talk to or make friends with someone in a wheelchair or someone with an obvious physical disability. I have never seen . . . a physical disability on CB . . . It transcends the physical attributes of the user, making all equal."

The National Issues and People SIG (NIPSIG) has long been a gathering point for many handicapped users to discuss issues affecting the handicapped. NIPSIG is recognized as a feature through which many disabled users meet others who are more experienced in using the service. CB is also popular with handicapped users. The anonymity it provides enables an individual's ideas to come through so that they become the only criterion by which he or she is judged. A disability, which in a traditional face-to-face meeting might create a barrier between two people, becomes irrelevant.

While loneliness may be a major problem faced by the handicapped, lack of access to timely information (especially in our information-driven society) is often a major obstacle to full participation in society. The Consumer Information Service can help the disabled by providing access to a wide range of information—information that is available whenever a disabled person wants it, without having to rely on someone else to "translate" it for them. The ability to know the latest news from the AP, the score of last night's basketball game

and to engage in a lively on-line conference on teaching methodologies helps eliminate the personal and knowledge barrier that the handicapped have had to face. The personal computer lets the disabled person keep current the same way other people do.

## SERVICES FOR THE DISABLED

While a good argument can be made for the value of the entire service to handicapped users, there are several services that address the needs of the disabled specifically. Currently, there are two services in this area: The Clarke School for the Deaf and the Handicapped User's Database.

The English language is the primary medium schools use to educate and which we rely upon to learn. A student with impaired hearing is cut off, either partially or completely, from this language making regular education in a traditional school a significant challenge. In the case of those with profound or total hearing losses, some degree of special education is first required in order to help them develop basic receptive and expressive communication skills. Helping deaf people function as independently as possible in a world that functions largely on sound has been the aim of the Clarke School for the Deaf since its founding in 1867.

The Clarke School for the Deaf is a privately endowed boarding and day school for deaf children located in Northampton, Mass. At present, over 120 adults are employed to provide an educational program for approximately 120 children on campus and over 90 children mainstreamed in their home town programs.

From its inception, the Clarke School opted for the oral method of education for the deaf. The school's educational program helps its students develop the speech, lipreading, and language skills they need to enable them to continue their education in schools for hearing children. This transition is called mainstreaming and it is Clarke School's major goal for its students. Since 1875, most alumni of the Clarke School have completed their high school education in a regular school with hearing students. The Clarke School for the Deaf's service can be accessed via the Home Services Education menu or directly by entering GO CSD.

The Clarke School for the Deaf's current service focuses on the mainstreaming process. Its newsletter, The Mainstream News, contains articles for teachers, tutors, guidance counselors, speech teachers, parents, and students. Although it is written primarily about mainstreaming on the high school level, it also contains information which is applicable to younger students.

Another feature of the Clarke School area is Feedback. Using this, it is possible to send comments or questions to the school or have your name added to its mailing list.

---

The Handicapped Users' Database is an exhaustive compendium of information for and about the handicapped. It covers everything from the latest developments in software and hardware, to a News section in which national and international news of interest to the handicapped is featured.

The Hardware and Software Reviews section provides in-depth information about the technical specifications, as well as the capabilities of microcomputer products of potential use to the handicapped. For example, the Hardware section contains an in-depth analysis of the new micro-based braille translation system and an overview of the new Voice Information Processing System (VIP).

The Organizations section covers the various organizations both nonprofit and commercial that provide services for the handicapped. Nonprofit organizations such as the National Braille Association and the National Muscular Dystrophy Association are featured. On the commercial side, such companies as Triformation Systems, Inc., which is responsible for the VIP system, are featured.

The National News Notes section is invaluable for keeping abreast of the latest national and international developments that have an impact on handicapped. The availability of a voice-indexed dictionary for the blind and handicapped from the National Library Service for the Blind was the subject of a recent national news story. The National News Notes' Profiles section highlights individuals with various disabilities and lets others share their experiences in dealing with their situation and, hopefully, learn from them.

Featured in this group is the amazing individual who is responsible for gathering, organizing, and editing the entire HUD database—Georgia Griffith. Georgia, who is both deaf and blind, has been working on the HUD database for about two years, ever since she began receiving a variety of requests for information for the handicapped. As the number of requests increased, handling them individually became impractical and she began working on a central area where all of her knowledge could be made accessible to the most people. This was the beginning of the Handicapped Users' Database which now contains hundreds of pages of information. HUD is accessible via the Home Education menu or directly by entering GO HUD.

There are disabled people who need help; there are disabled people who have found ways to help themselves and their peers; there are people with no disabilities who are able and willing to help handicapped persons when such assistance is needed and desired; and there are persons who have no information about the several forms of disability but would like to learn more. A major value of the Consumer Information Service is that it has the capability for allowing these diverse groups to communicate among and interact with each other.

While each of the several disabilities has its own unique characteristics, the one problem common to all has been dependence upon the assistance of others. The Consumer Information Service offers a vast potential for the development of greater independence by disabled persons. The benefits that this medium has brought to the handicapped are impressive and several additions are already being planned, such as a special interest group for the disabled. The synergy between the dedicated people who provide the information and the expanding scope of the service will continue to make this area grow.

*Questions and comments about the CompuServe Consumer Information Service can be sent to Richard A. Baker, editorial director, or Jacqueline A. Farthing, assistant editorial director, CompuServe Consumer Information Service, 5000 Arlington Centre Boulevard, P.O. Box 20212, Columbus, Ohio 43220 or through Feedback, main menu item 5, Consumer User Information.*

## Bulletin Boards

### CONNECTICUT EDUCATIONAL BULLETIN BOARD

There is a new Bulletin Board System for the New Haven, Connecticut area. It is called the Connecticut Educational Bulletin Board System. Access it free, at 203-795-3506. The hours are: Monday through Friday 9:00 P.M. to 4:00 A.M. and all day Saturday and Sunday.

Pasquale Tancreti  
35 North Place  
West Haven, CT 06516  
CompuServe I.D. 70215,347

### THE GAME MACHINE

I own and operate a Bulletin Board in Pomona, New York. There are seven boards with on-line games. Access time is from 7:00 A.M. to 11:00 P.M. (EST) at: 914-362-1422.

Seth Needleman, Sysop

### ISO-LINK BULLETIN BOARD

In January 1984, an on-line Bulletin Board Service (BBS) known as Iso-Link agreed to sponsor an Educational Section on its board. It is privately operated by Howard Leadmon of Baltimore, Maryland who is SYSOP of Iso-Link.

The areas covered in the Education Section are: Help, Science, Math, Reading, Other, and Teacher to Teacher. It was designed to give educators, administrators, teachers, and serious students an opportunity to react on various topics involving computers, software, and other facets involving education.

Currently, the board posts information on computer periodicals, educational on-line services, and BBS's. It allows free exchange of information of interest to the user and permits a new learning tool to disseminate the information.

Some additional sections are being planned for the board in the near future. The board operates 24 hours a day, 7 days a week, and is currently free. You can go on line with the board by calling 301-335-3103. This BBS is being run on a Radio Shack TRS-80 Model III.

David F. Menzies  
Baltimore City Public Schools  
Southeast Middle School #255  
6820 Fait Avenue  
Baltimore, MD 21224

# The Software Maintenance Program

*Subtitle: At Radio Shack We Support What We Sell*

By Linda Miller

Even extensive testing can fail to uncover every potential bug in a piece of software. Knowing that an error could occur in a released piece of Radio Shack software we were faced with three choices.

1. We could just ignore the problem. Why not? Other companies do.
2. We could fix the problem and leave it up to you to discover that we had a correction.
3. We could inform you as soon as we had a correction for a problem.

We chose option three. It seemed most consistent with our continued effort to support our TRS-80 customers. The Software Maintenance Program was implemented to accomplish this goal.

## TO BEGIN WITH

When you first open a Radio Shack disk software package, in addition to documentation and media, you should find:

A Software Registration Card

At least two Change of Address Cards

A Software Version Log with the heading Read Me First

While you may be tempted to toss these into the trash, don't do it. Using these items for their intended purpose can prevent problems down the road. A quick examination of them will clarify their individual purposes.

## THE SOFTWARE REGISTRATION CARD

**Software Registration Card**

Cat. No. \_\_\_\_\_  
Version **03.13.00**

**SAMPLE**

Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ Phone ( ) \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_

Filling out and returning the Software Registration Card sets in motion a vehicle to help you maintain your software in its most current and accurate condition. Pull any Software Registration Cards that may still be in your software package, fill them out, and mail them. This insures that your name, address, and current software version number are included in a mailing list maintained here at Radio Shack. Except on some cards printed early in this program, we pay the postage. Only those names on the list will receive software correction letters, and the best way to get your name on the list is by sending in the completed Software Registration Card. The card should already have the catalog number and version number of the software on it. You just fill in your name and address.

When the cards are received by us, the version and catalog numbers of the package are entered under the owner's name into a master mailing list that is maintained in our data processing department.

## THE CHANGE OF ADDRESS CARDS

**Change of address**

**NEW ADDRESS**

Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ Phone ( ) \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_

**OLD ADDRESS**

Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ Phone ( ) \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_

**SAMPLE**

If after mailing the original Software Registration Card to us, your address changes, you then fill out the change of address card and send that to us. It is important that you enter your old address just as it appears on the Software Registration Card. If you copy your address from the Software Registration Card into the Old Address area of one of the Change of Address Cards before you send in your Software Registration

Card, you will have a record of the address just as it appears on the Software Registration Card. The new address replaces the old address on the master mailing list in data processing.

### THE SOFTWARE VERSION LOG

If you look on your diskette you will see a sticker with a number like 01.00.01, 03.00.00, or Version 2.0. These are applications software version numbers. This version number should be entered in the Applications Software Version Log. You should also see a Disk Operating System (DOS) version number like TRSDOS 1.3 or MS-DOS 2.0 on the disk label. These are the version numbers for the DOS residing on the disk. Enter this version number under OP. SYSTEM VERSION LOG.

#### READ ME FIRST

All computer software is subject to change, correction, or improvement as the manufacturer receives customer comments and experiences. Radio Shack has established a system to keep you immediately informed of any reported problems with this software, and the solutions. We have a customer service network including representatives in many Radio Shack Computer Centers, and a large group in Fort Worth, Texas, to help with any specific errors you may find in your use of the programs. We will also furnish information on any improvements or changes that are "cut in" on later production versions.

To take advantage of these services, you must do three things:

- (1) Send in the postage-paid software registration card included in this manual immediately. (Postage must be affixed in Canada.)
- (2) If you change your address, you must send us a change of address card (enclosed), listing your old address exactly as it is currently on file with us.
- (3) As we furnish updates or "patches", and you update your software, you must keep an accurate record of the current version numbers on the logs below. (The version number will be furnished with each update.)

Keep this card in your manual at all times, and refer to the current version numbers when requesting information or help from us. Thank you.

APPLICATIONS SOFTWARE VERSION LOG	OP. SYSTEM VERSION LOG
03.13.00	
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

### IT'S REALLY QUITE SIMPLE

Briefly, you just:

1. Fill out the Software Registration Card
2. Copy the address exactly as it appears on the Software Registration Card into the Old Address area of one of the Change of Address cards.
3. Mail the Software Registration Card to us.
4. Update the Applications Software Version Log and the Operating System version sections of the Software Version Log with the current version numbers as they appear on your disk.

### WHAT HAPPENS NEXT?

Maybe a lot, or maybe nothing at all. Some software packages will require several changes and others will require none. It depends on the package.

If there is a necessary correction we will send you the changes in letter form through the mail. When you receive a letter, make the corrections as indicated. If you are still unsure of your ability to make the corrections after reading the instructions, your local store personnel can help you or as a last resort you may contact the Customer Service group in Fort Worth.

In addition to the corrections you make yourself, a letter will occasionally indicate a more extensive program revision which involves a new disk or documentation. This kind of fix involves what we call a 700 number item which you will need to pick up at your Radio Shack Computer Center or Radio Shack Plus Computer Center store or participating Radio Shack dealer. No disks will be sent through the mail.

Your letter will also indicate if there are any version number changes that should be recorded in your log. When applied changes create a new DOS version number or application version number, they will be indicated in the letter, and you should enter the new version number or numbers into the log.

Our master list is updated each time correction letters are sent, so we always have a record of the current version number of your software.

### WHAT ABOUT DISCARDED OR LOST CARDS?

What if unwittingly you have lost or discarded your Software Registration Card and now want to participate in this program? There are some steps that you can take. Write us a letter indicating that you want to be registered under this program for your software packages. Send the original title page (not a copy) of your software or a sales receipt (original or copy) showing your name and address to:

Radio Shack Computer Customer Services  
400 Atrium  
One Tandy Center  
Fort Worth, TX 76102

Another alternative is for the store to furnish from the store records written verification of software purchased.

When one of these conditions is met, Computer Customer Service will fill out a registration card for you and return your original title page, if appropriate.

### AND IF YOU OBTAINED YOUR SOFTWARE FROM AN INDIVIDUAL?

If as a second owner of a software package you want to become the registered owner, here's what you do. Submit a Change of Address card with the OLD owner as the OLD address and the NEW owner (you) as the NEW address. Send the card in and you will then be the registered owner of the software.

### A WORD TO THE WISE

Don't imagine that you will be inundated with letters making changes to every software package you buy from us. Some software packages will require no changes while others will require several changes or even a new release of the program. When software changes are necessary, we are ready to let you know about them.

The objective of the Software Maintenance Program is to supply owners of Radio Shack software with the most correct information as soon as it becomes available. Now that's support!

# BASIC Graphics

Graphics is something that nearly every computer user has some kind of interest in. This interest could be sparked by the use of graphics for a business application or by the simple enjoyment of playing with the computers capabilities.

Graphics can be broken down into three general categories:

1. Line printer graphics
2. Pixel graphics
3. Character graphics

## LINE PRINTER GRAPHICS

Line printer graphics refers to pictures, drawn either on the video screen or on the printer. These pictures are made up of characters printed at specific locations on the screen. These types of graphics are like a still photograph. Some examples of line printer graphics would include bar charts for business applications and wave charts for scientific applications. Among the more popular line printer graphic examples are party banners and simple pictures that most of us have toyed with.

To draw line printer graphics, you must first layout the diagram on a video worksheet. This will allow you to easily position the characters to achieve your desired output. Once you accomplish this then simply write a BASIC program that will print this picture to either the screen or to the printer.

Here is a sample program:

```
10 CLS
20 PRINT "HH  HH  EEEEE  LL      LL      000000"
30 PRINT "HH  HH  EE      LL      LL      00 00"
40 PRINT "HHHHHH EEEEE  LL      LL      00 00"
50 PRINT "HH  HH  EE      LL      LL      00 00"
60 PRINT "HH  HH  EEEEE  LLLLLL LLLLLL 000000"
70 END
```

Once you are satisfied with your output to the screen just change the PRINT's to LPRINT's and you will have the same output to the printer. Now that we have a basic idea on how to use characters to make simple pictures or messages on either the screen or the printer lets examine what makes up a graphic character.

## PIXEL GRAPHICS

The basic element of a graphics character is a pixel. Each graphic block is made up of six sections called pixels. The Model I and III screen will display sixty-four characters across one line and has a total of sixteen lines. However, each character is two pixels wide and is three pixels high. This means that there are a 128 pixels across each line since  $64 \times 2 = 128$  and we have 48 pixels down each column since  $16 \times 3 = 48$ . Taking this into consideration, when a program is written using pixels we refer to positions 0 to 127 across and 0 to 47 up and down. We also refer to the screen in terms of X and Y coordinates, where X is a horizontal position on the screen and Y is a vertical position on the screen. The three

basic commands used in pixel graphics are SET, RESET, and POINT.

First lets take a look at the SET command. SET is used to turn on a particular pixel point. The proper syntax to use is: SET(X,Y). For example, SET(35,25) lights up the block located thirty six pixels from the left of the screen and twenty six pixels from the top of the screen. The RESET command turns off the pixel. The proper syntax to use is: RESET(X,Y). Using the example above, to turn off that pixel we would type, RESET(35,25). Here is a simple program that will allow you to use the SET and RESET commands.

```
10 CLS
20 X = 35
30 Y = 25
40 SET(X,Y)
50 FOR Z = 1 TO 150
   : NEXT
60 RESET(X,Y)
70 FOR Z = 1 TO 150
   : NEXT
80 GOTO 40
```

To stop the program simply hit **(BREAK)**. The program will flash the pixel located at coordinate 35,25. Be sure to keep in mind that the pixel count starts at 0 (zero) and not at position 1. Notice the timing loop in lines 50 and 70, without these loops the flashing of the pixel would be almost impossible to detect. Of course this is a very simple demonstration of the SET and RESET commands. By adding another line we can make the blinking dot move. Try adding this line to your sample program.

```
75 X = X + 5
   : IF X > 127 THEN X = X - 127
```

To make the dot move diagonally we can add a second line.

```
77 Y = Y + 5
   : IF Y > 47 THEN Y = Y - 47
```

Notice when the dot reaches the bottom of the screen it reappears at the top of the screen. By using various combinations of the SET and RESET commands, one can design some relatively sophisticated graphics programs.

Another command used in conjunction with the SET and RESET commands is the POINT command. POINT looks at a specified graphics block and can tell you if that block is turned on or off. The proper syntax to use is: POINT(X,Y).

If the block is turned on a value of -1 is returned. And if the block is turned off a value of 0 will be returned. This command is useful if you want to check if a specific block is on or off and have your program react to the condition.

## CHARACTER GRAPHICS

The last type of graphics to discuss is character graphics. Character graphics allows one to take advantage of the

TRS-80's ability to print a complete graphics block as a single character. These are known as character strings. These appear in BASIC programs as CHR\$(X), with "X" being a variable. The available variables will depend upon what model computer you are using. For our example we use the MODELS I and III. In the Model I character set there are sixty-four graphic characters, made up of different combinations of lit pixels. There are also space compression codes on both the Models I and III. Space compression codes are a series of blank spaces from 0 to 63, which you can print as a single character string. The space compression codes range from CHR\$(192) to CHR\$(255) on both the Model I and III. The Model III has the same graphics character set as the Model I plus it's own special character set. This special set has such characters as the English pound sign, a happy face, a little man, a little woman, and many others. You can utilize any of these in your graphics program. To use this special set you must first turn them on. To do so simply type PRINTCHR\$(21) from BASIC. Once you have done this, use the same syntax as above and this will give you the special graphics characters. To turn off the special graphics use the PRINTCHR\$(21) again. If you notice PRINTCHR\$(21) works like an on and off switch, we call this a toggle.

With various combinations of the sixty-four graphics codes, you can do many of the same things as pixel graphics. Since you are printing six pixels at a time the character graphics will run much faster. Here is a program that will demonstrate the use of both pixel graphics and character graphics.

```

10 CLS
20 FOR Y = 1 TO 47
30 FOR X = 1 TO 127
40 SET(X,Y)
50 NEXT X
   : NEXT Y
60 CLS
70 FOR Y = 1 TO 1024
80 PRINT CHR$(191);
90 NEXT Y
100 PRINT @ 540, "THE END";
110 END

```

This program will first white out the screen using pixel graphics then do the same using character graphics. You can easily notice how much faster the character graphics were than the pixel graphics.

Using these ideas and sample programs you should be well on your way to writing impressive graphics programs.

## OVALS BACK INTO CIRCLES

If you have ever used a high-resolution graphics board to draw circles on the screen, only to find out that they turn into ovals on your dot matrix printer, then this program will give you a way to get "perfect" circles in hard copy.

A little explanation is due first. A graphics screen in good adjustment will give a perfect circle on the screen, setting whichever pixels (dots) that the CIRCLE command requires. A utility to transfer each pixel from the graphics screen to the pins of dot matrix print head can (and probably will) give distortion in the hard copy. The ratio of horizontal distance between pixels, to vertical distance between pixels on a graphics screen, is likely to be different from that same ratio of dots on the printer. So if the utility to "dump" the screen to the printer follows a direct pixel to dot correspondence, and the

ratios are different, the printout will be distorted if the screen is not, and vice versa.

The Model II graphics BASIC has one such utility called GPRINT. To get a "good" hard copy, it is necessary to "cheat" one way or another. With some DMP series printers, the printer's expanded mode may give reasonable results. In our experience though, the expanded mode improves but never perfects the circle. With some printers, the improvement is greater than with some others, because the ratio of expansion may be different.

The following Model II graphics BASIC program, is one way of getting perfect circles, no matter what printer you are using. The "cheat" in this program involves distorting the screen at the appropriate moment, so that when GPRINT distorts the printout, a perfect circle results.

```

1 REM ***** UP TO LINE 120 IS STANDARD SYNTAX.
   IT WILL DISTORT ON PRINT OUT TO A DIFFERENT
   PROPORTION ON DIFFERENT PRINTERS.
10 PRINT CHR$(2)
20 CLS2
30 LPRINT "ORIGINAL"
40 CIRCLE (320,120),100
50 LINE (300,0)-(340,20),,BF
60 LINE (0,0)-(639,239),,B
70 PRINT@(1,44), "<== This is a square"
80 PRINT@(1,1), "<== This is a rectangle"
90 PRINT@(12,53), "<== This is a circle"
100 SYSTEM"VDOGRPH"
110 SYSTEM"GPRINT"
120 CLS2
   : LPRINTCHR$(12)
121 REM ***** THE FOLLOWING IS SYNTAX TO GET AN
   UNDISTORTED PRINT-OUT
122 REM ***** EVERY TIME THE X (HORIZONTAL) AXIS
   IS ADDRESSED, WHETHER EXPLICITLY OR
   IMPLICITLY, IT IS MULTIPLIED BY A FACTOR
123 REM ***** FACTR=1 TO ADDRESS THE SCREEN
   FIRST TIME ROUND, FOR GOOD CIRCLES ON THE
   GRAPHICS DISPLAY
124 REM ***** FACTR=0.575 TO ADDRESS THE SCREEN
   THE SECOND TIME ROUND. THIS WILL DISTORT THE
   SCREEN BY AN AMOUNT WHICH WILL SUIT THE
   DISTORTION CAUSED BY A DMP-200
130 FACTR=1
140 CIRCLE (320*FACTR,120),100*FACTR,,,
   0.5*1/FACTR
150 LINE (300*FACTR,0)-(340*FACTR,20),,BF
160 LINE (0*FACTR,0)-(639*FACTR,239),,B
170 PRINT@(1,FACTR*44), "<== This is a square"
180 PRINT @(1,FACTR*1), "<== This is a rectangle"
190 PRINT @(12,FACTR*53), "<== This is a circle"
200 SYSTEM"VDOGRPH"
210 SYSTEM"GPRINT"
220 LPRINT "FACTOR=";FACTR
230 CLS2
   : LPRINTCHR$(12)
240 IF FACTR=1 THEN FACTR=0.575 ELSE INPUT "FACTOR
   TO BE SET TO 0 OR NEGATIVE TO EXIT,
   OTHERWISE TYPE IN NEW FACTR"; FACTR
250 IF FACTR <= 0 THEN END ELSE CLS2
   : GOTO140

```

## PROGRAM NOTES

1. The INPUT statement in line 240 will allow you to experiment with different factors for your printer.
2. Line 140 draws an upright ellipse, not a circle.
3. This is a Model II graphics BASIC program.
4. For other machines, the principles outlined above will work, but the syntax will vary.

5. When you find your perfect circle, it will now be "squashed" in two directions rather than just one.

### INTS AND TIPS

When writing graphics programs on a Model II to a line printer there are several things to remember.

1. Always use the TRSDOS command FORMS X, this will prevent unwanted interpretations by the operating system (TRSDOS).
2. Apply the following patch to TRSDOS version 02.00.00, to allow for CHR\$(0).

```
PATCH SYSRES/SYS A=10FC F=C8 C=00
```

After applying the patch reset the system.

3. Apply the following patches to prevent a carriage return after 132 characters on TRSDOS 01.02.00, and after 255 characters on TRSDOS 02.00.00.

```
PATCH BASIC A=3584 F=D4BA57 C=000000
```

```
PATCH BASIC A=35B8 F=D4BA57 C=000000
```

```
PATCH BASIC A=56F8 F=CC0D57 C=C30157
```

NOTE: This patch should be applied ONLY if a user is writing his own software. Under NO circumstances should this patch be applied for use by any Radio Shack software packages.

4. On a Model I to send out a CHR\$(0), rather than using LPRINT CHR\$(0), you must POKE 14312,0.
5. You cannot send out a CHR\$(10) from BASIC, regardless of the Model computer used.

### CONCLUSION

If you utilize these basic ideas and hints, you should be well on your way to having great success in the use of graphics on the TRS-80 microcomputers. Graphics can be used as a source of enjoyment or to greatly enhance the effectiveness of those vital business reports.



### Computer Customer Service Address and Phone Numbers

8AM to 5PM Central Time  
Computer Customer Services  
400 Atrium, One Tandy Center  
Fort Worth, Texas 76102

Productivity/Special Applications	(817)338-2390
Accounting Software	(817)338-2391
O/S and Languages, Group No. 1	(817)338-2392
O/S and Languages, Group No. 2	(817)338-2393
Hardware and Communications	(817)338-2394
Home Software	(817)338-2395
Educational Software	(817)338-2396
Newsletter Subscription Problems	(817)870-0407

A Wealth of Information for TRS-80 Computer Users

## TRS-80 MICROCOMPUTER NEWS REPRINTS



### SPECIAL PACKAGE SALE

3 VOLUME SET  
FOR JUST

**\$13.95**

A SAVINGS  
OF \$13.90

Volume I	Prior to January 1981
Volume II	1981 Reprints
Volume III	1982 Reprints

Cat. No.	Original Price
26-2115	\$ 4.95
26-2240	9.95
26-2241	12.95
	<u>27.85</u>

Sale Price	<u>13.95</u>
Savings	13.90

Sale ends June 24, 1984

# Orchestra-90's Special Composer Commands

By Bryan Eggers  
Software Affair, Ltd.

The Orchestra-90™ stereo music synthesizer for Model III/4 (26-1922) has three special commands which are extremely useful during music transcription and composition. A brief description of these commands appears in the Orchestra-90 manual under "Special Command Line Functions". This month I'll explain the commands in detail and offer a few subtle variations you might find useful.

The commands are invoked by entering a single character, **[F1]**, **[F2]** or **[F3]**, on the Command Line. Two of the commands, **[F1]** and **[F3]**, work in conjunction with the current cursor position in the music file. The third actually supplies the cursor position. The commands are line-oriented, so the actual position of the cursor WITHIN the line is irrelevant.

In general, these commands allow you to:

- [F1]** Play your music file from the current cursor position.
- [F2]** Locate the position in the file where "playing" was stopped.
- [F3]** Display the current Voice/Register assignments anywhere in the file.

Make sure you are in Command Mode (cursor at upper left corner of screen) when you invoke these commands.

## PLAY FROM CURSOR POSITION - **[F1]**

This command is probably the most powerful of the Special Composer Commands. I use it dozens of times during each file transcription. It's a combination SCORE/PLAY command from cursor location. Sometimes called the "quick compile", it reduces compile/play time by ignoring all notes prior to the cursor position during the compile. When the compile is finished, the command then automatically PLAYS the music file from the cursor position.

All Voice/Registers assignments and transpositions are accumulated and considered during this compile, so the music plays as if the ENTIRE file were compiled, yet the result is an almost immediate play from the cursor position. After the **[F1]** command, **[P] [ENTER]** will replay the same section.

The music file plays from the cursor position to the end of the music file, or until you press **[O]** to stop the music. It's particularly useful for listening to the most recently entered measure of music, without having to listen to the entire file.

You can preview each Voice of music immediately after you enter it, or the last complete Measure, or the last several Measures.

The position of your cursor is critical if positioned WITHIN a measure. For example, let's examine the effect of various

cursor positions using the **[F1]** command to preview these two measures from a Bach Fugue:

```
P01 K2&
M01 *S6756I4;6;Q7S$767      (Line A)
    V2S1201I-1;1;Q2
    V3Q$S4
    @V4Q3S$ABACBDCIE;C;
M02 *S5645I3;5;Q6S$656      (Line B)
    V2S01-10I-2;0;Q1S$101    (Line C)
    V3Q$S3                    (Line D)
    @V4QBS$4546576I8;6;      (Line E)
```

Press **[SHIFT]** and **[BREAK]** to invoke ORCH-90's EDIT mode. Type in the example as shown, then move the cursor back up to Line A and press **[BREAK]** to return to Command mode. Type:

**[F1] [ENTER]**

A compile occurs, during which the compiler checks for various types of errors. If an error is found, the cursor is positioned as close as possible to the source of the error. Just press **[SHIFT]** and **[BREAK]** to return to EDIT mode, fix the problem, and press **[BREAK]** to return to Command mode.

If no errors were found, you'll hear the first complete Measure (M01) playing in 4-part harmony, all 4 Voices playing simultaneously, followed by Measure 02, also in 4-part harmony. In other words, the entire file played normally from the beginning—the same result we'd get by using the regular SCORE and PLAY commands. Now let's move on to some really useful variations!

Return to Edit Mode by pressing **[SHIFT]** and **[BREAK]**, then move the cursor to Line B. Press **[BREAK]** and type:

**[F1] [ENTER]**

This time the music starts playing at Measure 02. The notes in Measure 01 were prior to Line B and therefore, ignored during compile.

Using the **[F1]** command with the cursor on Line C also plays Measure 02, except Voice 1 is silent! It's notes were ignored during the compile. Using the **[F1]** command with the cursor on Line D produces a play of Measure 02, except that Voices 1 AND 2 are silent. You hear only Voices 3 and 4. And finally, with the cursor on Line E, the **[F1]** command plays only Voice 4. In the examples, any music following Measure 02 would be played next.

Not only are these tricks handy for debugging a problem Measure, they provide an interesting way to study Voices and harmony within a Measure.

I'd better explain how REPEATS are handled, since they pose a minor problem if the original Part repeated exists prior to the cursor location. This variation of the previous example, with "P02" and "R01" added, will demonstrate the situation:

```
P01 K2&
M01 *S6756I4;6;Q7S$767      (Line A)
  V2S1201I-1;1;Q2
  V3Q$S4
  @V4Q3S$ABACBDCIE;C;
P02
M02 *S5645I3;5;Q6S$656      (Line B)
  V2S01-10I-2;0;Q1S$101      (Line C)
  V3Q$S3                       (Line D)
  @V4QBS$4546576I8;6;        (Line E)
R01
```

If we do a quick compile with the cursor on any of Lines B through E, the R01 Repeat produces no sound. With the example SCORED normally, R01 would Repeat the notes contained in Part 01, which now contains only Measure 01 because of the insertion of "P02". However, since the **[T]** compile was invoked with the cursor completely PAST Part 01, the notes in Part 01 weren't added to the compiled code, and R01 repeats the "empty" Part 01 with no duration and no sound. Keep this in mind when you use the **[T]** command. Don't let "empty" Repeats confuse you.

If you get into the habit of previewing each Measure as you enter it, you'll catch most of your transcription errors as they occur. I'm convinced this is the best technique, but there are some arrangers who prefer to type in the entire music file before listen to it. Tracking down a "sour note" after finishing the transcription is more difficult, but ORCH-90 has another Special Composer command to help you do it! This command is appropriately nicknamed:

#### WHERE IT'S AT - @

If you hear a "sour note" in your file, it's easy to refer to the original sheet music to locate the problem measure. But, suppose a friend sent you the file via modem, hoping you can fix a "sour note" in it. Or, perhaps you PACKed the file with the ORCHUTIL program and you no longer have Measure strings to refer to. Maybe the file was written without the benefit of sheet music - many of the ORCH-90 arrangers can do it! In any event, you heard a "sour note" but you're not sure where the corresponding code is in the music file.

The **@** command can't find the actual "sour note" for you. After all, ORCH-90 doesn't know a "sour note" from a good note - it just plays what you tell it to. But, if you play the music file and stop it (by pressing **[O]**) when you hear the problem, ORCH-90 can locate the corresponding code in your music language file. The procedure is very simple: SCORE (compile) the file normally by typing:

**[S] [ENTER]**

Then, type:

**[P] [ENTER]**

to play the file normally. The instant you hear the "sour note" or passage you want to change, stop the music by pressing **[O]**.

You're already in Command Mode, so type:

**@ [ENTER]**

In a couple of seconds you'll see the music file positioned with the cursor as close to the stopping point as possible. Technically, the cursor always appears at the beginning of the Measure, Part or Repeat FOLLOWING the actual code being played when you pressed **[O]**.

Now that you're in the approximate area of the file, use the **[T]** command to locate the EXACT problem Measure. Do this by scrolling the file backwards and doing a few "quick compiles" until you find the actual Measure containing the "sour note". Then, go into EDIT mode and start fixing the problem, continually testing your changes by playing the Measure with **[T]**.

You might get a bit confused if you stop the music DURING a Repeat, because the **@** command will position the cursor in the Measure, Part or Repeat following the ORIGINAL Part specified by that Repeat. Try to stop the music as close to the "sour note" as possible to avoid this. Another approach would be to stop the music just before the "sour note" plays, then start searching forward in the file until you find it.

This command is simple to use and fairly easy to describe, but it was very difficult to implement. The synthesizer routine is being interrupted as it's playing COMPILED code, not the text file you created. The author of Orchestra-90, Jon Bokelman, spent a considerable amount of time developing all three Special Composer commands.

Use the **@** command only after a file has been SCORED and PLAYED normally, not after a **[T]** compile command has been used. Also, the **@** command generates a special compile to determine the "stopping" location, therefore the file must be recompiled before it can be played normally. Use Either the SCORE or **[T]** command.

#### DISPLAYING VOICE/REGISTER STATUS - ?

This command displays the Register assignments and Voice transposition currently in effect at any point in the file. Let's review the flexibility of the music language which allows these parameters to be changed so often.

Orchestra-90 allows instrument Registers to be assigned to Voices at Part boundaries. Voices may be transposed at Measure boundaries. For example, if you wanted all four Voices to play the next Measure using TRUMPET sounds, you'd assign the A Register (trumpet) to all Voices by inserting the new assignments on a Part number just prior to the Measure where you want the changes to occur:

```
P10 V1YA V2YA V3YA V4YA
M37 *IB"B;QA#BI$S89%
  V2Q.5"15;4;5;Q6
  @V3S0+110210+110213210
  V4H$Q$I$3;
```

Similarly, Voices can be TRANSPOSED at Measure boundaries. For example:

```
M65 V1U-7 V2U-7 V3U-E V4U+7
  *IA;B;C;D;QCSCBA9%
  V2I5;7#;8;A;9%;8;7#;8
  V3I1;2%;S3434I5;2%;Q3
  @V4S435%4I6;+1;S0#+11%0I2;S+12
```

The symbols on the first line instruct the system to play Voices 1 and 2 one octave lower than the notation indicates. The "-7" means "down 7 steps". Voice 3 will be played two octaves lower than written (down 14 steps). And, Voice 4 will be played one octave HIGHER than written (UP 7 steps).

Let's say you're transcribing a music file and you've changed Register and transposition a few dozen times throughout the piece. After entering a few more Measures, you suddenly scratch your head and think, "Which instrument Register is currently assigned to Voice 3?"

Fortunately, there is an easy solution. You can determine which Registers are currently assigned to the Voices, and the amount each Voice is transposed, using the **[?]** command. Position your cursor on any line in the music file, press **[BREAK]** to return to Command mode, then type:

**[?] [ENTER]**

The second line of the screen (the STATUS line) displays the resulting information in the form:

```
V1YAU+7 V2YCU-7 V3YDU+E V4YAU+0 V5YCU+7
```

These symbols represent the settings for Voices 1-5 (in the 5-voice synthesizer mode). The letter after the "Y" in each group is the name of the Register (A, B, C, D or E) currently assigned to the Voice. The two symbols after the U indicate the direction (+ or -) of the transposition and the number of steps.

I hope this information helps you make better use of the Special Composer commands. The immediate feedback provided by these commands might inspire you to "enhance" some of the simpler sheet music arrangements.



With practice, you can learn to take advantage of Orchestra-90's full capabilities. In time, you may be able to dispense with sheet music altogether and write ORCH-90 arrangements "from scratch". It's not very difficult, and it's really a lot of fun! Even if you're "all thumbs" or can't read a note of music, you can program ORCH-90 to play musical arrangements far more complex than any human being could ever play!

We have a few hundred free music files on our ORCH-90 SIG on CompuServe™ on page HOM-13. Study these and you'll pick up all kinds of ideas. You can leave messages to other users, share "special effects", programming techniques, and even upload your own music files to the database. The SIG exists to support Orchestra-90 and related products, so visit us soon!

"Orchestra-90" is a registered trademark of Software Affair, Ltd.

"CompuServe" is a registered trademark of CompuServe, Inc. 

## Computer Clubs

COCO MUG  
c/o Wayne Johnson  
900 Rock Hill Road  
Jefferson City, MO 65101  
314-893-2789

DALTRUG TRS-80 USERS GROUP  
c/o D. D. Freeman, Pres.  
334 Fieldside Drive  
Garland, TX 75043  
214-270-0625

DEARBORN TRS-80 USERS GROUP  
P.O. Box 942  
Dearborn, MI 48121

FORT WORTH TRS-80 USERS GROUP  
c/o Pat Coyne, Pres.  
2001 Beach Street #626  
Fort Worth, TX 76103  
817-429-7055

MID-CITIES TRS-80 USERS GROUP  
c/o Charles Knight, Pres.  
2708 Roberts Circle  
Arlington, TX 76010  
817-640-4452

NORTHERN VIRGINIA COLOR COMPUTER CLUB  
Office of the Secretary  
6048 Belleview Drive, Apt.#1  
Falls Church, VA 22041

SANTA CLARITA COLOR COMPUTER CLUB  
c/o David Barry, Jr.  
23920 Fambrough Street  
Newhall, LA 91321 

# Pixels in Circular Motion

John R. Flitton  
23 Statesman Square  
Agincourt, Ontario  
Canada M1S 4H7

Here is a machine language subroutine which some readers could possibly incorporate into their Model III BASIC games. It turns all the pixels on using a circular motion.

```

10 POKE16526,48
   : POKE16527,117
20 RESTORE
30 FOR Z = 30000 TO 30270
   : READ Y
   : POKE Z,Y
   : NEXT
40 X =USR(0)
50 END
60 DATA 33,58,118,54,1,33,56,118,54,64,35,54,47,
   205,137,117,33,56,118,126
70 DATA 254,0,40,4,53,195,61,117,205,137,117,33,
   57,118,126,254,0,40,4,53
80 DATA 195,76,117,205,137,117,33,56,118,126,254,
   127,40,4,52,195,91,117,205,137
90 DATA 117,33,57,118,126,254,47,40,4,52,195,106,
   117,205,137,117,33,56,118,126
100 DATA 254,65,40,4,53,195,121,117,201,221,
   33,54,118,221,102,2,221,110,3,229
110 DATA 221,86,0,221,94,1,213,124,38,1,146,48,
   4,38,255,237,68,71,125,46
120 DATA 1,147,48,4,46,255,237,68,79,17,255,
   255,19,184,56,3,72,71,235,221
130 DATA 116,0,221,115,1,221,114,2,221,117,3,
   104,97,72,203,57,4,209,213,229
140 DATA 197,38,255,123,36,214,3,48,251,198,3,71,
   106,203,37,203,44,203,29,203
150 DATA 44,203,29,203,16,4,124,230,3,246,60,103,
   175,55,143,16,253,203,126,32
160 DATA 2,54,128,221,203,4,70,40,3,182,24,2,47,
   166,119,193,225,122,221,134
170 DATA 0,87,123,221,134,1,95,121,132,79,189,
   56,12,149,79,122,221,134,2,87
180 DATA 123,221,134,3,95,16,168,209,221,114,0,
   221,115,1,225,221,116,2,221,117
190 DATA 3,201,64,24,0,0,0,0,0,0

```

```

120 K1 = 5900
130 CLS
140 GOTO 40
150 REM SPEEDUP SUBROUTINE
160 FOR A1 = 1 TO 127 STEP 10
170 B1 = K1/A1
180 IF B1 > 47 THEN AA = A1
   : GOTO 200
190 A1 = 127
200 NEXT A1
210 RETURN

```

# Amazing Random Mazes

Scott Cheatham  
Radio Shack #01-3470  
15422 North 99th Avenue  
Sun City, AZ 85351

This program's function is to design random mazes. There are two versions, a screen version, which will run on a 16K Model III and display using Model III graphics, and a printer version set for a 48K Model III.

The computer has to think a lot to create a maze, from eight to ten minutes for the screen version and from a few minutes to many days on the printed version, depending on maze dimensions.

Maze width and length are determined by memory capacity, array dimension, and user determined size, in that order. For systems with less memory, adjust the value of variable L5 downward until "out of memory" terminations stop. If the Printer being used has greater width, increase array width and decrease L5 accordingly. To determine proper width use the following formula:

$$W = \text{INT}((P-1)/3)$$

W = maze width

P = number characters per printer line

Each maze is made up of "Blocks", each block is one array element and on the printer is three characters wide and two lines down. Mazes can be designed to any length (depending upon memory capacity). The length you choose is determined only by the amount of paper you wish to waste and the number of days you can stand keeping your hands off the keyboard. Good Luck! And have fun solving them.

## Screen Program

```

110 CLS
   : W = 63
   : L = 23
   : DIMM(L,W)
   : C1 = W * L
   : C = 2
   : B = RND(W)
   : M(1,B) = 1
   : A = B - 1
   : D = 1
120 IF A <> W THEN A = A + 1
   : GOTO 150
130 IF D <> L THEN A = 1
   : D = D + 1
   : GOTO 150
140 A = 1
   : D = 1
150 IF INT(M(D,A)) = 0 THEN 120
160 IF A <> W IF INT(M(D,A + 1)) = 0 THEN 210

```

# Pattern

Scott Shepard  
P.O. Box 397  
Gulfport, MI 39501

This program makes an interesting pattern on the screen of your Model III. Pressing any key draws a new pattern.

```

10 CLS
20 K1 = 5900
30 K2 = 10
40 GOSUB 150
50 FOR A = AA TO 127 STEP 2
60 B = K1/A
70 IF B < 47 THEN IF B >= 0 THEN SET (A,B)
   : SET (A-1,B)
80 NEXT A
90 K1 = K1-K2
100 Z$ = INKEY$
   : IF Z$ = "" THEN GOTO 40
110 K2 = K2+10

```

```

170 IF A <> 1 IF INT(M(D,A - 1)) = 0 THEN 210
180 IF D <> L IF INT(M(D + 1,A)) = 0 THEN 210
190 IF D <> 1 IF INT(M(D - 1,A)) = 0 THEN 210
200 GOTO 120
210 ON RND(4) GOTO 220 ,240 ,260 ,280
220 IF A = 1 THEN 210 ELSE IF M(D,A - 1) <> 0
    THEN 210
230 M(D,A - 1) = C + 0.2
    : A = A - 1
    : C = C + 1
    : IF C > C1 THEN 300 ELSE 150
240 IF D = 1 THEN 210 ELSE IF M(D - 1,A) <> 0
    THEN 210
250 M(D - 1,A) = C + 0.1
    : D = D - 1
    : C = C + 1
    : IF C > C1 THEN 300 ELSE 150
260 IF D = L THEN 210 ELSE IF M(D + 1,A) <> 0
    THEN 210
270 M(D,A) = M(D,A) + 0.1
    : D = D + 1
    : M(D,A) = C
    : C = C + 1
    : IF C > C1 THEN 300 ELSE 150
280 IF A = W THEN 210 ELSE IF M(D,A + 1) <> 0
    THEN 210
290 M(D,A) = M(D,A) + 0.2
    : A = A + 1
    : M(D,A) = C
    : C = C + 1
    : IF C > C1 THEN 300 ELSE 150
300 FOR K = 15360 TO 16383
    : POKEK,191
    : NEXT K
    : E = RND(W)
    : M(L,E) = M(L,E) + 0.1
310 FOR K1 = 1 TO 45 STEP 2
    : FOR K = 1 TO 125 STEP 2
    : RESET(K,K1)
320 NEXT K
    : NEXT K1
    : RESET(B * 2 - 1,0)
    : FOR K1 = 1 TO L
    : FOR K = 1 TO W
330 K2 = INT((M(K1,K) - INT(M(K1,K))) * 10 + 1.1)
    : IF K2 > 2 RESET(K * 2, K1 * 2 - 1)
340 K2 = K2/2
    : IF K2 = INT(K2) RESET(K * 2 - 1, K1 * 2)
350 NEXT K
    : NEXT K1
    : FOR K = 0 TO 127
    : RESET(K,47)
    : NEXT K
360 GOTO 360

```

#### Printer Program

```

110 RANDOM
    : L5 = 250
    : DIM M(L5 + 10,31)
    : INPUT "WIDTH , LENGTH ";W,L1
120 CLS
    : B = RND(W)
    : E = RND(W)
    : Q = 0
130 IF L1 < L5 + 1 THEN L = L1
    : L1 = 0
    : GOTO 150
140 L = L5 + 10
    : L1 = L1 - L5
150 K1 = 0
    : K = 0
    : FOR K1 = 1 TO L
    : FOR K = 1 TO W
    : M(K1,K) = 0
    : NEXT K
    : NEXT K1

```

```

160 M(1,B) = 1
    : C = 2
    : C1 = W * L
    : A = B
    : D = 1
    : GOTO 200
170 IF A <> W THEN A = A + 1
    : GOTO 200
180 IF D <> L THEN A = 1
    : D = D + 1
    : GOTO 200
190 A = 1
    : D = 1
200 IF INT(M(D,A)) = 0 THEN 170
210 IF A <> W IF INT(M(D,A + 1)) = 0 THEN 260
220 IF A <> 1 IF INT(M(D,A - 1)) = 0 THEN 260
230 IF D <> L IF INT(M(D + 1,A)) = 0 THEN 260
240 IF D <> 1 IF INT(M(D - 1,A)) = 0 THEN 260
250 GOTO 170
260 ON RND(4) GOTO 270 ,290 ,310 ,330
270 IF A = 1 THEN 260 ELSE IF M(D,A - 1) <> 0
    THEN 260
280 M(D,A - 1) = C + 0.2
    : A = A - 1
    : C = C + 1
    : IF C > C1 THEN 350 ELSE 200
290 IF D = 1 THEN 260 ELSE IF M(D - 1,A) <> 0
    THEN 260
300 M(D - 1,A) = C + 0.1
    : D = D - 1
    : C = C + 1
    : IF C > C1 THEN 350 ELSE 200
310 IF D = L THEN 260 ELSE IF M(D + 1,A) <> 0
    THEN 260
320 M(D,A) = M(D,A) + 0.1
    : D = D + 1
    : M(D,A) = C
    : C = C + 1
    : IF C > C1 THEN 350 ELSE 200
330 IF A = W THEN 260 ELSE IF M(D,A + 1) <> 0
    THEN 260
340 M(D,A) = M(D,A) + 0.2
    : A = A + 1
    : M(D,A) = C
    : C = C + 1
    : IF C > C1 THEN 350 ELSE 200
350 M(L,E) = M(L,E) + 0.1
    : IF Q = 1 THEN 390
360 LPRINT TAB(3 * (B - 1));"START"
370 FOR K = 1 TO W
    : IF K = B LPRINT ": ";ELSE LPRINT ":-";
380 NEXT K
    : LPRINT ":"
    : Q = 1
390 FOR K1 = 1 TO L
400 IF K1 > L5 AND L1 > 0 THEN 480
410 LPRINT "I";
    : FOR K = 1 TO W
420 IF INT((M(K1,K) - INT(M(K1,K))) * 10 + 1.1) >
    2 LPRINT " ";ELSE LPRINT " I";
430 NEXT K
    : LPRINT
440 FOR K = 1 TO W
    : K2 = INT((M(K1,K) - INT(M(K1,K))) * 10 +
    1.1)/2
450 IF K2 = INT(K2) LPRINT ": ";ELSE LPRINT
    ":-";
460 NEXT K
    : LPRINT ":"
    : NEXT K1
470 LPRINT TAB(3 * (E - 1));"FINISH"
    : END
480 B = RND(W)
    : K2 = INT((M(L5,B) - INT(M(L5,B))) * 10 +
    1.1)
490 IF K2/2 = INT(K2/2) THEN 130 ELSE 480

```

# Weaver

Rejean Giasson  
P.O. Box 361  
Ville de Saint-Georges  
Beauce, P. Quebec  
Canada G5Y 5C8

Here is a short program for Model I or III, named WEAVER. I hope that you will appreciate the figures created by the computer. The weavers will like it particularly. In this program, there is a possibility of using mathematical demonstration of code ASCII 128 to 191 (see line 90 to 200).

```
6 PRINT@960,"<ENTER>";
  : INPUTA$
  : CLS
  : FOR X=0 TO 300
  : NEXT X
10 '      THE WEAVER
      PROGRAM CREATED BY

      REJEAN GIASSON
      P.O. BOX 361
      SAINT-GEORGES BEAUCE, CANADA
      G5Y 5C8
15 CLS
  : FOR X=1 TO 10
20 PRINT@474, "PRESS A KEY"; @535, "FOR A NEW
  FIGURE"
22 FOR Y=1 TO 75
  : NEXT Y
  : CLS
  : FOR Y=1 TO 25
  : NEXT Y
24 IF INKEY$ <>"" THEN 50
30 NEXT X
40 '      *** PREPARATION ***
50 CLEAR 1000
  : DIM A(12), B(12), C(12), D(12), F(12)
60 CLS
  : CA=RND(5) + 1
  : FOR X=1 TO 7
  : F(X)=RND(CA)
  : NEXT X
70 FOR X=8 TO 12
  : F(X)=F(14-X)
  : NEXT X
80 CH(1)=3
  : CH(2)=12
  : CH(3)=48
90 FOR X=1 TO 12
100 FOR Y=1 TO 3
110 IF F(Y)=F(X) THEN A(X)=A(X) + CH(Y)
120 IF F(Y + 3)=F(X) THEN B(X)=B(X) + CH(Y)
130 IF F(Y + 6)=F(X) THEN C(X)=C(X) + CH(Y)
140 IF F(Y + 9)=F(X) THEN D(X)=D(X) + CH(Y)
150 NEXT Y
160 A$=A$ + CHR$(A(X) + 128)
  : PRINT@0, A$;
170 B$=B$ + CHR$(B(X) + 128)
  : PRINT@64, B$;
180 C$=C$ + CHR$(C(X) + 128)
  : PRINT@128, C$;
190 D$=D$ + CHR$(D(X) + 128)
  : PRINT@192, D$;
200 NEXT X
210 '      *** WEAVING ***
220 FOR X=1 TO 4
230 FOR Y=1 TO 5
240 PRINT@Q, A$; @Q + 64, B$; @Q + 128, C$; @Q +
  192, D$;
  : Q=Q + 12
```

```
250 NEXT Y
260 PRINT@Q, CHR$(A(1) + 128); @Q + 64, CHR$(B(1)
  + 128); @Q + 128, CHR$(C(1) + 128); @Q +
  192, CHR$(D(1) + 128);
  : Q=Q + 196
270 NEXT X
290 IF INKEY$="" THEN 290 ELSE 50
```

## Printer Graphics Generator

David Whitney  
400 South West 99th Street  
Oklahoma City, OK 73139

One of the best features of the DMP series printers is the Bit Image Graphics capability, whereby finely detailed graphic pictures can be created by use of high-density graphics mode. For those who have dabbled in this creative process before, the procedure for preparing such pictures is familiar. Break down your picture on graph paper into rows seven dots high, then convert each vertical column into its equivalent CHR\$ code which is issued to the printer.

The individual dots in each column are numbered 1 to 64 in powers of two. By adding the desired values of "on" dots to 128, the desired code is achieved. In a picture with a great deal of data, errors in addition are very easy to make. When I attempted a very large-scale graphic drawing on my DMP-200, I decided I would let my Model 4 do the tedious detail work for me by writing a program for the creation and printing of printer graphics. The Printer Graphics Generator was therefore born.

The basic operation of the program is simple; the program creates on disk a sequential file of numbers corresponding to the data values in your picture. The graphics codes are entered simply by selecting the proper numbers in the bit image; now in numerical order instead of powers of 2. When the desired image is achieved, you can proceed to the <N>ext Pattern, <C>hange this pattern if it is incorrect, enter a <R>epeat value if this pattern is to be printed several times in succession, <S>top this line, or <Q>uit the picture altogether.

The program also offers the ability to change printer dot density in the graphics mode. The current density is always displayed on the screen during file creation and on the Main Menu. You can also print the picture, and obtain a print of file values.

After your picture is complete, or if you wish to make changes after it is complete, the program also can create a BASIC program from the data file values. The program creates an ASCII data file, which is an executable BASIC program. The Rectangle Program below was generated using the example rectangle sequential file.

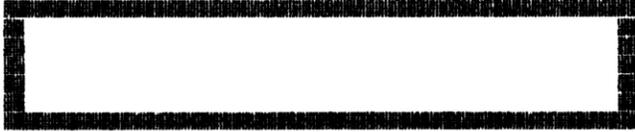
### Rectangle Program

```
10 CLS
  : LPRINT CHR$(18);
20 READ NUM
30 IF NUM = 99999 THEN LPRINT CHR$(30)
  : END
40 IF NUM < 0 THEN LPRINT CHR$(28);CHR$(-NUM);
  : GOTO 20
```

```

50 IF NUM = 999 THEN LPRINT
   : GOTO 20
60 LPRINT CHR$(NUM);
   : GOTO 20
70 DATA -200,255,0,999,0,-7,255,0,-186,128,0,-7,
        255,0,999,0,-7,255,0,-186,128,0,-7,255,0,999,
        0,-7,255,0,-186,128,0,-7,255,0,999,0,-7,255,0
        ,-186,128,0,-7,255,0,999,0,-7,255,0,-186,128,
        0,-7,255,0,999,0,-200,255,0,999,0,
71 DATA 99999

```



To generate the sequential file for the example rectangle, OPEN a new file and use the following input values. The input value "1234567" generates a CHR\$(255).

Line	Dot Position	Input
1	1	1234567
		R
		200
1	200	S
2	0	N
2	1	1234567
		R
		7
2	7	N
2	8	<input type="text" value="ENTER"/>
		R
		186
2	193	N
2	194	1234567
		R
		7
2	200	S
3-6		Same as Line 2
7	0	N
7	1	1234567
		R
		200
7	200	S
		Q

The Printer Graphics Generator program was written in Model 4 BASIC, but can readily be adapted to any other version of BASIC with some minor changes. I hope that everyone finds it as useful and helpful as I do in creating finely detailed and impressive graphics on the DMP-200, and on most other DMP printers.

#### Printer Graphics Generator Program

```

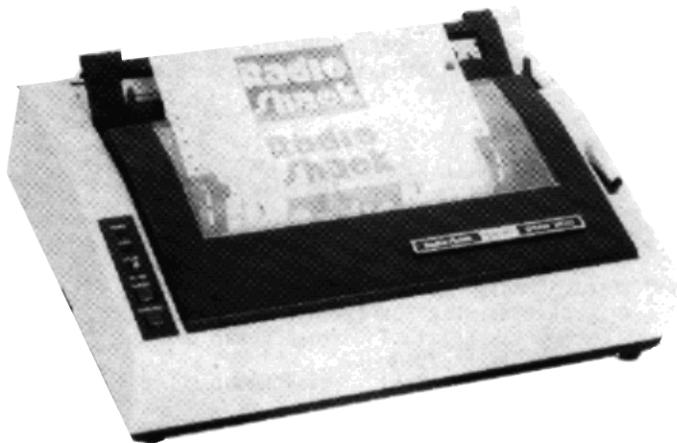
10 *****
20 '*      Printer Graphics Generator      *
25 '*      Model 4 64K 2-disk, program by  *
30 '*      David Whitney, 12/20/83        *
40 '*      For use with DMP printers with  *
45 '*      bit image graphic ability.     *
50 *****
60 LIN = 0
   : DOT = 0
   : DDEN = 479
   : LPRINT CHR$(30);CHR$(27);CHR$(19);CHR$(18)
70 CLS
   : PRINT@(5,27), "Printer Graphics Generator"
   : PRINT@(7,38), "Menu:"

```

```

80 PRINT@(9,23), "1. Open New/Existing Picture
   file"
90 PRINT@(10,23), "2. Print an existing file
   picture"
100 PRINT@(11,23), "3. Print listing of data
   file values"
110 PRINT@(12,23), "4. Change printer dot
   density"
120 PRINT@(13,23), "5. Close files and exit"
130 PRINT@(14,23), "6. Create Basic program from
   file"
140 PRINT@(16,27), "Current Dot Density: ";DDEN
150 PRINT@(18,30), "Select option (1-6)";
160 Y$ = INKEY$
   : IF Y$ = "" THEN 160
170 OP = VAL(Y$)
   : IF OP<1 OR OP>6 THEN 160
180 CLS
190 ON OP GOTO 200,470,570,660,780,790
200 IF FILE$<>"" THEN OPEN"O",1,FILE$+"/dat:1"
   : PRINT#1,LIN,DOT
   : CLOSE
210 CLS
   : PRINT@(11,28), "Enter File Name >";
   : LINE INPUT FILE$
   : LIN = 1
   : DOT = 0
   : CLS
220 OPEN "O",1,FILE$+":1"
   : IF LOF(1)>0 THEN CLOSE 1
   : OPEN"E",1,FILE$+"
   : 1"
   : OPEN"I",2,FILE$+"/dat:1"
   : INPUT #2,LIN,DOT
   : CLOSE 2
   : CLS
230 PRINT CHR$(15);
   : REP = 1
   : TOT = 0
   : DOT = DOT+1
240 PRINT@(13,10),SPC(25);
   : PRINT@(14,10),SPC(25);
250 PRINT@(6,27), "Preparing dot matrix codes";
   : PRINT@(18,0),SPC(79);
260 GOSUB 980
270 FOR X = 1 TO 7
   : PRINT @(9+X,38),X;"> ";
   : NEXT X
280 PRINT CHR$(14);
   : PRINT@(18,30), "Select ON bits (1-7)";
290 Y$ = INKEY$
   : IF Y$ = "" THEN 290
300 PRINT CHR$(15);
   : IF Y$ = CHR$(13) THEN 350
310 W = VAL(Y$)
   : IF W<1 OR W>7 THEN 290 ELSE IF L(W) = 1
   THEN 290 ELSE L(W) = 1
320 TOT = TOT+2^(W-1)
   : PRINT@(9+W,42), "0";
   : GOTO 280
330 GOTO 320
340 NEXT X
350 FOR L = 1 TO 7
   : L(L) = 0
   : NEXT
   : BITS = TOT+128
   : PRINT@(13,10),USING"Pattern Value : ##";
   BITS;
   : REP = 1
360 GOSUB 980
370 GOSUB 970
380 PRINT CHR$(14);
   : Y$ = INKEY$
   : IF Y$ = "" THEN 380
390 IF Y$ = "N" OR Y$ = "n" THEN GOSUB 450
   : GOTO 230

```



```

400 IF Y$ = "C" OR Y$ = "c" THEN DOT = DOT-REP
   : GOTO 230
410 REP = 1
   : IF Y$ = "R" OR Y$ = "r" THEN
   PRINT@(19,27), "Enter repetition value >";
   : LINE INPUT REP$
   : REP = VAL(REP$)
   : PRINT@(19,27), SPC(27);
   : PRINT@(14,10), USING "Repeat Value :
   ###"; REP;
   : DOT = DOT+REP-1
   : GOSUB 450
   : GOTO 360
420 IF Y$ = "S" OR Y$ = "s" THEN GOSUB 450
   : PRINT#1, 999
   : LIN = LIN+1
   : DOT = 0
   : GOTO 360
430 IF Y$ = "Q" OR Y$ = "q" THEN GOSUB 450
   : OPEN "O", 2, FILE$+"dat:1"
   : PRINT#2, LIN, DOT
   : CLOSE
   : GOTO 70
440 GOTO 380
450 IF REP <> 1 THEN PRINT#1, -REP
460 PRINT#1, BITS
   : REP = 1
   : BITS = 0
   : RETURN
470 CLOSE
   : CLS
   : PRINT@(11,28), "Enter File Name >";
   : LINE INPUT FILE$
480 LPRINT CHR$(18)
   : OPEN "I", 1, FILE$+":1"
490 WHILE NOT EOF(1)
500 INPUT #1, NUM
510 IF NUM = 999 THEN LPRINT
   : GOTO 540
520 IF NUM < 0 THEN LPRINT CHR$(28); CHR$(-NUM);
   : GOTO 540
530 LPRINT CHR$(NUM);
540 WEND
550 CLOSE
560 GOTO 70
570 CLS
   : PRINT@(11,28), "Enter File Name >";
   : LINE INPUT FILE$
580 OPEN "I", 1, FILE$+":1"
   : LPRINT CHR$(30)
   : LPRINT
590 LPRINT "Data for line # 1"
   : LINENUM = 1
600 WHILE NOT EOF(1)
610 INPUT #1, NUM

```

```

620 IF NUM => 999 THEN LINENUM = LINENUM+1
   : LPRINT
   : LPRINT
   : LPRINT "Data for line #"; LINENUM
   : GOTO 640
630 LPRINT USING "###"; NUM;
640 WEND
645 CLOSE
650 GOTO 70
660 CLS
   : PRINT@(7,28), "Current Dot Density:
   "; DDEN;
670 PRINT@(9,28), "Change to one of these:";
680 PRINT@(11,26), "1. Style #1: 479 dots/line";
690 PRINT@(12,26), "2. Style #2: 575 dots/line";
700 PRINT@(13,26), "3. Style #3: 799 dots/line";
710 PRINT@(15,26), "Select Style Option (1-3)";
720 Y$ = INKEY$
   : IF Y$ = "" THEN 720
730 O = VAL(Y$)
740 IF O = 1 THEN LPRINT CHR$(30); CHR$(27);
   CHR$(19); CHR$(18)
   : DDEN = 479
750 IF O = 2 THEN LPRINT CHR$(30); CHR$(27);
   CHR$(21); CHR$(18)
   : DDEN = 575
760 IF O = 3 THEN LPRINT CHR$(30); CHR$(27);
   CHR$(20); CHR$(18)
   : DDEN = 799
770 GOTO 70
780 END
790 CLS
   : PRINT@(11,28), "Creating Program for >";
   : LINE INPUT FILE$
800 CLOSE
   : OPEN "I", 1, FILE$+":1"
   : OPEN "O", 2, FILE$+"/pro:1,A"
810 PRINT@(13,10), "Creating ASCII Basic program
   file under name: "; CHR$(34); FILE$; "/pro:1"
   CHR$(34)
820 PRINT#2, "10 cls:lprint chr$(18);"
830 PRINT#2, "20 read num"
840 PRINT#2, "30 if num = 99999 then lprint
   chr$(30):end"
850 PRINT#2, "40 if num < 0 then lprint
   chr$(28); chr$(-num); goto 20"
860 PRINT#2, "50 if num = 999 then lprint:goto
   20"
870 PRINT#2, "60 lprint chr$(num);:goto 20"
880 LNUM = 70
890 DA$ = RIGHT$(STR$(LNUM), LEN(STR$(LNUM))-1) + "
   DATA "
900 WHILE NOT EOF(1)
   : INPUT #1, NUM
   : IF NUM < 0 THEN DA$ = DA$+STR$(NUM) ELSE DA$
   = DA$+RIGHT$(STR$(NUM), LEN(STR$(NUM))-1)
910 IF LEN(DA$) < 235 THEN DA$ = DA$+", "
   : GOTO 930
920 LNUM = LNUM+10
   : PRINT#2, DA$
   : DA$ = RIGHT$(STR$(LNUM), LEN(STR$(LNUM))-1)
   + " DATA "
930 WEND
940 PRINT#2, DA$
950 PRINT#2, STR$(LNUM+1) + " DATA 99999"
960 CLOSE
   : GOTO 70
970 PRINT@(17,3), "<N>ext Pattern, <C>hange this
   pattern, <R>epeat value, <S>top this line,
   <Q>uit";
   : RETURN
980 PRINT@(8,12), USING "Line: ### Dot Position:
   ### Max Dots in Mode: ###"; LIN, DOT, DDEN;
   : RETURN

```

# Color Change Routine for Model 2000 COBOL

Dennis Weiss

This article will show you how to implement color options in COBOL on a Model 2000 with color graphics. With this routine and its associated COBOL patch in place, you will be able to use the ACCEPT and DISPLAY statements of COBOL with your choice of foreground and background color. A CALL to "COLORCNG.COM" can change the colors of your next display as many times as you wish. The colors remain in effect until the next CALL.

To implement Color Change, you will need some knowledge of assembler language, experience with the DEBUG utility and the MS™-ASSEMBLER (26-5252). Below are the steps required. The COLORCNG.COM routine, the patch to RUNCOBOL and the sample COBOL program follow.

1. Key-in the color change sub-routine. It is possible to combine this with other assembler sub-routines you are using. Refer to pages 40-41 of the Model 2000 COBOL User's Guide. The set of all routines is known as a "procedure table". If you are not going to use the PUTCHAR and GETCHAR routines included on the COBOL disk, you will need to rename the PROCTBL.COM that exists on the disk to another name. Assemble the sub-routine(s) using your assembler package. If the assembler is the Microsoft package (26-5252), the source file must have an extension of .ASM. Let the output file of this be named "PROCTBL.OBJ".
2. Link the sub-routines to RUNCOBOL. Follow the procedure in the User's Guide. Upon LINKING there will be a warning of "NO STACK SEGMENT". This warning and the resultant error count can be ignored. You may use either method shown to strip the header from the link file. Both will work. In either case, the output should be named "PROCTBL.NEW".
3. Apply the patch to RUNCOBOL using DEBUG. If you will be using a color monitor all the time, enter 01 in address 02B4. If not, you will have to modify this location every time you switch monitors. Write out the patched RUNCOBOL.COM. It is now ready to run the test program.
4. Key-in, compile and run the sample COBOL program to test the color change CALLs. If there is a problem, go back over the preceding steps. Look carefully at what you keyed in. Correct if necessary and rerun the COBOL program.

Once you have verified that you can change the colors of your COBOL displays, you will have a permanent color changing ability for your COBOL programs.

## Color Change Assembly Routine

```

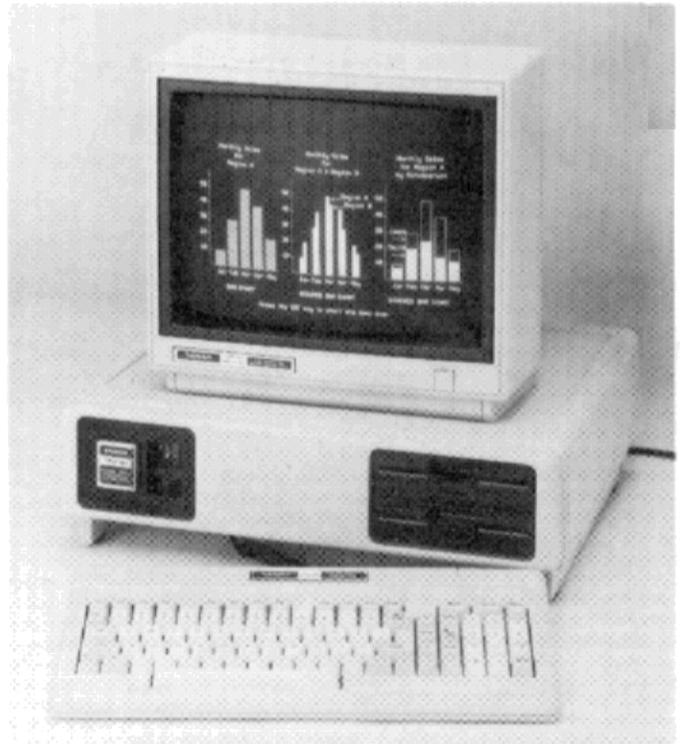
;
;   COLOR CHANGE ASSEMBLY SUBROUTINE
;
;   THIS ROUTINE IS CALLED FROM RM/COBOL
;   WITH THE FOLLOWING SYNTAX
;
;       CALL "COLORCNG.COM" USING fg
;       color, bg color
;

```

```

;   NOTE THAT THE ADDRESS OF THE HOLD-BYTES
;   FOR THE FOREGROUND-BACKGROUND COLOR
;   COMBINATION AND REVERSE COMBINATION IS
;   KNOWN TO THIS ROUTINE.
;
CODE SEGMENT
    DW  MODULE_ENDS
    DB  12
    DB  'COLORCNG.COM'
    DW  COLORCNG_ENTRY
    DB  0
COLORCNG_ENTRY PROC FAR
C_COLOR_BYTE_ADDR EQU 0104H
    MOV  DI,ES:2[BX]    ;FOREGROUND COLOR
    MOV  DL,ES:[DI]
    MOV  DI,ES:4[BX]    ;BACKGROUND COLOR
    MOV  DH,ES:[DI]
    SUB  DL,'0'         ;ADJUST FOR ASCII
    SUB  DH,'0'
    MOV  BX,DX          ;SAVE ORIGINAL VALUES
;
    MOV  CL,4
    SHL  DH,CL         ;BACKGROUND
    OR   DL,DH         ;DL HAS BG-FG COMBINATION
    SHL  BL,CL         ;MAKE REVERSE VIDEO
    OR   BH,BL
    MOV  DH,BH         ;DH HAS REVERSED COLORS
    MOV  BX,C_COLOR_BYTE_ADDR
    MOV  ES:[BX],DX
    XOR  AX,AX
    RET
COLORCNG_ENTRY ENDP
MODULE_ENDS EQU $
CODE ENDS
END

```



## Patch to RUNCOBOL.COM

Address	Old	New
1EA2	00 (14)	2E F6 06 B4 02 FF 74 05 B8 03 00 CD 10 C3
02B4	00	01 = color 00 = mono

4563	00 (27)	53 E8 6B 0D 5B 4F 8B 0E 04 01 A9 00 02 74 02 86 CD E3 05 80 25 88 08 0D E9 34 0D	4599	88 3E 05 0E E8 3F 01 C3	C6 06 05 0E 00 E9 3E 01
457E	xx	00	51AB	EB 00	B6 F3
4588	88 0E 03 0E	E8 17 D9 90	5AE6	xx	8B
			6903	00	4F

**Sample COBOL Program for Testing Color Changes**

Column -----  
1.....7 8.....72

```

IDENTIFICATION DIVISION.
PROGRAM-ID.                COLORTST.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.          MODEL-2000.
OBJECT-COMPUTER.         MODEL-2000.

DATA DIVISION.
WORKING-STORAGE SECTION.

01  COLORS.
    03  BLUE                PIC 9           VALUE 1.
    03  GREEN              PIC 9           VALUE 2.
    03  RED                 PIC 9           VALUE 4.
    03  WHITE              PIC 9           VALUE 7.
    03  DEFAULT            PIC 9           VALUE 0.

01  DISPLAY-CONSTANTS.
    03  FIELD-1            PIC X(7)        VALUE "COLOR-1".
    03  FIELD-2            PIC X(7)        VALUE "COLOR-2".
    03  FIELD-3            PIC X(7)        VALUE "COLOR-3".
    03  FIELD-4            PIC X(7)        VALUE "COLOR-4".
    03  FIELD-5            PIC X(7)        VALUE "COLOR-D".

PROCEDURE DIVISION.
BEGIN.
    DISPLAY "BEGIN", LINE 1, ERASE.

*           SET FOREGROUND, BACKGROUND TO BLUE, RED
    CALL "COLORCNG.COM" USING BLUE RED.
    DISPLAY FIELD-1, LINE 3.
    DISPLAY FIELD-4, LINE 4.

*           SET FOREGROUND, BACKGROUND TO GREEN, WHITE
    CALL "COLORCNG.COM" USING GREEN WHITE.
    DISPLAY FIELD-2, LINE 6.
    DISPLAY FIELD-4, LINE 7.

*           SET BACK TO DEFAULT
    CALL "COLORCNG.COM" USING DEFAULT DEFAULT.
    DISPLAY FIELD-5, LINE 10.
    STOP RUN.

END PROGRAM.

```

# Tandy Model 2000 Character Editor

Paul Opitz  
6816 Meadow Creek  
North Richland Hills, TX 76180

## INTRODUCTION

In reading the literature about the Tandy Model 2000, quite frequently one will come across a reference to the ease with which the character set can be modified. This feature of the Model 2000 can be of great commercial value. With virtually all presently available microcomputers, it is necessary to replace the character generator ROM for alternate character sets to be accessed. With the Tandy 2000, one need only POKE the correct value into the correct area of memory and presto, you have a new value and the correct memory location. Wouldn't it be so much easier if there was a computer program to do this for you . . . ?

## THE PROGRAM

Throughout this program, you will be dealing with one of two different character sets. For ease of discussion the character sets are defined as follows:

- Working Set: The set that is displayed on the screen while in the Selection Mode.
- Active Set: The set that is currently used to display characters.

For example, if you were to create a Japanese character set, while you are working on the characters, the set is the working set. If the characters that the program displayed as it's prompts were in Japanese, then this would also be your active character set. If, on the other hand, the characters were displayed in the standard English character set, the active character set would be English.

CHAREDIT allows the user to easily design, modify, save, and recall complete character sets. Operation is centered around two modes. First is the Selection Mode. In this mode, the user can select a character to be modified, save a modified character set onto a file on disk, or load a character set into memory. Next is the Edit Mode. In this mode, a character has been selected in the Selection Mode, and is displayed in an 8 x 16 grid. The user is allowed to turn each individual pixel either on or off to build the character. This greatly reduces the effort required in changing a character set, since the user need have no technical knowledge.

The next section will walk you through the steps required to change one character in the standard character set, save it to disk, replace the set in memory, and finally save the new active set on disk.

## AN EXAMPLE

As an easy example, we are going to modify the letter "Z" to have a horizontal line through the center of the character. This is useful for some persons in distinguishing this letter from the numeral "2".

To start, put in the program disk and type:  
BASIC CHAREDIT

This will load and run the program. You are prompted to enter the CHARACTER SET TO MODIFY. At this prompt, type:  
ENGLISH

This will load the default character set into the working area. As you can see, the entire character set is displayed at the top of the screen. Using the arrow keys, place the underline below the letter "Z" and press the **ENTER** key. You could simply press the **Z** key. This causes the program to load the single letter "Z" into the Editor Mode. At the Edit mode, you will see the letter "Z" displayed in a large outlined area on the screen. At this point, pressing any of the arrow keys will cause the flashing cursor to move around within the outlined area. Move the cursor to a point approximately two locations to the left of the center of the "/" on the "Z" and press the **W** key, then the **RIGHT ARROW**. The place the cursor was on when you pressed the **W** key is now white. If you need to turn a location black, you would press the **B** key. Continue to press **W** then **RIGHT ARROW** until you have completed the character. At this point press **ENTER**. Your first screen should reappear with the "Z" now modified.

Now that we have a new character set, we need to save it to disk for future editing, and for transfer to the active set. To do this, simply press the **F1** key. You will be prompted to SAVE WORKING CHARACTER SET (ABOVE) OR ACTIVE CHARACTER SET (IN MEMORY) (W/A). At this time, we want to save the working character set, so press the **W** key. Now you will be prompted for NAME TO SAVE AS. Enter a name that is both descriptive and legal under MS-DOS. For our example, let's call the file "ZED". The working character set is now stored on disk. To later call this file for editing, you would simply answer "ZED" to the initial prompt for CHARACTER SET TO MODIFY.

It does not do much good to have a character set in working memory if we want that character set to print from our own programs. For this, we need to transfer the working set to the active set. This is the function of the **F2** key or REPLACE SET function. To transfer your working set into the active set, simply press the **F2** key. This will do the transfer.



At this point, if you were to exit the program and press the letter "Z" the letter would have the horizontal line through it.

Once a new character set has been loaded into active memory it is possible to save the active set for quicker retrieval later. Press the (F1) key again. This time at the prompt press the (A) key for ACTIVE. Enter "ZEDACT" as the file to save as. Now, any time you wish to use this alternate character set from your BASIC program, simply include the statement:

```
BLOAD "ZEDACT"
```

## INITIALIZATION

Before running the main program for the first time, it is necessary to run an initialization program. This program will create ENGLISH as the default working character set, and ENGLISHA as the default active character set. The set used will be the standard character set as provided with the system. Simply type in the following program as shown and RUN it.

```
10 CLS
20 XLOC = 1
   YLOC = 1
30 FOR OFFSET = &HF800 TO &HF9FF STEP 2
40 FOR BYTE = 0 TO 31 STEP 2
50 DEG SEG = OFFSET
60 A = PEEK(BYTE)
70 DEG SEG = &HE000
80 POBYTE = (BYTE*40+(XLOC-1)*2+(YLOC-1)*1600)
90 POKE POBYTE,A
100 NEXT BYTE
110 IF XLOC = 40 THEN YLOC = YLOC+1
   : XLOC = 1 ELSE XLOC = XLOC+1
120 NEXT OFFSET
130 DEG SEG = &HE000
140 BSAVE"ENGLISH",0,10912
150 DEG SEG = &HF800
160 BSAVE"ENGLISHA",0,8176
170 LOCATE 20,1
   : END
```

If you wish to "jazz up" the opening screen, create a boldface font on the disk, and call it BOLD (working) and BOLDDACT (active), and an italic font, calling it ITALIC (working) and ITALACT (active). If you have a color monitor or a monochrome monitor which has the BW Graphics Kit (26-5140) installed, all three fonts (English, Bold, and Italic) will be used in the opening screen. If you are using a monochrome monitor without the BW Graphics Kit modification, only one character set can be displayed at any one time. Therefore, the opening screen section of the program will have to be modified.

The program as shown will work on the monochrome monitor only. For the color monitor delete line 5 and change the REM statements in lines 95, 135, 165, 175, and 230 to COLOR statements.

```
5 COLOR 0,7
   : SCREEN 3,0
10 '*****
20 '*** CHARACTER SET EDITOR ***
30 '*** BY - PAUL OPITZ, COMPUTER CONSULTANT***
40 '*** RADIO SHACK FRANCHISE INTERNATIONAL ***
50 '*** INTERNATIONAL DEALER DIVISION ***
60 '*****
70 ON ERROR GOTO 190
```

```
80 BLOAD"BOLDDACT"
90 CLS
95 REM COLOR 12,8
100 LOCATE 2,30
   : PRINT"CHAR"
   : LOCATE 2,44
   : PRINT"EDIT"
110 LOCATE 4,35
   : PRINT"PAUL OPITZ"
120 LOCATE 6,23
   : PRINT"RADIO SHACK FRANCHISE INTERNATIONAL"
130 BLOAD"ITALACT"
135 REM COLOR 13,8
140 LOCATE 3,39
   : PRINT"BY"
   : LOCATE 5,31
   : PRINT"COMPUTER CONSULTANT"
150 LOCATE 7,25
   : PRINT"INTERNATIONAL DEALER DIVISION"
160 BLOAD"ENGLISHA"
165 REM COLOR 14,8
170 LOCATE 2,34
   : PRINT"ACTER SET"
   : LOCATE 2,48
   : PRINT"OR"
175 REM COLOR 15,8
180 FOR X = 1 TO 25000!
   : NEXT
   : GOTO 200
190 IF ERL = 80 THEN RESUME 90 ELSE IF ERL = 130
   THEN RESUME 140 ELSE IF ERL = 160 THEN
   RESUME 170
200 DIM A$(16)
210 DEF SEG = &HE000
220 KEY OFF
230 'COLOR 0,7
   : SCREEN 3,0
240 KEY(14) ON
   : KEY(15) ON
   : KEY(13) ON
   : KEY(16) ON
   : KEY(1) ON
   : KEY(2) ON
   : KEY(3) ON
   : KEY(4) ON
250 'PALETTE
260 CLS
270 ON ERROR GOTO 1820
280 '*****
290 '*** GET A "CHARACTER" FILE TO EDIT ***
300 '*****
310 INPUT"CHARACTER SET TO MODIFY";LANG$
320 CLS
330 BLOAD LANG$
340 '*****
350 '*** ROUTINE TO SELECT A CHARACTER ***
360 '*** FROM THE CURRENT WORKING SET ***
370 '*** TO EDIT ***
380 '*****
390 LOCATE 22,1
   : PRINT"PLACE THE CURSOR ON THE CHARACTER TO
   MODIFY AND PRESS <ENTER>"
400 LOCATE 23,1
   : PRINT "<F1> TO SAVE : <F2> TO REPLACE SET
   : <F3> TO QUIT : <F4> TO GET NEW SET"
410 POKE 1520,255
420 XLOC = 1
   : YLOC = 1
430 ON KEY(14) GOSUB 550
   : ON KEY(15) GOSUB 600
   : ON KEY(13) GOSUB 650
   : ON KEY(16) GOSUB 700
440 ON KEY(1) GOSUB 1450
   : ON KEY(2) GOSUB 1580
   : ON KEY(3) GOSUB 1790
   : ON KEY(4) GOSUB 260
```

```

450 A$ = INKEY$
   : IF A$ = "" THEN 450
460 IF A$ = CHR$(13) THEN 800
470 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
480 CHLOC = ASC(A$)
   : YLOC = INT(CHLOC/40)+1
   : XLOC = CHLOC-(YLOC-1)*40+1
490 GOTO 800
500 '*****
510 '*** THE FOLLOWING 4 ROUTINES SERVE ***
520 '*** TO MOVE THE UNDERLINE UNDER THE ***
530 '*** WORKING CHARACTER SET ***
540 '*****
550 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
560 IF XLOC = 1 AND YLOC = 1 THEN XLOC = 15
   : YLOC = 7
   : GOTO 580
570 IF XLOC = 1 THEN XLOC = 40
   : YLOC = YLOC-1 ELSE XLOC = XLOC - 1
580 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,255
590 RETURN
600 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
610 IF XLOC = 15 AND YLOC = 7 THEN XLOC = 1
   : YLOC = 1
   : GOTO 630
620 IF XLOC = 40 THEN XLOC = 1
   : YLOC = YLOC + 1 ELSE XLOC = XLOC + 1
630 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,255
640 RETURN
650 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
660 IF YLOC = 1 THEN YLOC = 7 ELSE YLOC = YLOC-1
670 IF YLOC = 7 THEN IF XLOC>15 THEN XLOC = 15
680 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,255
690 RETURN
700 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
710 IF YLOC = 7 THEN YLOC = 1 ELSE YLOC = YLOC+1
720 IF YLOC = 7 THEN IF XLOC>15 THEN XLOC = 15
730 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,255
740 RETURN
750 '*****
760 '*** THIS ROUTINE DISPLAYS THE SELEC-***
770 '*** TED CHARACTER IN A LARGE FORMAT ***
780 '*** FOR EDITING PURPOSES ***
790 '*****
800 CAR = XLOC+(YLOC-1)*40-1
810 ON KEY(13) GOSUB 1190
   : ON KEY(14) GOSUB 1210
   : ON KEY(15) GOSUB 1230
   : ON KEY(16) GOSUB 1250
820 OFFSET = &HE000
830 FOR X = 1 TO 16
840 A$(X) = " "
850 NEXT X
860 DEF SEG = OFFSET
870 FOR B = 0 TO 15
880 A = PEEK(B*80+(XLOC-1)*2+(YLOC-1)*1600)
890 FOR X = 8 TO 1 STEP -1
900 IF INT(A) >= INT(2^(X-1)) THEN MID$(A$(B+1),17-X*2,2) = CHR$(219)+CHR$(219)
   : A = A-(2^(X-1)) ELSE
   : MID$(A$(B+1),17-X*2,2) = " "
910 NEXT X
920 NEXT B
930 POKE (XLOC-1)*2+(YLOC*1520)+(YLOC-1)*80,0
   : BSAVE"TEMP",0,10912
940 CLS
950 FOR X = 1 TO 16
   : LOCATE 6,19+X
   : PRINT CHR$(205)
   : LOCATE 23,19+X
   : PRINT CHR$(205)
   : NEXT X
960 FOR X = 1 TO 16
   : LOCATE 6+X,19
   : PRINT CHR$(186)
   : LOCATE 6+X,36
   : PRINT CHR$(186)
   : NEXT X
970 LOCATE 6,19
   : PRINT CHR$(201)
   : LOCATE 6,36
   : PRINT CHR$(187)
   : LOCATE 23,19
   : PRINT CHR$(200)
   : LOCATE 23,36
   : PRINT CHR$(188)
980 FOR X = 1 TO 16
   : LOCATE X+6,20
   : PRINT A$(X)
   : NEXT X
990 LOCATE 1,1
   : PRINT"PRESS <B> FOR BLACK, <W> FOR WHITE,
   : <ENTER> TO SAVE"
1000 XPOS = 20
   : YPOS = 7
1010 LOCATE YPOS,XPOS+FLIP
   : IN$ = INKEY$
1020 FLIP = ABS(FLIP=0)
1030 '*****
1040 '*** WAIT FOR EITHER A "B", "W", OR ***
1050 '*** <ENTER>. "B" WILL BLANK THE ***
1060 '*** CURRENT CELL, "W" WILL WHITE ***
1070 '*** THE CURRENT CELL, AND <ENTER> ***
1080 '*** WILL CAUSE THE WORKING SET TO ***
1090 '*** UPDATED W/THIS CHARACTER ***
1100 '*****
1110 IF IN$<>CHR$(13) AND IN$<>"B" AND IN$<>"W"
   AND IN$ <> "b" AND IN$ <> "w" THEN 1010
1120 IF IN$ = "B" OR IN$ = "b" THEN
   MID$(A$(YPOS-6),XPOS-19,2) = " "
   : LOCATE YPOS,20
   : PRINT A$(YPOS-6)
   : GOTO 1010
1130 IF IN$ = "W" OR IN$ = "w" THEN
   MID$(A$(YPOS-6),XPOS-19,2) =
   CHR$(219)+CHR$(219)
   : LOCATE YPOS,20,1
   : PRINT A$(YPOS-6)
   : GOTO 1010
1140 GOTO 1270
1150 '*****
1160 '*** ROUTINE TO MOVE THE CURSOR IN ***
1170 '*** THE CHARACTER EDIT MODE ***
1180 '*****
1190 IF YPOS = 7 THEN YPOS = 22 ELSE YPOS =
   YPOS-1
1200 RETURN
1210 IF XPOS = 20 THEN XPOS = 34
   : GOTO 1190 ELSE XPOS = XPOS-2
1220 RETURN
1230 IF XPOS = 34 THEN XPOS = 20
   : GOTO 1250 ELSE XPOS = XPOS+2
1240 RETURN
1250 IF YPOS = 22 THEN YPOS = 7 ELSE YPOS =
   YPOS+1
1260 RETURN
1270 CLS
1280 '*****
1290 '*** <ENTER> ROUTINE TO EXIT EDIT ***
1300 '*** MODE. ***
1310 '*****
1320 BLOAD"TEMP"
1330 FOR B = 0 TO 15
1340 A = 0
1350 FOR X = 1 TO 8
1360 IF MID$(A$(B+1),17-X*2,2) =
   CHR$(219)+CHR$(219) THEN A = A+2^(X-1)
1370 NEXT X
1380 POKE (B*80+(XLOC-1)*2+(YLOC-1)*1600),A
1390 NEXT B
1400 GOTO 390

```

```

1410 *****
1420 *** ROUTINE TO SAVE THE WORKING OR ***
1430 *** ACTIVE CHARACTER SET: ***
1440 *****
1450 LOCATE 18,1
      : INPUT"SAVE WORKING CHARACTER SET (ABOVE)
      OR ACTIVE CHARACTER SET (IN MEMORY)
      (W/A)";WH$
1460 IF WH$ = "W" THEN OFFSET = &HE000
      : QUAN = 10912 ELSE IF WH$ = "A" THEN OFFSET
      = &HF800
      : QUAN = 8176 ELSE GOTO 1450
1470 LOCATE 20,1
      : INPUT"NAME TO SAVE AS";NME$
      : DEF SEG = OFFSET
      : BSAVE NME$,0,QUAN
1480 LOCATE 21,1
      : PRINT"<DONE>"
      : FOR TIMER = 1 TO 500
      : NEXT
1490 LOCATE 20,1
      : PRINT"
1500 LOCATE 21,1
      : PRINT"
1510 LOCATE 18,1
      : PRINT STRING$(79," ")
1520 OFFSET = &HE000
1530 RETURN
1540 *****
1550 *** ROUTINE TO TRANSFER THE WORKING ***
1560 *** SET TO THE ACTIVE SET ***
1570 *****
1580 X = XLOC
      : Y = YLOC
      : XLOC = 1
      : YLOC = 1
1590 FOR M = 13 TO 16
      : KEY(M) OFF
      : NEXT M
      : KEY(1) OFF
      : KEY(2) OFF
      : KEY(3) OFF
1600 FOR OFFSET = &HF800 TO &HF9FF STEP 2
1610 LOCATE 20,1,0
      : PRINT XLOC-1+(YLOC-1)*40
1620 FOR BYTE = 0 TO 31 STEP 2
1630 DEF SEG = &HE000
1640 A = PEEK(BYTE*40+(XLOC-1)*2+(YLOC-1)*1600)
1650 DEF SEG = OFFSET
1660 POKE BYTE,A
1670 NEXT BYTE
1680 IF XLOC = 15 AND YLOC = 7 THEN XLOC = 1
      : YLOC = 1
      : GOTO 1700
1690 IF XLOC = 40 THEN YLOC = YLOC+1
      : XLOC = 1 ELSE XLOC = XLOC+1
1700 NEXT OFFSET
1710 FOR M = 13 TO 16
      : KEY(M) ON
      : NEXT M
      : KEY(1) ON
      : KEY(2) ON
      : KEY(3) ON
1720 XLOC = X
      : YLOC = Y
1730 DEF SEG = &HE000
1740 LOCATE 20,1
      : PRINT "
1750 RETURN
1760 *****
1770 *** END PROGRAM ROUTINE ***
1780 *****
1790 ON ERROR GOTO 1810

```

```

1800 KILL"temp.bas"
1810 LOCATE ,,1
      : END
1820 IF ERR = 53 OR ERR = 54 THEN PRINT "NOT
      AVAILABLE"
      : FOR TIMER = 1 TO 1000
      : NEXT
      : RESUME 310
1830 PRINT "ERROR ";ERR;" IN LINE ";ERL
16514 ,8

```

## MAGAZINES

Below are nine magazines of special interest to TRS-80 owners that we believe have editorial content of high quality and will be of use to our customers.

### *Advanced Computing Magazine\**

131 East Orange Street  
Lancaster, PA 17602  
(717)397-3364 (Corrected from May)

### *Color Computer Magazine*

Highland Hill  
Camden, ME 04843  
(207)236-9621

### *Color Micro Journal*

5900 Cassandra Smith Rd.  
Hixson, TN 37343

### *Computer User*

16704 Marquardt Ave.  
Cerritos, CA 90701

### *80 Micro*

P.O. Box 981  
Farmingdale, NY 11737

### *Hot CoCo*

P.O. Box 975  
Farmingdale, NY 11737

### *Portable 100—The Magazine for Model 100 Users*

67 Elm Street  
P.O. Box 250  
Camden, ME 04843  
(207)236-4365

### *PCM—The Portable Computing Magazine*

9529 U.S. Highway 42  
P.O. Box 209  
Prospect, KY 40059

### *Rainbow (Covers the TRS-80 Color Computer)*

P.O. Box 209  
Prospect KY 40059  
(502)228-4492

\*Formerly two/sixteen magazine

# Business Reports That "Show" and Tell With Business Graphics Analysis Pak for Models II/12

by Carol Morton

When you're talking statistics, to paraphrase Eliza Dolittle, don't just tell 'em, show 'em. It's great to be able to say productivity is rising while production costs are easing, but it is more effective to graphically show the productivity line steadily climbing and the cost factor line rolling down hill.

Statistics are more impressive, emphatic, dramatic, and, in some cases, more meaningful when presented in concrete form, such as graphs. Whatever your statistical data, it has more impact when accompanied by charts.

If you deal with statistical data and/or are responsible for the reporting of such data you are probably aware of how much charts can add to your report. However, you may have difficulty finding, or justifying, the time needed to translate your data into charts. Or perhaps your artistic skills just don't measure up to your own expectations for such material. If you are doing without charts or if you are still producing your graphs with your own busy hands, we have a package to lighten your load and brighten your reports.

With Business Graphics Analysis Pak for Models III/4 and II/12 you can create bar, line, and scatter graphs as well as pie charts at your computer keyboard. You can enter and/or generate data at the time you create the graph or you can use data extracted from VisiCalc DIF, SCRIPSIT, BASIC, or FORTRAN files. The program is entirely menu driven and when you make a mistake, it not only tells you where the problem lies but what steps you can take to correct it.

And what kind of printout do you get? Well, what kind of printer do you have? The hardcopy can be produced on nearly any Radio Shack printer from the Line Printer V through the LPVIII, including the Daisy Wheel II, 210 and 410 printers, the Multi-Pen Plotter, both the Color Graphics Printers 115 and 220, the Single-Pen Flatbed Color Plotter 215, and the DMPs 100 through 2100.

Note to Model III/4 Users: Business Graphics Analysis Pak for the Models III/4 has been upgraded to support additional printers, plotters, and the high-resolution graphics board. The upgrade supports the following printers: DMP-100, DMP-120, DMP-200, DMP-400, DMP-420, DMP-500, DMP-2100, Line Printer V, Line Printer VI, Line Printer VII, Line Printer VIII, TRS-80 Multi-Pen Plotter, FP-215 Flatbed Plotter/Printer, CGP-115 Color Graphics Printer, CGP-220 Color Ink-Jet Printer, Daisy Wheel II, DWP-210, and DWP-410. To obtain this upgrade order catalog number 700-6205 from your local Radio Shack store or dealer. Suggested retail price of the upgrade is \$50.00.

What we're saying here is that, bottom line, this package not only has a lot to offer, it is EASY to use. To give you an idea of the procedure, we'll start with some data and create a chart.

## DATA

To make any of the charts you need data points. This can be done as a part of the chart creation process, entering the data at the keyboard, or you can actually use data extracted from VisiCalc DIF, Scripsit, BASIC or FORTRAN files. To access data from Scripsit, BASIC and FORTRAN files, you first convert to ASCII files.

The following procedures were used to access some data from a Scripsit file. If the procedure seems a bit lengthy remember this example contains a very short list. The value of the conversion process increases significantly when you are faced with files containing long strings of data.

The data must be formatted so that it is separated by commas, blanks, or carriage returns. We created a new Scripsit file, duplicated the page containing the data, deleted all extraneous material and then formatted the data into columns. (See figure 1.)

17.20	28.90	168.00
22.00	34.20	205.00
26.80	41.50	249.00
34.00	52.00	312.00
46.00	73.75	442.00
58.00	82.00	492.00

Figure 1

Next, we converted to an ASCII file which we called BUSDAT/ASC. If you have a multiple disk drive system, place the Business Graphics Setup Disk in Drive 1 and save your ASCII file to that drive (BUSDAT/ASC:1). Otherwise, you will need to copy the ASCII file to the Business Graphics Setup Disk you will use to create your chart.

From the list above we created three separate data files to be used in our chart. With the Business Graphics Setup Disk in Drive 0 (or Drive 1 in a multiple floppy drive system), at TRSDOS READY type:

```
BASIC READATA1/BAS-F:3
```

READATA1/BAS is the conversion utility for floppy disk systems and -F:3 indicates we will convert three files. (The manual cautions you to always convert at least two files.)

Now it is a simple matter of answering the prompts.

NAME OF FILE TO BE READ? BUSDAT/ASC

NAME OF FILE TO BE CREATED? BDAT1/DAT

We used a number in the name of the file to be created for convenience; rather than remembering three different names we need only remember BDAT and the number of BDAT files created.

READ ALL VALUES OR READ SELECTED VALUES (ALL/SELECT) S

We responded with "S" so that we could put the data from one column only into the first file.

START WITH WHICH VALUE (ENTER SEQUENCE #)? 1

SELECT EVERY 'NTH' VALUE (E.G., ENTER 5 FOR EVERY 5TH)? 3

NUMBER OF VALUES TO BE SELECTED (OR ALL)? 4

With these three responses we placed the first four items from column one above into our first file.

DO YOU WANT TO SKIP NON-NUMERIC DATA OR CHANGE THEM? (SKIP/CHANGE)? <u>S</u>			
INPUT SEQUENCE#	INPUT VALUE	OUTPUT SEQUENCE#	OUTPUT VALUE
1	17.20	1	17.2
4	22.00	2	22
7	26.80	3	26.8
10	34.00	4	34

OUTPUT FILE BDAT1/DAT CONTAINS 4 VALUES  
DO YOU WANT TO CREATE ANOTHER FILE (YES/NO)? Y

At this point the process begins again with the "FILE TO BE READ" prompt. We named the second file BDAT2/DAT and the third BDAT3/DAT. For the second file the response to the "START WITH WHICH VALUE . . ." prompt was 2 and for the third it was 3. All the other responses remain the same until you are ready to respond "N" to the ". . . CREATE ANOTHER FILE . . ." query.

## MENU

The menu all but takes you by the hand and leads you step by step through the creation of the chart. (We underlined the responses we input. Responses not underlined are default responses.)

Place the Business Graphics Analysis Pak Processing Disk in Drive 0 and at TRSDOS READY type:  
TRSCHART ENTER.

The Main Menu will appear on the screen.

MAIN MENU	
1	DATA HANDLING MENU
2	LINE CHART MENU
3	BAR CHART MENU
4	PIE CHART MENU
5	SCATTER CHART MENU
6	CHART TEXT EDITOR MENU
7	STOP

We'll use our BDAT files to create a bar chart. Select option 3 "Bar Chart Menu."

## BAR CHART MENU

- 1 DATA & FORMAT SETTINGS
- 2 TEXT SETTINGS
- 3 PRINTER/PLOTTER SETTINGS
- 4 LOAD SETTINGS:
- 5 PRINT SETTINGS
- 6 SAVE SETTINGS:
- 7 DISPLAY CHART
- 8 PRINT CHART
- 9 SAVE CHART:
- 0 RETURN TO MAIN MENU

To load the data files, choose the format of the bars and enter the scale and dimensions of the chart, we select Bar Chart Menu option 1, "Data & Format Settings" and make the following entries.

DATA & FORMAT SETTINGS	
1 DATA FILE FOR BAR SET 1:	<u>BDAT1/DAT</u>
2 DATA FILE FOR BAR SET 2:	<u>BDAT2/DAT</u>
3 DATA FILE FOR BAR SET 3:	<u>BDAT3/DAT</u>
4 BAR SET 1 FORMAT:	DARK BLACK
5 BAR SET 2 FORMAT:	LIGHT BLACK
6 BAR SET 3 FORMAT:	NOFILL BLACK
7 LAYOUT FOR MULTIPLE BAR SETS:	<u>GROUPED</u>
8 HORZ AXIS LENGTH (CHARS):	50
9 VERT AXIS LENGTH (LINES):	36
0 VERTICAL SCALE RANGE:	<u>0,350</u>

(PRESS "ESC" KEY TO RETURN TO BAR CHART MENU)

Next we return to the Bar Chart Menu, select option 2, "Text Settings," and make the following entries.

TEXT SETTINGS	
1 TOP TITLE:	<u>SAMPLE BAR CHART</u>
2 LEFT TITLE:	NONE
3 BOTTOM TITLE:	NONE
4 VERTICAL-AXIS LABELS:	<u>NOCOMMAS DEC = 1 \$</u>
5 HORIZONTAL-AXIS LABELS:	<u>1Q83</u>
6 CHART FRAME:	<u>YES</u>

(PRESS "ESC" KEY TO RETURN TO BAR CHART MENU)

The entries by option 4 indicate that we will have no commas, one decimal place and a dollar sign as the leading character on the vertical-axis labels.

The entry for the horizontal-axis label should be read "First Quarter, 1983."

We skipped option 3, "PRINTER/PLOTTER SETTINGS," on the Bar Chart Menu since we kept the automatic settings for this chart. Option 7, "DISPLAY CHART," is a good choice at this point. Once satisfied that the chart appears to be correct, and providing no error messages are received, we can either print the chart or save it. Because we want to add a descriptive line which will identify the bars, we saved the chart (option 9) under the title SAMPLE1.

Now we return to the Main Menu. To complete the last step, adding the identifying label, select option 6, "CHART TEXT EDITOR MENU," and load the chart.

### CHART TEXT EDITOR MENU

- 1 LOAD CHART: SAMPLE1
- 2 DISPLAY CHART
- 3 CREATE HORIZONTAL LABEL
- 4 CREATE VERTICAL LABEL
- 5 TYPE OVER EXISTING CHARACTERS
- 6 INSERT OR DELETE CHARACTERS
- 7 MOVE LABEL
- 8 PRINT CHART
- 9 SAVE CHART:
- 0 RETURN TO THE MAIN MENU

In order to create the label, first go to option 2, "DISPLAY CHART." When the chart appears on the screen, move the cursor, using the arrow keys, to the place on the chart where you want the label to begin. Return to the Chart Text Editor Menu and select option 3, "CREATE HORIZONTAL LABEL." When the chart reappears, begin typing the label. Once you are satisfied with the appearance of the label, return to the Chart Text Editor Menu. Options 5 through 7 provide a means of changing the label once you have returned to the Chart Text Editor Menu. Figure 2 is our version of the chart produced on a DMP-2100.

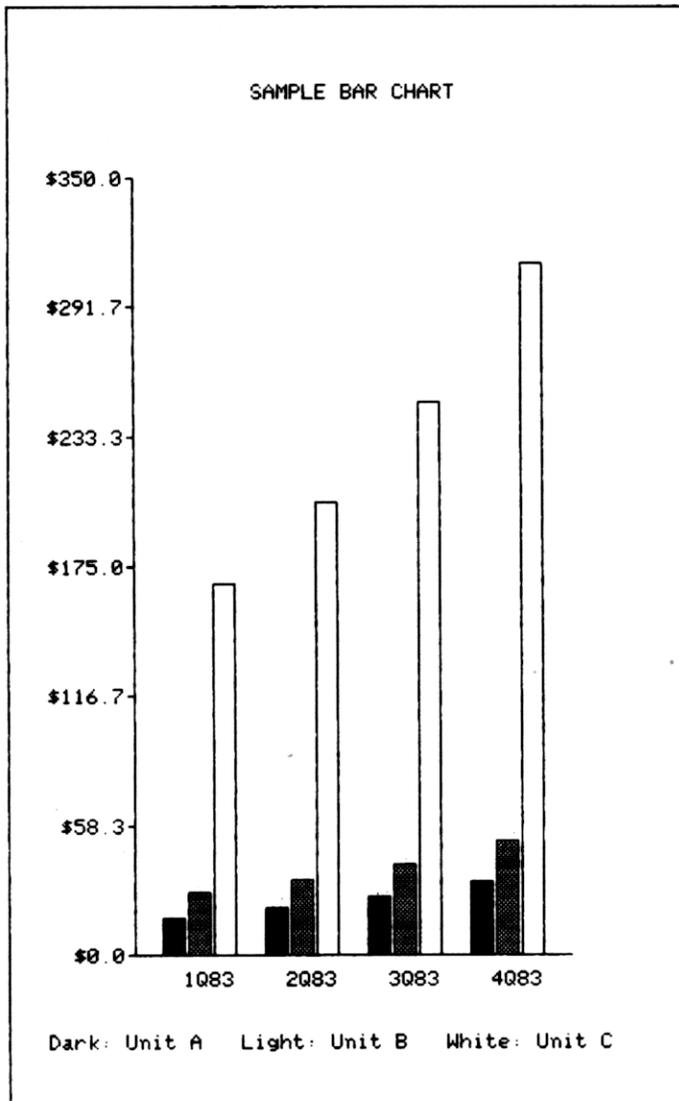


Figure 2

## PRINTER

The choice of printer is as broad as the Radio Shack printer line. Business Graphics Analysis Pak comes with three disks, the Business Graphics Analysis Processing Disk and two Setup Disks. One Setup Disk is primarily for use with most of the black and white only printers. Setup Disk 2 contains the configuration programs for all color devices (the Multi-Pen Plotter, CGP-220, CGP-115, and the Single-Pen Flatbed Color Plotter FP-215).

With this broad range of printer options you can see that you don't have to have a dot-matrix printer or a color printer to create your graphs and charts. However, let me forewarn you. Once you see what a CGP-220 and Business Graphics Analysis Pak can do with graphs, you're going to want one.

There are two potential problems emanating from the way the printer option programs are provided on the different disks. The first potential problem will not occur unless you have more than one type of printer at your disposal. If you have the option of using more than one printer, make sure that you have configured your disk for the printer you will use BEFORE you create your chart. Once the chart is created and saved, the printer settings for the chart (or graph) will be formatted for the printer for which it was configured. In other words, if you want to print the same chart on another printer, you will have to recreate the chart after reconfiguring your disk for the new printer. (Printer disk configuration is a simple matter of typing in the appropriate printer configuration program code, such as DMP2100, at TRSDOS READY. The configuration program will be initiated and will set up the printer defaults on the disk for use with that printer. The configuration does not affect data files so you can still use your data files to create more charts.)

The second potential problem effects only those who might want to use Setup Disk 2 to convert SCRIPSIT, BASIC and FORTRAN files. In order to use Setup Disk 2 for this procedure, you will need to copy READATA1/BAS and RD-SUBR20 from Setup Disk 1 to Setup Disk 2.

Those who would be using Setup Disk 2 exclusively in a system which has a high resolution screen installed will probably want to move the screen resolution programs from Setup Disk 1 to Setup Disk 2 also.

By the way, although it is by no means a necessity, a high resolution screen does enhance chart design on the screen. Large charts, seen in sections on a low-resolution screen, are reduced to fit the viewing area on high-resolution screens so that the whole chart can be viewed. A high-resolution screen heightens the accuracy of placement of data points and more clearly defines very small bar segments and pie slices. Curved lines are smoother, available shading types are increased, and all shading is finer grained on the high-resolution screen. In short the high-resolution screen is an optional feature that can take you one step further toward simplifying graph design.

## THE PACKAGE

We've only given you a glimpse of this package and a hint of its potential. The Business Graphics Analysis Pak can generate an arithmetic or geometric series of up to 100 values; change, insert, or delete data values; transform data by supplying a constant; fit a linear, quadratic or exponential curve to the data and project the curve; compute arithmetic

or geometric growth projection; consolidate data; and compute the logarithm of each data value.

Depending on your printer, you can set chart width from 1" to 10" and height from 3" to 8". You can use solid, dashed or dotted lines for curves and any character for plot points. Choose from three to six types of shading for each curve, bar, bar segment, and pie slice.

If statistics and the reporting of same play an important role in your workaday life, you owe it to yourself to investigate the features of this package. 

## Bit-Image Graphics For the LP VIII

Earl R. Kooi  
2196 Albright Avenue  
Upland, CA 91786

Thank you for including the article on "Radio Shack Dot-Addressable Printers" appearing on page 5 of the May 1983 issue. It has helped considerably in understanding the mechanism of use of bit-image graphics.

This program was developed to help analyze how the characters in the published program were formed and should be helpful in designing patterns with bit-image graphics.

The program gives in three successive sets of 5 columns each:

- (1) The CHR\$ value of the bit graphic.
- (2) 128th CHR\$ value.
- (3) A printout of three successive CHR\$(213) bit graphics to establish positioning.
- (4) A printout of three successive bit graphics corresponding to the CHR\$ value in the first column of the 5-column set.
- (5) The Binary representation of the last 7 bits of the data byte in reverse of the usual order.

Therefore, the 7 bits, when read from left to right, will represent the 7 possible dots in the bit graphic from top to bottom.

A separate analysis of the program is included. If other programming amateurs have as much difficulty in following published programs as I do, they will appreciate this.

```

10 'DOT ADDRESSABLE PRINTER BIT GRAPHICS PATTERNS
20 'EARL R. KOOI
30 '2196 ALBRIGHT AV
40 'UPLAND, CA 91786
50 'MAY 19, 1983
60 'TRS-80 MODEL III, LINE PRINTER VIII
70 LPRINT TAB(25)"BIT GRAPHICS, LINE PRINTER
  VIII"
80 LPRINT
  : LPRINT
100 FOR N = 128 TO 170
110 IF N < 138 X = 16
120 IF N > 137 X = 15
150 Q = N
200 LPRINT TAB(10)Q;TAB(X)(Q - 128);
  : LPRINT" ";
210 GOSUB 500
300 Q = Q + 43
310 LPRINT Q;
  : LPRINT(Q - 128);

```

```

  : LPRINT" ";
320 GOSUB 500
330 IF Q = 213 END
400 Q = Q + 43
410 LPRINT Q;
  : IF N < 142 LPRINT" ";
  : LPRINT(Q - 128); ELSE LPRINT (Q - 128);
420 LPRINT " ";
430 GOSUB 500
440 LPRINT CHR$(13);
450 NEXT N
500 LPRINT CHR$(18);
510 LPRINT STRING$(3,CHR$(213));
520 LPRINT STRING$(3,CHR$(128));
530 LPRINT STRING$(3,CHR$(Q));
540 LPRINT CHR$(30);
550 GOSUB 1000
560 LPRINT" ";
  : LPRINT A$ + B$ + C$ + D$ + E$ + F$ + G$;
570 LPRINT" ";
580 RETURN
1000 A = Q/2
1020 IF A > INT(A) THEN A$ = "1" ELSE A$ = "0"
1030 B = INT(A)/2
1040 IF B > INT(B) THEN B$ = "1" ELSE B$ = "0"
1050 C = INT(B)/2
1060 IF C > INT(C) THEN C$ = "1" ELSE C$ = "0"
1070 D = INT(C)/2
1080 IF D > INT(D) THEN D$ = "1" ELSE D$ = "0"
1090 E = INT(D)/2
1100 IF E > INT(E) THEN E$ = "1" ELSE E$ = "0"
1110 F = INT(E)/2
1120 IF F > INT(F) THEN F$ = "1" ELSE F$ = "0"
1130 G = INT(F)/2
1140 IF G > 1 THEN G$ = "1" ELSE G$ = "0"
1150 RETURN

```

### PROGRAM ANALYSIS

Line	Description
100	Range of CHR\$ values for the first column
110	Tab setting for (N-128) for N of 128 to 137
120	Tab setting for N values 138 to 170
150	Set Q=N to avoid messing up the N values
200	Print N and (N-128)
210	GOSUB to instruction for printing bit graphics
300	Add 43 to N for second set of 5 columns
310	See line 200
320	See line 210
330	End program when last 2nd column values for CHR\$(213) are printed
400	Add 43 to N for third set of 5 columns
410	Print N; provide proper spacing for (N-128)
440	Carriage return for next line
450	Get the next N
500	Shift to bit graphics
510	Print 3 CHR\$(213) bit graphics
520	Print 3 bit graphics spaces
530	Print 3 CHR\$(N) bit graphics
540	Escape from bit graphics
550	Go to calculation of binary representation of bit graphics
560	Print concatenation of strings of binary bits
580	Go back to 310, 400, or 440
1100-1140	Divide the CHR\$ value for the bit graphic by 2 and divide the integer of six successive quotients by 2. If there is no remainder, the bit is 0 (off); if there is a remainder, the bit is 1 (on). Since the bit values are in reverse order, this is just fine for reading left to right values as up to

down dots. The bit values are expressed as string values so they can be concatenated at line 560.

```
BASIC (Insert BASIC program name here, e.g.
      DOFILE) <ENTER>
Type in up to 63 Characters
<BREAK>
```

```
BUILD DRIVE2 <ENTER>
```

```
Hit BREAK to exit
Type in up to 63 Characters
CLS <ENTER>
Type in up to 63 Characters
PAUSE ***ENTER DISKETTE TO BE REPORTED IN DRIVE
      #1*** <ENTER>
Type in up to 63 Characters
DIR :1 (PRT) <ENTER>
Type in up to 63 Characters
FREE :1 (PRT) <ENTER>
Type in up to 63 Characters
BASIC (BASIC program name) <ENTER>
Type in up to 63 Characters
<BREAK>
```

# Notes on Previous Issues

## JANUARY 1984

### Time/Date Display

Stan Bass  
GPO Box 6063  
San Juan, Puerto Rico 00936

This modified program prints the date as "January 23, 1984 17:24". At the end of the year, the program line which contains the year (line 230) will need alteration.

```
10 Z=TIME
20 A$=STR$ Z
30 IF LEN A$=11 THEN GOTO 50
40 A$="0" + A$
50 B$=LEFT$(A$,2)
60 C$=MID$(A$,3,2)
70 D$=MID$(A$,5,2)
80 E$=MID$(A$,8,2)
90 IF B$="01" LET B$="JANUARY"
100 IF B$="02" LET B$="FEBRUARY"
110 IF B$="03" LET B$="MARCH"
120 IF B$="04" LET B$="APRIL"
130 IF B$="05" LET B$="MAY"
140 IF B$="06" LET B$="JUNE"
150 IF B$="07" LET B$="JULY"
160 IF B$="08" LET B$="AUGUST"
170 IF B$="09" LET B$="SEPTEMBER"
180 IF B$="10" LET B$="OCTOBER"
190 IF B$="11" LET B$="NOVEMBER"
200 IF B$="12" LET B$="DECEMBER"
230 PRINT B$;" ";C$;" ",1984;" ";D$;" ";E$
```

## FEBRUARY 1984

### Do File Directory

Mark Rife  
5405 North Charles Street  
Baltimore, MD 21210

I have received a few letters concerning my program "DO File Directory". People are having problems with TRSDOS "eating" memory while returning to BASIC. Listed below are alterations that will solve the problem:

Delete Lines 590-640 from the BASIC program.  
The DO files should be listed as:

```
BUILD DRIVE1 <ENTER>
```

```
Hit BREAK to exit
Type in up to 63 Characters
CLS <ENTER>
Type in up to 63 Characters
PAUSE ***ENTER DISKETTE TO BE REPORTED IN DRIVE
      #0*** <ENTER>
Type in up to 63 Characters
DIR :0 (PRT) <ENTER>
Type in up to 63 Characters
FREE :0 (PRT) <ENTER>
Type in up to 63 Characters
```

## Extended Play with Orchestra-90

Bryan Eggers  
Software Affair

The February 1984 Musical Notes column described how to patch ORCHUTIL/CMD to accept input from job files. These new patches are for Version 01.02.00, which is currently being shipped with Orchestra-90.

Note: Please refer to the article before applying these patches. They do not correct "bugs" in the software, but are used to create a special version of the program that accepts input from job files. Do not patch your master copy of ORCHUTIL/CMD. Apply them to a backup copy of ORCHUTIL/CMD that has been renamed UTIL/CMD.

```
PATCH UTIL/CMD (ADD=5AA2,FIND=9554,CHG=0000)
PATCH UTIL/CMD (ADD=5AAD,FIND=06,CHG=0C)
PATCH UTIL/CMD (ADD=5AC6,FIND=2430,CHG=2B00)
```

## Hearts for Valentine's Day

Henry H. Herrdegen  
7110 Matchette Road  
Windsor Ontario  
Canada N9J 2S3

I enjoyed this little program very much. It came just in time to punch it in and surprise my family.

But a little error crept in. It does not harm the printout. Line 100 should read:

```
100 'CHANGE LINES 140,160,170
```

After the first excitement settled down, I could not help myself, but to add a few wrinkles of my own. I thought a heading and an end greeting would be nice. Add "TAB(10)" after each LPRINT in lines 140, 160, and 170. Then make the following changes and/or additions.

```
60 PRINT "The picture will have the name of a
      special friend."
72 T% = 27-L/2
74 INPUT "Your name please, for the Greetings. ";
      B$
76 LPRINT TAB(T%) "For my special friend "; A$;
      "!"
78 LPRINT TAB(T%) STRING$((23+L),"-")
      : LPRINT "
      : LPRINT
320 LPRINT
      : LPRINT
```

```

: LPRINT
: L = LEN(B$)
: T% = 28-L/2
330 LPRINT TAB(5) "With greetings from "; B$; "!"

```

## MARCH 1984

### Balloon Dart Toss

An FC error occurs in line 170 when the program is run as shown. We were unaware of the problem with this program when published. We apologize for any inconvenience that you may have experienced.

The program will run properly if you make the following modifications.

Delete line 350.

Add the following lines:

```

5 T = 0
275 T = T+1

```

The following lines should read:

```

80 FOR E = 1 TO 10
160 B$ = "BM"+STR$(X)+"",170;G10;D10;R10;U10;H10"
370 IF T = 10 THEN PRINT "YOU WON"
400 PRINT "YOU POPPED" T "BALLOONS"

```

## Bugs, Errors, and Fixes

The following changes and corrections are optional and provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that changes should be made, but you do not feel qualified to make the change yourself, contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Service in Fort Worth for assistance.

### Color Computer

#### OS9 'C' COMPILER (26-3038)

Page 1-2 First paragraph, last sentence.

Reads: A copy of this book, which serves as the language reference manual, is included with each software package.

Change: Use this book as the language reference manual for this software package.

### Model I/III/4

#### MODEL I/III STOCKPAK (26-1507 Vers. 03.00.00)

The list of SECURITIES held is incomplete when listed to the printer. This occurs when more than 72 have been entered into the system.

After making a Backup copy of the diskette, in BASIC load the program by typing LOAD"PCTRL:0". Change line 4952 to read:

```

4952 POKE 16425,1
: P8 = 0
: LPRINT" ";Q$;"1"
: LPRINT" "
: FOR I = 2 TO 4
: J = 0

```

#### MODEL III ACCOUNTS PAYABLE (26-1554 Vers. 03.01.00)

Check number does not increment for voided checks.

Backup the diskette and make changes on the Backup copy of the program. In BASIC load the program by typing, LOAD"CHECKS". Change lines 111 and 123 to read:

```

111 KC = 0
GOSUB 279
GET I,KR
: IF I(K) < 0 AND CVI(I0$) < 0 ANS CVI(I2$)
> 0 THEN IC = IC+1
: IF IC = 15 AND ABS(EP) = 2 THEN LPRINT L1$
: LC = LC+1
: GOSUB 271
: POKE 16425,1
: GOSUB 169
: IC = 1
: GOSUB 127 ELSE GOSUB 127 ELSE GOSUB 113

```

```

123 GOSUB 281
: IF ABS(EP) = 1 THEN GOSUB 153 ELSE IF
ABS(EP) = 2 THEN GOSUB 171 ELSE GOSUB 188

```

Add this line:

```

188 IF IC = 15 THEN CN = CN+1 LPRINT TAB(63)" **
VOIDED **"
: ID = 1 GOSUB 187

```

#### MODEL I/III PAYROLL (26-1556 Vers. 02.00.00)

Wisconsin State needs a 10% surcharge added to their PAYROLL program.

From a Backup copy of the program diskette, load the program, in BASIC, by typing LOAD"PAYROLL". Change line 5900 to read:

```

5900 E#(I) = (E#(I)*1.1#
: E#(I) = (E#(I)-CVS(MID$(N$,111,4)))/K#
: IF E#(I) > 0 THEN 6500 ELSE E#(I) = 0
: GOTO 6500

```

### Model II/12/16

#### MODEL II PAYROLL (26-4503 Vers. 02.00.00)

Washington State needs a display of four decimal places in Workman's Compensation.

After making a Backup copy of the disk, in BASIC load the program by typing LOAD"PAYROLL". Change lines 1500 and 2160 to read:

```

1500 CLS
: PRINT CF$
: PRINT@(2,8),R$"-- WORKMAN'S CO
M P C L A S S I F I C A T I O N S "N$
: F9$="###,###.###-###,###.##-
###,###.##-"
: PRINT@(4,40-LEN(A1$)/2),A1$

```

```

2160 LPRINT TAB(6);"-- WORKMAN'S COM
P C L A S S I F I C A T I O N S "
: F9$="###,###.###-###,###.##-
###,###.##-"
: LPRINT

```

# Calendar Program for the PC-2

H. B. Franklin  
2061 Lakeman Drive  
Bellbrook, Ohio 45301

The calendar computations in this program are a modified "Days Dates" program for the PC-2 from Radio Shack's Business Finance package (26-3703). This program fits the PC-2 basic memory with room for minor modification. On line 720 put the YES NO over the F3 F4 keys and over the F5 F6 keys on line 740. By adding line 190:GRAPH:GOTO 280 a calendar will be printed without boxes. Add line 712 IF M = 12 GOTO 720 and line 715:GOTO 40 to print a full year without entering each month. This program will print every month from January 1753 to December 2200.

```

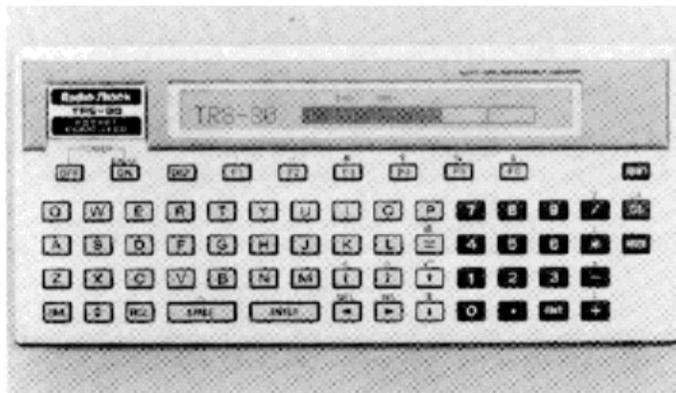
10 "PC"WAIT 75
   : CLS
   : CURSOR 8
   : PRINT "CALENDAR"
   : CLEAR
   : LOCK
20 INPUT " Mo/Yr ? ";G$
30 M=VAL LEFT$(G$,2)
   : Y=VAL RIGHT$(G$,4)
   : GOTO 50
40 M=M+1
   : IF M>12 LET M=1
   : Y=Y+1
50 WAIT 0
   : PRINT M;Y
60 IF M<1 OR M>12 OR Y<1753 OR Y>2200 WAIT 90
   : BEEP 3
   : PRINT " Out of range"
   : GOTO 20
70 IF M<3 LET M=M+12
   : Y=Y-1
80 M=M+1
   : J=INT(365.25*Y)-INT(Y/100)+INT(Y/400)+INT
     (30.6001*M)+1+1720997
90 W=J-7*INT(J/7)
   : WAIT 0
   : RESTORE 100+W
   : READ D$
100 DATA "M"
101 DATA "T"
102 DATA "W"
103 DATA "TH"
104 DATA "F"

```

```

105 DATA "S"
106 DATA "SU"
110 IF M>13 LET M=M-12
   : Y=Y+1
120 M=M-1
200 GRAPH
   : COLOR 0
220 LINE(0,-10)-(215,-234),,B
   : LINE(28,-10)-(56,-234),,B
230 LINE(84,-10)-(112,-234),,B
   : LINE(140,-10)-(168,-234),,B
240 LINE(185,-10)-(185,-234)
250 LINE(0,-170)-(185,-202),,B
   : LINE(0,-106)-(185,-138),,B
260 LINE(0,-42)-(185,-74),,B
   : LINE(0,-10)-(215,-234),,B
280 ROTATE 1
   : COLOR 1
290 H=31
   : D=1
   : IF M=2 LET H=28
300 IF M=4 OR M=6 OR M=9 OR M=11 LET H=30
320 K=(Y/100-INT(Y/100))*100
330 FOR L=4 TO 96 STEP 4
340 IF M=2 AND K=LLET H=29
350 NEXT L
360 IF M=2 AND Y=2000 LET H=29
370 X=148
380 IF D$="SU" LET Z=-4
390 IF D$="M" LET Z=-36
400 IF D$="T" LET Z=-68
410 IF D$="W" LET Z=-100
420 IF D$="TH" LET Z=-132
430 IF D$="F" LET Z=-164
440 IF D$="S" LET Z=-196
470 GLCURSOR(X,Z)
   : LPRINT D
480 IF Z=-196 LET Z=-4
   : X=X-28
   : GOTO 500
490 Z=Z-32
500 IF D=H GOTO 520
510 D=D+1
   : GOTO 470
520 WAIT 0
   : RESTORE 530+M
   : READ M$
531 DATA "JANUARY"
532 DATA "FEBRUARY"
533 DATA "MARCH"
534 DATA "APRIL"
535 DATA "MAY"
536 DATA "JUNE"
537 DATA "JULY"
538 DATA "AUGUST"
539 DATA "SEPTEMBER"
540 DATA "OCTOBER"
541 DATA "NOVEMBER"
542 DATA "DECEMBER"
550 IF M=1 OR M=10 LET V=-49
560 IF M=3 OR M=4 LET V=-64
570 IF M=5 LET V=-76
580 IF M=2 OR M=11 OR M=12 LET V=-46
590 IF M=6 OR M=7 LET V=-70
600 IF M=8 LET V=-60
610 IF M=9 LET V=-42
620 COLOR 2
   : GLCURSOR(193,V)
   : LPRINT M$;Y
630 X=174
   : COLOR 3
   : CSIZE 1
640 GLCURSOR(X,-18)
   : LPRINT "SUN"
650 GLCURSOR(X,-50)
   : LPRINT "MON"

```



```

660 GLCURSOR(X,-82)
  : LPRINT "TUE
670 GLCURSOR(X,-113)
  : LPRINT "WED
680 GLCURSOR(X,-145)
  : LPRINT "THU
690 GLCURSOR(X,-178)
  : LPRINT "FRI
700 GLCURSOR(X,-210)
  : LPRINT "SAT
710 GLCURSOR(0,-300)
720 WAIT 0
  : PRINT "Next Mo ? YES NO
730 GOTO 750
740 WAIT 0
  : PRINT "Another date ?   YES NO
750 F=ASC INKEY$
  : IF F<19OR F>22 THEN 750
760 IF F=19 GOTO 40
770 IF F=20 GOTO 740
780 IF F=21 CLEAR
  : GOTO 20
790 TEXT
  : COLOR 0
  : UNLOCK
  : LF 4
  : END

```

```

689 COLOR 3
  : LPRINT "ESTIMATED SALES."
690 COLOR 2
  : LPRINT "ACTUAL SALES."
  : LF 5
  : END

```

### EXAMPLE INPUT

Prompt	Input	Example 1	Example 2
DATA PTS./LINE (MAX=35)=		5	5
INPUT # OF LINES:		2	2
INPUT LARGEST VALUE:		100	5
INPUT SMALLEST VALUE:		1	1
INPUT LINE STYLE (0->9):		1	5
COLOR=(0BLK1BLU2GRN3RED):		3	3
INPUT VALUE:		30	2
INPUT VALUE:		40	4
INPUT VALUE:		60	1
INPUT VALUE:		70	5
INPUT VALUE:		75	3
INPUT LINE STYLE (0->9):		4	1
COLOR=(0BLK1BLU2GRN3RED):		2	2
INPUT VALUE:		60	5
INPUT VALUE:		60	3
INPUT VALUE:		70	4
INPUT VALUE:		70	1
INPUT VALUE:		85	2
DO YOU WANT Y-GRADS.?		Y	Y
HOW MANY Y-GRADS.?		5	5
INPUT Y-GRADS.:		20	1
INPUT Y-GRADS.:		40	2
INPUT Y-GRADS.:		60	3
INPUT Y-GRADS.:		80	4
INPUT Y-GRADS.:		100	5
DO YOU WANT X-GRADS.?		Y	Y
INPUT X-GRADS.:		JAN	JAN
INPUT X-GRADS.:		FEB	FEB
INPUT X-GRADS.:		MAR	MAR
INPUT X-GRADS.:		APR	APR
INPUT X-GRADS.:		MAY	MAY
IS THIS DOME TEMPS (Y-N)?		Y	-
IS THIS SALES 1984 (Y-N)?		-	Y

# Super Graph

Richard B. Drumm  
 WVIR-TV 29  
 P.O. Box 769  
 Charlottesville, VA 22902

*Editor's Note: Mr. Drumm uses Grads as the unit of linear measurement for X-Y axes in his program. GRADS can also be used as a unit of measurement with circles. (See your PC-2 Manual and other references for additional information.)*

This program will take a set of up to 35 numbers and graph it. You can run numerous lines on top of each other using various colors and line styles. You must enter the largest and smallest numbers from the set so that the PC-2 can scale the graph to the size of the box outline.

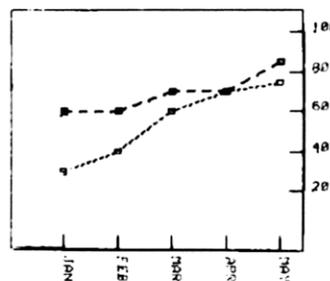
You can have graduations on the X and/or Y ordinates, but it is up to you to decide how many graduations (Grads) you will have on the vertical (Y) side, and you will have to tell the PC-2 which Grads you want. The horizontal (X) Grads will line up under the data points, so you won't need to tell it how many there will be, but you will have to tell it what label to attach there. The label can be alpha or numeric, as it is loaded into a string.

You can print a description of a two line graph. However, the labels on your horizontal (X) side cannot be over four characters long. The two lines must be input as RED (Color 3) and GREEN (Color 2) to match the description lines that follow. Change lines 685, 688, 689, and 690 to correspond to the graph you print. For Example 2 output change the lines to:

```

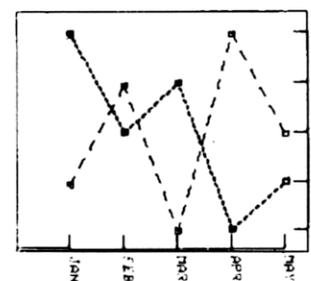
685 INPUT "IS THIS SALES 1984 (Y-N)?" ; I$
688 TEXT
  : CSIZE 1
  : COLOR 0
  : LF 4
  : LPRINT "SALES 1984."

```



DOME TEMPERATURE.  
 OBJECTIVE OUTSIDE TEMPERATURE.  
 OBJECTIVE INSIDE TEMPERATURE.

Example 1



SALES 1984.  
 ESTIMATED SALES.  
 ACTUAL SALES.

Example 2

In Example 1 the dashed line is the GREEN line. In Example 2 the dashed line is the RED line.

```

500 REM SUPER GRAPH -- BY RICHARD B. DRUMM WVIR
  TV-29 P.O. BOX 769 CHARLOTTESVILLE
501 REM THIS PROGRAM WILL TAKE A SET OF UP TO 35
  NUMBERS AND GRAPH IT. YOU CAN RUN

```

```

502 REM NUMEROUS LINES ON TOP OF EACH OTHER USING
    VARIOUS COLORS & LINE STYLES.
503 REM YOU MUST ENTER THE LARGEST & SMALLEST
    NUMBERS FROM THE SET SO THAT PC-2
504 REM CAN SCALE THE GRAPH TO THE SIZE OF THE
    BOX OUTLINE. YOU CAN HAVE
505 REM GRADUATIONS ON THE X AND/OR Y ORDINATES,
    BUT IT IS UP TO YOU TO DECIDE HOW
506 REM MANY GRADUATIONS (GRADS) YOU WILL HAVE ON
    THE VERTICAL (Y) SIDE, AND YOU
507 REM WILL HAVE TO TELL PC-2 WHICH GRADS YOU
    WANT. THE HORIZONTAL (X) GRADS WILL
508 REM LINE UP UNDER THE DATA POINTS, SO YOU
    WONT NEED TO TELL IT HOW MANY THERE
509 REM WILL BE, BUT YOU WILL HAVE TO TELL IT
    WHAT LABEL TO ATTACH THERE. THE
510 REM LABEL CAN BE ALPHA OR NUMERIC, AS IT IS
    LOADED INTO A STRING.
511 REM -----
600 "SUPER GRAPH"
    : LF 7
    : GRAPH
    : GLCURSOR (0,0)
    : SORGN
602 INPUT "DATA PTS./LINE (MAX=35)=";A
603 DIM Z(A)
    : INPUT "INPUT # OF LINES:";H
604 INPUT "INPUT LARGEST VALUE:";L
    : INPUT "INPUT SMALLEST VALUE:";S
606 FOR N = 1 TO H
607 INPUT "INPUT LINE STYLE (0->9)";C
608 INPUT "COLOR=(0BLK1BLU2GRN3RED)";B
609 INPUT "INPUT VALUE:";U
    : X = 175/A
    : D = L - S
610 T = (U - S) * (131/D)
619 GLCURSOR (X - 2,T - 2)
    : LINE (X - 2,T - 2)-(X + 2,T + 2),0,B,B
    : GLCURSOR (X,T)
620 FOR M = 1 TO A - 1
624 INPUT "INPUT VALUE:";U
    : T = (U - S) * (131/D)
625 X = X + (175/A)
    : COLOR B
628 LINE -(X,T),C
    : GLCURSOR (X + 2,T + 2)
629 LINE (X + 2,T + 2)-(X - 2,T - 2),0,B,B
    : GLCURSOR (X,T)
630 NEXT M
631 GLCURSOR (0,0)
632 NEXT N
634 GLCURSOR (0,-15)
    : LINE (0,-15)-(190,146),0,0,B
635 INPUT "DO YOU WANT Y-GRADS.?" ;I$
    : IF I$ = "Y" THEN 637
636 GOTO 650
637 INPUT "HOW MANY Y-GRADS.?" ;R
638 DIM Y(R)
639 FOR V = 1 TO R
640 INPUT "INPUT Y-GRAD.:" ;Y(V)
642 NEXT V
643 GLCURSOR (190,-15)
    : J = 0
645 FOR V = 1 TO R
    : J = (Y(V) - S) * (131/D)
646 LINE -(190,J)-(180,J)-(190,J)
647 CSIZE 1
    : LPRINT Y(V)
    : GLCURSOR (190,J)
648 NEXT V
650 INPUT "DO YOU WANT X-GRADS.?" ;I$
    : IF I$ = "Y" THEN 652
651 TEXT
    : END
652 GLCURSOR (0,-15)
    : X = 0

```

```

653 FOR V = 1 TO A
    : X = X + (175/A)
654 LINE -(X,-15)-(X,-5)-(X,-15)
655 INPUT "INPUT X-GRAD.:" ;I$
656 ROTATE 1
    : LPRINT " ";I$
    : GLCURSOR (X,-15)
657 NEXT V
684 GLCURSOR (0,-15)
685 INPUT "IS THIS DOME TEMPS (Y-N)?" ;I$
686 IF I$ = "Y" THEN 688
687 TEXT
    : COLOR 0
    : LF 5
    : END
688 TEXT
    : CSIZE 1
    : COLOR 0
    : LF 4
    : LPRINT "DOME TEMPERATURE."
689 COLOR 3
    : LPRINT "OBJECTIVE OUTSIDE TEMPERATURE."
690 COLOR 2
    : LPRINT "OBJECTIVE INSIDE TEMPERATURE."
    : LF 5
    : END

```

## Italian Flag

Fabio Marzocca  
Via delle  
Baleniere n.20  
00121 Ostia Lido  
Rome Italy

This PC-2 program draws the Italian flag and plays the national hymn "Fratelli D'Italia" with the internal buzzer.

```

10 GRAPH
    : LINE (0,0)-(210,-5),,0,B
20 GLCURSOR (200,-5)
    : SORGN
    : LINE (0,0)-(-104,-180),,0,B
30 FOR I=0 TO 60 STEP 2
    : LINE (0,-I)-(-104,-I),,1
    : NEXT I
40 FOR I=120 TO 180 STEP 2
    : LINE (0,-I)-(-104,-I),,3
    : NEXT I
50 TEXT
60 "A" FOR I=1 TO 2
70 BEEP 2,122,180
    : BEEP 1,110,100
    : BEEP 1,122,400
80 BEEP 2,70,350
    : BEEP 1,65,150
    : BEEP 1,70,550
90 BEEP 1,70,400
    : BEEP 1,58,300
    : BEEP 1,65,200
    : BEEP 1,70,550
    : IF I=2 THEN 120
100 BEEP 1,80,400
    : BEEP 1,70,300
    : BEEP 1,80,150
110 BEEP 1,90,450
    : NEXT I
120 BEEP 1,58,200
    : BEEP 1,80,200
    : BEEP 1,70,150
    : BEEP 1,90,500
130 END

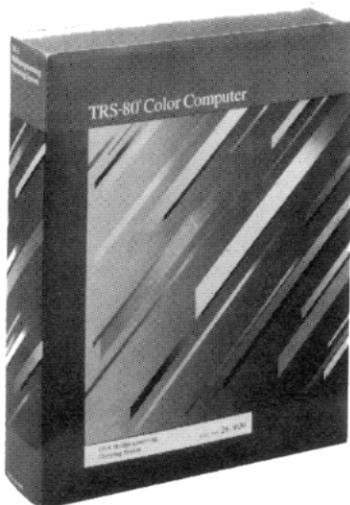
```

# OS9 Assembly Language

Earl W. Bollinger

It is sometimes useful to be able to develop specific device driver software for the OS9 environment. But it can be difficult to write a device driver without some place to start from. The purpose of this article is to provide some software tools that will allow you to be able to develop Sequential Character File (SCF) device driver software more easily. Your OS9 Technical Reference manual provides more detailed information about these drivers in Chapter Six.

A SCFman device driver is primarily designed to perform the actual character by character data transfers for OS9. It handles all the actual details of input/output to or from a specific hardware device or controller. The SCF drivers are usually designed so as to be re-entrant. Thus one copy of the driver can handle all identical devices resulting in the minimum use of available memory.



Program Example 1 demonstrates the typical structure of a basic SCF device driver module. This particular program listing is actually a template or boilerplate used to start the development of custom SCF driver modules.

The SCF driver must have a header, body and cyclic redundancy checksum (CRC) value in order to be a valid OS9 module. Otherwise, OS9 will simply refuse to load the module into memory or execute it. You must use the OS9 assembler definition files, located in the /D0/DEFS directory, in order to interface to the OS9 system properly. The dispatch jump table provides the six major functions that have to be provided by the driver for OS9.

INIT has to provide for two basic functions. First, INIT has to initialize the device or program it for later use. Second, if required, INIT must install an interrupt service routine into OS9 for that specific device. READ simply gets or reads the next character from the device or input buffer. WRITE outputs a character via the device or output buffer. GETSTAT and SETSTAT provide a means whereby the device's status can

be obtained or changed. TERM is used to de-allocate or terminate the device from the system when it is no longer needed. TERM must also remove the interrupt routine from the OS9 system if it was installed earlier.

Program Example 2 demonstrates the typical structure of a SCF based NULL device driver. Program Example 3 is the SCF device path descriptor module for the NULL device driver.

A null device is a very useful tool for multitasking types of operations. Any data redirected to a null device will simply be thrown away or ignored. This is useful for background tasks that would otherwise clutter up the console screen with extraneous junk or data. For example:

```
OS9:ASM /D2/PROGRAM L O = /D2/  
PROG.OBJ & (ENTER)
```

causes OS9 to assemble the program as a concurrent background task. But the output listing would also be to the screen. This would cause problems if you wanted to use the text editor or some other program at the same time. But if you used a null device like this:

```
OS9:ASM /D2/PROGRAM L O = /D2/  
PROG.OBJ >/NL & (ENTER)
```

then the output listing would simply disappear, and only error messages would get displayed on your screen.

To use the device driver as a null device is relatively simple. First a null device does not require any initialization and it should always be ready to ignore anything sent to it. Thus you only need to have INIT, WRITE, GETSTAT, PUTSTAT, and TERM return to OS9 the fact that the null device is always ready. If OS9 requests a READ from the null device you can have it always return an End of File (EOF) condition.

In order to use the null device driver, you must set up an appropriate device path descriptor. In this case you will call the path descriptor "NL". To then use the null device, you simply redirect all output to it with a ">/NL" redirection command. Of course, before you can use the driver and path descriptor you must have them loaded into memory using the LOAD command.

The /D0/DEFS/OS9defs and /D0/DEFS/SCFdefs system definition files require a large amount of memory to be needed by the OS9 assembler. When you assemble the programs, you should allocate about 15k bytes of memory for the assembler to utilize. For example:

```
OS9:ASM /D1/NULLDRIVER.SRC L O = /D0/CMDS/  
NULLDRIVER #15k (ENTER)
```

would cause the assembler to assemble the program and install the object module in the execution directory, also allocating 15k bytes of memory for the assembler's symbol table storage area.

The Color Computer does offer some interesting possibilities for those of you who want to address these items. If you have a Radio Shack Multi-Pak Interface (26-3024) and a Deluxe RS-232 Program Pak™ (26-2226) you can design your own custom SCF serial port driver to use with OS9. The Multi-Pak Interface is designed for switching Program Paks via software.

Pin 8 on the Color Computer's Program Pak connector provides for an interrupt line that can be used to advantage. This line goes to pin 18 of the U4 PIA chip. U4 is a MC6821 PIA device and pin 18 allows for a Fast Interrupt Request

(FIRQ) to be sent to the MC6809 microprocessor. Thus if your own device driver's interrupt routine is installed into OS9, you then can get to your serial port via software switching of the Multi-Pak Interface.

You could have more than one Deluxe RS-232 Program Pak™ in the interface also. By using the path descriptor serial port address location (offset 0Ehex) as a Program Pak slot select value, you could use the same device driver for more than one port. The path descriptor DVINIT byte can be used to program each port as required for each application.

**Program Example 1**

```

00001
00002          NAM   SCFDRIVER
00003          TTL   TEMPLATE
00004          *****
00005          *
00006          *     SCF DEVICE DRIVER TEMPLATE
00007          *
00008          *****
00009          *
00010          *
00011          * USE /D0/DEFS/OS9DEFS AND USE /D0/DEFS/SCFDEFS ARE
00012          * PUT BETWEEN THE IFPI AND ENDC COMMANDS
00013          *
00014          IFPI
00017          ENDC
00018          *
00019  00E1          TYPE   SET   DRIVR+OBJECT
00020  0081          REVS   SET   REENT+1
00021  0000 87CD02B MOD   DRVEND,DRVNAM,TYPE,REVS,DRSTRT,SIZE
00022  000D 03          FCB   UPDAT.
00023  000E 53434644  DRVNAM FCS  /SCFDRIVR/
00024  0015 01          FCB   1           EDITION NUMBER
00025          *
00026          *
00027  D 001D          ORG   V.SCF       ALLOW ROOM FOR SCF VARIABLES
00028          *
00029          * IF DESIRED, ADD ADDITIONAL VARIABLE
00030          * TEMPORARY STORAGE HERE.
00031          *
00032  0016          SIZE   EQU   *
00033          *
00034          * SCFDRIVER ENTRY POINT HERE
00035  0016          DRSTRT
00036          *
00037          *   DISPATCH TABLE FOR DEVICE DRIVER
00038          *
00039  W 0016 16000F  LBRA  INIT       INITIALIZE DEVICE
00040  W 0019 16000C  LBRA  READ       GET A CHARACTER FROM
00041  W 001C 160009  LBRA  WRITE      PUT A CHARACTER TO
00042  W 001F 160006  LBRA  GETSTAT    GET DEVICE STATUS
00043  W 0022 160003  LBRA  PUTSTAT    SET DEVICE STATUS
00044  W 0025 160000  LBRA  TERM       TERMINATE THE DEVICE
00045          *
00046          * INSERT APPROPRIATE ROUTINES FOR EACH
00047          * SECTION AS REQUIRED

```

```

00048      *
00049  0028      INIT
00050  0028      READ
00051  0028      WRITE
00052  0028      GETSTAT
00053  0028      PUTSTAT
00054  0028      TERM
00055  0028 F8E1EA      EMOD
00056  002B      DRVEND  EQU  *
00057                      END

```

```

00000 error(s)
00006 warning(s)
$002B 00043 program bytes generated
$0000 00000 data bytes allocated
$2050 08272 bytes used for symbols

```

Program Example 2

```

00001
00002          NAM  NULLDRIVER
00003          TTL  FROM TEMPLATE
00004  *****
00005  *
00006  *      SCF NULL DEVICE DRIVER
00007  *
00008  *****
00009  *
00010  *
00011  * USE /D0/DEFS/OS9DEFS AND USE /D0/DEFS/SCFDEFS ARE
00012  * PUT BETWEEN THE IFPI AND ENDC COMMANDS
00013  *
00014          IFPI
00017          ENDC
00018  *
00019  00E1      TYPE  SET  DRVR+OBJCT
00020  0081      REVS  SET  REENT+1
00021  0000 87CD0034      MOD  DRVEND, DRVNAM, TYPE, REVS, DRSTRT, SIZE
00022  000D 03          FCB  UPDAT.
00023  000E 4E554C4C      DRVNAM  FCS  /NULLDRVR/
00024  0016 01          FCB  1          EDITION NUMBER
00025  *
00026  *
00027  D 001D          ORG  V.SCF      ALLOW ROOM FOR SCF VARIABLES
00028  *
00029  * IF DESIRED, ADD ADDITIONAL VARIABLE
00030  * TEMPORARY STORAGE HERE.
00031  *
00032  0017          SIZE  EQU  *
00033  *
00034  * SCFDRIVER ENTRY POINT HERE
00035  0017          DRSTRT
00036  W 0017 16000F      LBRA  INIT
00037  W 001A 16000E      LBRA  READ
00038  W 001D 160009      LBRA  WRITE
00039  W 0020 160006      LBRA  GETSTAT
00040  W 0023 160003      LBRA  PUTSTAT
00041  W 0026 160000      LBRA  TERM

```

```

00042      *
00043      * INSERT APPROPRIATE ROUTINES FOR EACH
00044      * SECTION AS REQUIRED
00045      *
00046      0029          INIT
00047      0029          WRITE
00048      0029          GETSTAT
00049      0029          PUTSTAT
00050      0029          TERM
00051      *
00052      0029 5F          CLRB          RETURN A ZERO CODE
00053      002A 39          RTS          DON'T DO ANYTHING
00054      *
00055      *
00056      002B          READ
00057      002B 4F          CLRA
00058      002C 1A01        ORCC #01          SET CARRY FLAG
00059      002E C6D3        LDB #E$EOF        SEND BACK EOF
00060      0030 39          RTS
00061      *
00062      *
00063      0031 7120A6      EMOD
00064      0034          DRVEND EQU *
00065      END

```

```

00000 error(s)
00006 warning(s)
$0034 00052 program bytes generated
$0000 00000 data bytes allocated
$2050 08272 bytes used for symbols

```

Program Example 3

```

00001          NAM      NL
00002          TTL      NULL DEVICE PATH DESCRIPTOR
00003      *****
00004      *          *
00005      * SERIAL PORT DEVICE DESCRIPTOR *
00006      *          *
00007      * USED WITH NULL DEVICE DRIVER *
00008      *          *
00009      *****
00010      *
00011      0003          UPDATE SET 03
00012      00F0          DEVICE SET $F0
00013      0001          OBJECT SET $01
00014      0080          REENT SET $80
00015      *
00016      * MODULE HEADER INFORMATION
00017      *
00018      00F1          TYPE      SET      DEVICE+OBJECT
00019      0000 87CD03D          MOD      DVEND, DVNAM, TYPE, REENT+1, SCFNAM, DRVNAM
00020      000D 03          FCB      UPDATE
00021      000E 000000          FCB      $00, $00, $00 ADDRESS OF SERIAL PORT
00022      0011 1B          FCB      OPTSIZ
00023      0012          OPTBGN EQU *
00024      *

```

```

00025      * PATH DESCRIPTOR OPTION LIST
00026      *
00027  0012 00      CLASS      FCB  00      SCF TYPE DESIGNATION
00028  0013 00      UPC        FCB  00      UPPER/LOWER CASE
00029  0014 01      BSE       FCB  01      BACKSPACE ECHO BS, SP, BS
00030  0015 00      DCLRL    FCB  00      DELETE OR CLR LINE
00031  0016 01      ECHO     FCB  01      ECHO CHARS
00032  0017 00      AUTOLF   FCB  00      AUTO LINE FEED
00033  0018 00      NULL    FCB  00      NO END OF LINE NULLS
00034  0019 00      PAUSE   FCB  00      END OF LINE PAUSE OFF
00035  001A 10      PAGE    FCB  16      LINES PER PAGE
00036  001B 08      BKSPC   FCB  08      BACKSPACE CHAR
00037  001C 18      DELETE  FCB  24      DELETE LINE CHAR
00038  001D 0D      EOFREC  FCB  13      END OF RECORD (LINE)
00039  001E 1B      ENDFIL  FCB  27      END OF FILE CHAR
00040  001F 04      RPRLL   FCB  04      REPRINT LAST LINE
00041  0020 01      DUPE    FCB  01      DUPLICATE LAST LINE
00042  0021 17      PSCHAR  FCB  23      PAUSE CHAR
00043  0022 03      INTRUP  FCB  03      KEYBOARD INTERRUPT CHAR
00044  0023 15      ABORT   FCB  $15     KEYBOARD ABORT CHAR
00045  0024 08      BKSPE   FCB  08      BACKSPACE ECHO CHAR
00046  0025 07      OVRFLW  FCB  07      BELL-LINE OVERFLOW CHAR
00047  0026 00      DVINIT  FCB  00      DEVICE INITIALIZATION VALUE
00048  0027 02      BAUD    FCB  02      BAUD RATE VALUE
00049  0028 002D    DV2     FDB  DVNAM   OFFSET TO ATTACHED DEVICE NAME
00050  002A 0000    STATUS  FDB  00      OFFSET TO DEVICE STATUS ROUTIN
00051  002C 00      ERROR   FCB  00      I/O ERROR CODE STATUS
00052  001B      OPTSIZ  EQU  *-OPTBGN
00053      *
00054      *
00055  002D 4ECC    DVNAM   FCS  /NL/     DEVICE PATH NAME
00056  002F 5343C6  SCFNAM  FCS  /SCF/   DEVICE MANAGER NAME
00057  0032 4E554C4C  DRVNAM  FCS  /NULLDRVR/ SERIAL PORT DRIVER NAME
00058  003A 645CBB      EMOD    FCB  MODULE CRC VALUE
00059  003D      DVEND   EQU  *
00060      END

00000 error(s)
00000 warning(s)
$003D 00061 program bytes generated
$0000 00000 data bytes allocated
$0250 00592 bytes used for symbols

```

## Squares and Circles

Karen Salvini  
137 Trapelo Road  
Lincoln, MA 01773

This program will make some really beautiful designs. It draws concentric squares and circles on a 16K Extended Color Computer. The circles and squares vary in width, and the hi-res white gives the designs a lot of color, like bright pink, en, and even a little purple.

```

10 FOR N = 1 TO 2000
  : NEXT
  : R = RND(40)
  : R = R/5

```

```

20 X = 128
  : Y = 96
30 A = 128
  : B = 96
40 PMODE 4,1
  : SCREEN 1,1
  : PCLS
50 X = X+R
  : Y = Y+R
  : A = A-R
  : B = B-R
60 IF A<0 OR B<0 OR X>256 OR Y>191 THEN 100
70 LINE(A,B)-(X,Y),PSET,B
80 GOTO 50
100 FOR D = 1 TO 100 STEP RND(20)/5
110 CIRCLE(128,96),D
120 NEXT D
130 GOTO 10

```

# Drawing With Joysticks

Will Yeckley  
1182 Bannister Circle  
Waldorf, MD 20601

This program allows you to use the Color Computer's right joystick to select any desired screen location and set points to draw lines and figures. The keyboard is used for enhancement features such as circle and paint. PMODE 3 is used to provide colorful high resolution graphics allowing the screen to be divided into 12 sectors. The joystick draws in one sector at a time and then the movement keys are used to move the cursor to the next sector.

A description of the keyboard controls is given below. With a little practice, you will be easily producing artistic pictures with your Color Computer.

## COLOR SELECTION

Key	Color
<b>G</b>	Green (used to erase)
<b>Y</b>	Yellow
<b>B</b>	Blue (automatically selected when the program is run.)
<b>R</b>	Red

## CURSOR MOVEMENT

Key	Function
<b>E</b>	Move to adjacent sector East of current position.
<b>W</b>	Move to adjacent sector West of current position.
<b>N</b>	Move to adjacent sector North of current position.
<b>S</b>	Move to adjacent sector South of current position.
<b>M</b>	Flashes cursor to move from one location to another within a sector. Hit <b>M</b> again to exit this mode.
<b>P</b>	Paints bound area with current color.
<b>C</b>	Sets location for circle center with current cursor position. Center must be set first and then the <b>D</b> key is used.
<b>D</b>	Sets circle radius as the distance from the current cursor position to the location selected by the <b>C</b> key.
<b>A</b>	Sets location for start of line with current cursor position. Start point is set first and then the <b>Z</b> key is used.
<b>Z</b>	Sets location for last point of line with current cursor position. Line is automatically drawn.

## Program

```
5 REM COLOR DRAW WITH JOYSTICK
10 PMODE3,1
   : SCREEN1,0
   : PCLS
15 C = 3
20 H = JOYSTK(0)
   : V = JOYSTK(1)
22 K$ = INKEY$
   : IF K$ <> "" THEN GOTO 50
24 PSET(H+63*X, V+63*Y, C)
   : GOTO 20
```

```
50 SOUND 100,5
51 IF K$ = "G" THEN C = 1
52 IF K$ = "Y" THEN C = 2
53 IF K$ = "B" THEN C = 3
54 IF K$ = "R" THEN C = 4
55 IF K$ = "E" THEN GOTO 70
56 IF K$ = "W" THEN GOTO 72
57 IF K$ = "N" THEN GOTO 74
58 IF K$ = "S" THEN GOTO 76
60 IF K$ = "M" THEN GOTO 80
61 IF K$ = "P" THEN GOTO 90
62 IF K$ = "C" THEN GOTO 100
63 IF K$ = "D" THEN GOTO 102
64 IF K$ = "A" THEN GOTO 110
65 IF K$ = "Z" THEN GOTO 112
69 SOUND 150,5
   : GOTO 20
70 X = X+1
   : IF X >= 4 THEN X = 0
71 GOTO 20
72 X = X-1
   : IF X < 0 THEN X = 3
73 GOTO 20
74 Y = Y-1
   : IF Y < 0 THEN Y = 2
75 GOTO 20
76 Y = Y+1
   : IF Y >= 3 THEN Y = 0
77 GOTO 20
80 PSET(H+63*X, V+63*Y, C)
81 FOR T = 1 TO 10
   : NEXT
82 PRESET(H+63*X, V+63*Y)
83 H = JOYSTK(0)
   : V = JOYSTK(1)
84 K$ = INKEY$
   : IF K$ = "" THEN GOTO 80 ELSE GOTO 20
90 PRESET(H+63*X, V+63*Y)
91 PAINT(H+63*X, V+63*Y), C, C
   : GOTO 20
100 S1 = H+63*X
   : S2 = V+63*Y
   : GOTO 20
102 T1 = H+63*X
   : T2 = V+63*Y
104 U1 = S1-T1
   : U2 = S2-T2
   : R1 = SQR((U1*U1)+(U2*U2))
106 CIRCLE(S1, S2), R1, C
   : GOTO 20
110 A1 = H+63*X
   : A2 = V+63*Y
   : GOTO 20
112 B1 = H+63*X
   : B2 = V+63*Y
114 COLOR C, 1
   : LINE(A1, A2)-(B1, B2), PSET
   : GOTO 20
```



# Magic Color Square

Ryan  
1 Wren  
Mount Vernon, IL 62864

I hope your readers like this short, easy to input program for the MC-10 and the CGP-115 Color Graphics Printer. If you want to eliminate the lettering, delete line 11. Change the value of B in line 17 to increase or decrease the size of the square. In the example square B = 3.



```

10 CLS
11 GOSUB 780
15 A = 4
17 B = 3
20 REM
30 LPRINT CHR$(18)
40 LPRINT "M240,0"
45 LPRINT "I"
46 LPRINT "C";0
48 Q = 1
50 FOR N = 1 TO 10
60 LPRINT "J0,";A
70 LPRINT "J";A;"",0"
80 LPRINT "J0,";-A
90 LPRINT "J";-A;"",0"
100 LPRINT "M";-B;"",-B
105 LPRINT "I"
110 A = A+(B*2)
115 NEXT
500 C = C+1
505 Q = Q+1
510 IF C > 3 THEN C = 0
515 LPRINT "C";C
518 IF Q = 6 THEN GOTO 600
520 GOTO 50
600 LPRINT "M0,-15"
610 T$ = "MAGIC COLOR SQUARE"
620 LPRINT "Q0"
640 END
780 REM
790 LPRINT "A"
800 LPRINT CHR$(18)
801 LPRINT "C";0
805 LPRINT "M70,0"
810 T$ = "MAGIC COLOR SQUARE"
815 LPRINT "S";2.5

```

```

818 LPRINT "Q0"
819 LPRINT "C";3
820 LPRINT "P";T$
830 LPRINT "Q";1
835 LPRINT "C";1
840 LPRINT "P";T$
845 LPRINT "C";2
850 LPRINT "Q";2
860 LPRINT "P";T$
870 LPRINT "Q";3
875 LPRINT "C";0
880 LPRINT "P";T$
900 LPRINT "M-60,-125"
915 LPRINT "A"
920 RETURN

```

# Indian Blanket

Eric M. White  
Route 2 Box 63  
Cumberland, MD 21502

I wrote this random graphics program on my 4K Color Computer. If you watch the graphics with some favorite music it can be very entertaining.

The INPUT C in Line 10, will control how intricate the design on the screen will become before it is erased to start a new design. If you tape your favorite song directly behind the program, and add AUDIO ON and MOTOR ON statements, the program will turn on your music automatically.

```

5 CLS
10 INPUT"DESIGN NUMBER";C
30 CLS(0)
40 X = RND(64)-1
   : Y = RND(32)-1
60 FOR I = 1 TO C
   : D = RND(10)
   : FOR J = 1 TO RND(20)+5
70 ON D GOSUB 150,170,190,210,160,180,200,
   : 220,225,230
80 IF X<0 THEN X = 63
   : ELSE IF X>63 THEN X = 0
90 IF Y<0 THEN Y = 31
   : ELSE IF Y>31 THEN Y = 0
95 IF D>8 THEN D = RND(9)-1
100 SET(X,Y,D)
   : SET(63-X,Y,D)
   : SET(X,31-Y,D)
   : SET(63-X,31-Y,D)
110 NEXT J,I
   : GOTO 30
150 X = X+1
155 Y = Y+X
160 Y = Y+1
   : RETURN
170 Y = Y+1
180 X = X-1
   : RETURN
190 X = X-1
200 Y = Y-1
   : RETURN
210 Y = Y-1
220 X = X+1
   : RETURN
225 X = X*X
   : Y = Y*Y
   : RETURN
230 X = X-5
   : Y = Y-5
   : RETURN

```

# Vertical Variation

Stefan Cernusca  
Box A-46 Route 1  
Susquehanna, PA 18847

This is a short program but interesting for beginning Color Computer programmers.

```
0 CLSRND(8)
  : PRINT@96, " STEFAN CERNUSCA
    RDI,SUSQUEHANNA,PA
    BOX A-46 18847"
1 PRINT"IT IS A SMALL AND INSIGNIFICANT PROGRAM
  BUT INTERESTING FOR BEGINNERS."
3 PRINT
4 PRINT" PRINT'ENTER'"
  : INPUT A
10 SCREEN2,1
  : PMODE RND(2),1
20 Y = RND(10)+5
  : COLOR RND(4)+4
30 FOR X = 0 TO 255 STEP Y
  : LINE(X,0)-(X,191),PSET
  : NEXT
40 FOR X = 255 TO 0 STEP-Y
  : LINE(X,0)-(X,191),PSET
  : NEXT
  : GOTO 10
```

```
40 FOR G = 1 TO 1
45 FOR P = 1 TO 6
50 PM = 4
55 FC = 2
  : BC = 1
  : CS = RND(4)-1
60 PMODEPM,1
  : COLORFC,BC
  : PCLS
  : SCREEN1,CS-INT(CS/2)*2
65 BE = 1
  : XF = 255
  : YF = 191
70 N0 = 0
  : N1 = 1
  : N4 = 4
75 XR = 256
  : YB = 192
80 IL = 6
85 N = RND(150)+10
90 X = RND(XR)-N1
  : Y = RND(YB)-N1
95 X1 = RND(XR)-N1
  : Y1 = RND(YB)-N1
100 I1 = RND(IL)-N1
  : I2 = RND(IL)-N1
105 I3 = RND(IL)-N1
  : I4 = RND(IL)-N1
110 IF (I1=N0)+(I2=N0)+(I3=N0)+(I4=N0) < -N2 THEN
  100
115 Q1 = X
  : Q2 = Y
  : Q3 = X1
  : Q4 = Y1
  : Q5 = I1
  : Q6 = I2
  : Q7 = I3
  : Q8 = I4
120 GOSUB 180
125 X = X+I1
  : IF X >= XR OR X <= N0 THEN I1 = -I1
  : X = X+I1
  : CO = RND(N4)
130 Y = Y+I2
  : IF Y >= YB OR Y <= N0 THEN I2 = -I2
  : Y = Y+I2
  : CO = RND(N4)
135 X1 = X1+I3
  : IF X1 >= XR OR X1 <= N0 THEN I3 = -I3
  : X1 = X1+I3
  : CO = RND(N4)
140 Y1 = Y1+I4
  : IF Y1 >= YB OR Y1 <= N0 THEN I4 = -I4
  : Y1 = Y1+I4
  : CO = RND(N4)
145 IF BE = N0 THEN IF CO = BC THEN CO = RND(N4)
  : GOTO 145
150 COLORCO,BC
  : IF PM = N4 THEN COLORN2,N1
155 K$ = INKEY$
  : IF K$ = " " THEN 170
160 IF N<>N0 THEN N = N-N1
  : GOTO 120
165 FOR D = 1 TO 1000
  : NEXT
170 BC = RND(N4)
  : IF PM = 4 THEN BC = N1
175 COLORCO,BC
  : CLS8
  : NEXT
  : GOTO 20
180 LINE(X,Y)-(X1,Y1),PSET
185 LINE(XF-X,Y)-(XF-X1,Y1),PSET
190 LINE(X,YF-Y)-(X1,YF-Y1),PSET
195 LINE(XF-X,YF-Y)-(XF-X1,YF-Y1),PSET
200 RETURN
```

# Symmetry

Stefan Cernusca  
Box A-46 Route 1  
Susquehanna, PA 18847

This program is just for fun. By touching the **SPACEBAR** you can interrupt the designing and start a new design. The program demonstrates the Color Computer's potential for designing beautiful and steadily changing patterns.

```
0 CLSRND(8)
  : PRINT@64, STRING$(32,250);
  : PRINT@416, STRING$(32,250);
1 PRINT@128, " DEAR SIR,
  THIS PROGRAM IS JUST FOR FUN AND IF YOU
  FIND IT SUITABLE FOR PRINTING DO IT. BY
  TOUCHING THE SPACEBAR YOU CAN INTERRUPT THE
  DESIGNING AND START A NEW DESIGN"
  : PRINT" TYPE'ENTER' ";
  : INPUT A
5 CLSRND(8)
  : PRINT@128, STRING$(32,233);
10 PRINT@352, STRING$(32,233);
15 PRINT@192, " THE COMPUTER DEMONSTRATES HIS
  POTENTIAL IN DESIGNING BEAUTIFUL AND STEADY
  CHANGING PATTERNS."
  : PRINT" TYPE'ENTER' ";
  : INPUT A
16 GOTO 40
20 PRINT@128, STRING$(32,233);
25 PRINT@160, " THE COMPUTER WILL SHOW YOU SIX
  MORE BEAUTIFUL ORNAMENTS CREATED BY THE
  POTENTIAL OF HIS OWN IMAGINATION."
30 PRINT@288, STRING$(32,233);
35 FOR X = 1 TO 3500
  : NEXT
```

# Magnificent Micro Painter

by Maggie Sherrod-Stone

Being a great artist has always been one of my hidden desires. However, I have had to settle for being an admirer and not a participant. Well, no more! Thanks to Micro Painter (26-3077) and my CGP-220 Ink Jet Printer (26-1268) the true artiste in me has been released.

To paint with Micro Painter you will need a TRS-80 16K Color Computer, a color television, a cassette recorder, and connecting cables. A blank cassette tape is required to save your pictures. Joysticks are optional.

## GETTING STARTED

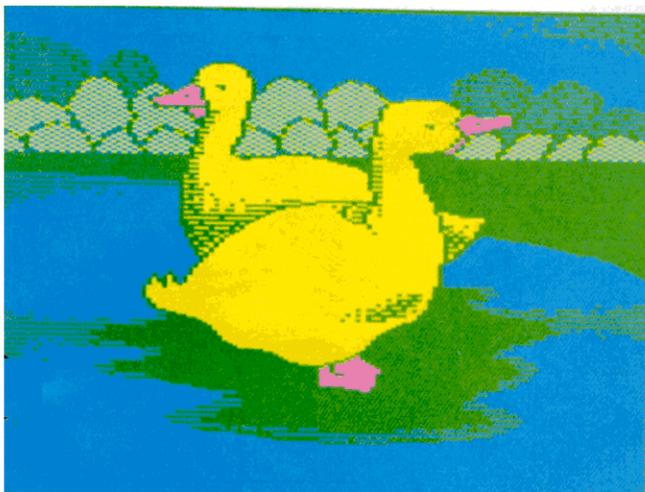
After loading Micro Painter you have the choice to paint your own picture or load a picture from the many data tapes compatible with Micro Painter. I chose to use Images I (26-3300) and Images II (26-3301) data tapes for my first try at Micro Painting. Each tape includes twelve different pictures. They are:

### IMAGES I

KNIGHT in Armor  
CASTLE  
DINOSAUR/Triceratops  
TREASURE Chest  
HORSE  
LION  
David and GOLIATH  
Christmas CANDLE  
  
TURKEY  
HOCKEY  
LILIES  
NEW YEAR/Father Time  
and Baby New Year

### IMAGES II

DINOSAUR/Brontosaurus  
TRAIN/Steam Locomotive  
RHINOCEROS  
DUCKS  
BUFFALO  
MOOSE  
CAMEL  
Skull and Cross Bones/  
XBONES  
FLOWERS  
SANTA CLAUS  
Backpacker/HIKER  
PUMPKIN



To load a picture, make sure the cassette recorder is correctly connected to the color computer, then press the **L** key and the program's Tape Loading function will begin. The screen will show:  
LOAD SUB-SYSTEM  
ENTER PICTURES NAME  
Enter the picture's name (the file names are in capital letters above) and press **ENTER**. In a few moments the picture will appear on the screen.

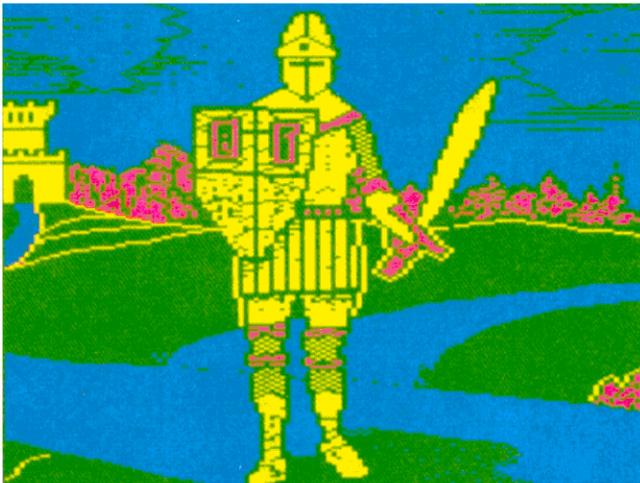
## CODE OF THE COLOR PRISM

Micro Painter provides three sets of colors, called prisms, for every picture you paint. When a picture first appears on the screen, it will be in Prism 3 colors, with a white background outlined in dark (black). Press **1** and the background color changes to red with outlines of green, for Prism 1. Press **2**, the background changes to orange with buff outlines. This is Prism 2.

Each prism has four colors that correspond to different colors in the other prisms. As you switch from one prism to another the colors will correspond as follows:

PRISM 1		PRISM 2		PRISM 3	
<b>G</b> reen	→	<b>B</b> uff	→	<b>D</b> ark (black)	
<b>Y</b> ellow	→	<b>C</b> yan	→	<b>B</b> lue	
<b>B</b> lue	→	<b>M</b> agenta	→	<b>R</b> ed	
<b>R</b> ed	→	<b>O</b> range	→	<b>W</b> hite	

Deciding which prism you wish to paint in is a matter of preference. If you change your mind in the middle of painting, you can switch to another prism without harming the picture. Just remember, the way in which the colors correspond from prism to prism may result in a blue sky in Prism 3 changing to yellow in Prism 1!



### PAINTING WITH THE MAGIC PAINTBRUSH

The blinking cross, otherwise known as the Magic Paintbrush, will appear when the picture has been loaded. You can maneuver the Paintbrush using the arrow keys or a Joystick. I prefer using the arrow keys because then you will be using the keyboard for all painting commands.

### CHOOSING A BACKGROUND

As with any picture you must first choose the surface on which to paint. Micro Painter is no exception. You must specify a background color or the painting functions will not work. The background color you select must be currently on the screen and the Magic Paintbrush must be over that color. Once the Paintbrush is positioned properly, press the **(B)** key (for Background), followed by the first letter of the color the Paintbrush is over.

### PAINTING—STRIPES, CHECKS, OR SOLIDS

One of Micro Painter's biggest assets is that it lets you paint in stripes or checks. To paint Stripes, press the **(S)** key. To paint Checks, press the **(C)** key. Next select two different colors from the prism you are in. Neither of the colors chosen can be the current background color. Enter the first letters of the two colors you selected. The final step is to press the **(P)** key to paint.

Micro Painter will not paint beyond the outlines of the area you choose to paint. If you wish to repeat the same pattern or color somewhere else on the screen, simply position the Paintbrush and press the **(P)** key again. If you are going to repeat a color or pattern several times go ahead and use the **(P)** key to fill in these areas. No sense in repeatedly entering the code for the same pattern or color over and over again.

For solids you have the option of either pressing **(C)** (Checks) or **(S)** (Stripes) followed by the first letter of a single color twice. Then press the **(P)** key to paint.

### CHANGING THE BACKGROUND COLOR

There are some striped color patterns which can only be produced if you color an area; change the background to that color, and then paint over the color with a new pattern of stripes. For example, in Prism 3 you cannot paint in striped white and red if the background color is white or red. You must change the background color.

To paint red and white stripes over a white background with black outlines, you would have to do the following:

1. Move the Paintbrush to the area you want to paint. Repaint the area blue.
2. Change the background color to blue by pressing the **(B)** key for Background, then the **(B)** key for Blue.
3. Press **(S)** for Stripes, followed by **(W)** for White, **(R)** for Red, then **(P)** to Paint.

Not only can changing the background color enable you to produce more patterns, you can use it to correct your mistakes. For example, if you accidentally painted an area red you could change the background to red and then change the area to the correct pattern or color.

### REPLACING COLORS

Replacing colors is a convenient method of changing existing colors. For example, to replace blue with red, you would press the **(R)** key (for Replace), then **(B)** (for Blue) and finally **(R)** (for Red). Using this function will change all specified pre-painted areas on the screen, including outlines. Be very careful when you decide to replace colors. Micro Painter will not stop at the outlines, it will replace the color wherever it appears on the screen.

Be especially careful when it comes to the outlines on the screen. For example, if you are painting in Prism 2 and you replace buff with orange, your entire picture may disappear leaving you with an orange screen. There is no way to reverse such a change. Your only course of action is to reload the picture and start over.

### THE PERISCOPIK-ELEKTRON-MAGNIFIER (P.E.M.)

Often, single dots will not be colored because they are isolated from the larger area by outlines. That is why Micro Painter has a giant paintbrush, the Periscopik-Elektron-Magnifier or P.E.M., which can be used to paint these isolated areas.

To activate the P.E.M. press the **(SPACEBAR)** and like magic a section of your screen is magnified. Instead of moving about the screen like the Paintbrush, the P.E.M. stays in the center while the picture moves. Use the arrow keys or the joystick to move to the areas you wish to paint. Press the **(SPACEBAR)** again and the entire picture will return to the screen.

To paint with P.E.M. simply press the first letter of the color, then press **(P)** for paint. Just like the Paintbrush, you



can continue to press (P) and repeat the color as many times as you need.

P.E.M. will not only take care of the isolated dots but it can and extra detail, paint over areas which have already been painted, change the outlines to different colors, develop new patterns and paint over stripes or checks.

The pictures included on the Images tapes often have detail lines which are difficult to paint using just the Paintbrush. The P.E.M. in combination with the Paintbrush enables you to fill in these detail areas fast and easily. Use the Paintbrush to move to the detail area, press the (SPACEBAR), then use the arrow keys to position the P.E.M. Once you are positioned, return to the entire picture by pressing the (SPACEBAR). Press the (P) key and the detail line will be painted. Continue with the (SPACEBAR), position, (SPACEBAR), (P) pattern until all detail lines have been completed.

### SAVING PICTURES ON TAPE

You can save your pictures on tape for further retouches, later viewing, or to load into the Hi-Res Screen Print Utilities program for printing.

After properly connecting your cassette recorder, press the (W) key and the screen will show:

```
WRITE SUB-SYSTEM
ENTER PICTURES NAME
```

Type in any name, no greater than eight characters, and press (ENTER); this will save the picture on tape.

### I-RES SCREEN PRINT UTILITIES

If your 16K Color Computer has Extended Color BASIC, you can use the Hi-Res Screen Print Utilities (26-3121) to produce a color printout of your pictures on the CGP-220. The Hi-Res Screen Print Utilities Package contains CODUMP (Color Dump Program) and BWDUMP (Black and White Program). CODUMP works especially well with Micro Painter and gives you the option of selecting a color-set for printing.

When you switch to the Hi-Res Screen Print Utilities Package you are no longer working with the color prisms used in Micro Painter. The color combinations are still the same but they are called color designation keys. They correspond as follows:

Prism 1 = (0)  
Prism 2 = (1)  
Prism 3 = (2)



If you saved your picture to tape using Prism 1 colors you do not have to use color designation key (0) to print the picture. You can select any of the color designation keys for your printout. The colors will convert just as they did in the prisms.

### Procedure for Producing a Color Printout

1. Set up the CGP-220 Ink Jet printer by following the instructions in the CGP-220 Operations Manual. Be sure to set the baud rate to 2400 by setting the Baud Rate Select Switch to the right.
2. Type CLEAR 200,14848 and press (ENTER).
3. Load the CODUMP program by typing CLOADM"CODUMP" and press (ENTER). The OK prompt returns to the screen and the tape stops after the program has been loaded.
4. Type EXEC and press (ENTER).
5. Load the Micro Painter picture to memory with the CLOADM command.
6. Dump the picture by pressing the (←) key and one of the color designation keys.

If you wish to view the picture before printing, type PMODE 3,1 then enter and run the following program:

```
10 SCREEN 1,0
20 GOTO 20
```

Press the (BREAK) key followed by the (←) key and one of the color designation keys, when you are ready to print the picture. If the picture was saved to tape in Prism 3 colors, add the following line to the above program.

```
15 POKE 65314,248
```

### MASTER MICRO PAINTER

Keep practicing, and before long you will become a Master Micro Painter. The patterns and techniques you can develop are only limited by your own imagination. After you have mastered your painting skills, be sure to try creating your own pictures. Who knows, maybe Micro Painting will be the next great art form!

# Radio Shack Computer Center Addresses

**ALABAMA**  
**BIRMINGHAM** 2426 Green Springs Hwy., 9233 Parkway East  
**HUNTSVILLE** 1400 N. Memorial Pkwy.  
**MOBILE** 405 Bel-Air Blvd.  
**MONTGOMERY** #24 Union Square S/C

**ARIZONA**  
**PHOENIX** 10233 Metro Pkwy. E., 4301 N. 7th Ave.  
**SCOTTSDALE** 2525 N. Scottsdale Rd.  
**TEMPE** 83 E. Broadway  
**TUCSON** 5622 E. Broadway, Campbell Plaza, 2830 N. Campbell Ave.

**ARKANSAS**  
**LITTLE ROCK** Town & Country S/C, University & Asher

**CALIFORNIA**  
**ANAHEIM** 509 Katella  
**BAKERSFIELD** 2018 Chester Ave.  
**BERKELEY** 1922 Grove St.  
**BEVERLY HILLS** 8500 Wilshire Blvd  
**BREA** Imperial Shopping Center, 391 South State College  
**CANOGA PARK** 8371 Topanga Canyon  
**CARMICHAEL** 6305 Fairroads Blvd.  
**CHICO** 1834 Mangrove Ave.  
**CHULA VISTA** 1201 3rd Ave.  
**CITRUS HEIGHTS** 7405 Green Back Ln. at San Juan  
**DOWNNEY** 8031 Florence Ave.  
**EL TORO** El Toro Rd. at San Diego Freeway  
**ESCONDIDO** 347 W. Mission Ave.  
**FREMONT** Fremont Hub, 39114 Fremont  
**FRESNO** Princeton S/C, 2721 N. Blackstone Ave.  
**GARDEN GROVE** 12621 Knott St.  
**GLENDALE** 236 N. Brand Blvd.  
**HAYWARD** 24784 Hesperian Blvd.  
**HOLLYWOOD** 6922 Hollywood Blvd.  
**IRVINE** Redhill at Main St.  
**LAKESIDE** 5830 Lakewood Blvd.  
**LA MESA** 5346 Jackson Dr.  
**LONG BEACH** 2119 Bellflower Blvd.  
**LOS ANGELES** 740 S. Olive St., 5240 Century Blvd. (Airport area)  
**MODESTO** 221 McHenry Ave.  
**MONTCLAIR** 5237 Arrow Hwy.  
**MONTREY** 484 Washington St.  
**MOUNTAIN VIEW** 1933 El Camino Real W.  
**OAKLAND** 1733 Broadway, (415) 763-3183  
**PASADENA** 575 S. Lake Ave.  
**PLEASANT HILL** 508 G. Contra Costa Blvd.  
**RIVERSIDE** 3844 La Sierra Ave.  
**SACRAMENTO** 4749 J. St.  
**SAN BERNARDINO** 784 Inland Center Dr.  
**SAN DIEGO** 3092 Clairemont Dr., 3902 El Cajon Blvd.  
**SAN FRANCISCO** One Market Place, 2920 Geary Blvd.  
**SAN JOSE** 1228 S. Bascom Ave.  
**SAN MATEO** 3180 Campus Dr.  
**SANTA ANA** 2320 S. Fairview St. (at Warner)  
**SANTA BARBARA** 4141 State St. A-1  
**SANTA FE SPRINGS** 14138 East Firestone Blvd.  
**SANTA MONICA** 511 Wilshire Blvd.  
**SANTA ROSA** 823 4th St.  
**SHERMAN OAKS** 14936 Ventura Blvd.  
**STOCKTON** College Sq. S/C, 963 West March Lane  
**TARZANA** 18545 Ventura Blvd.  
**TORRANCE** 3840 Sepulveda at Hawthorne  
**VENTURA** 4005 E. Main St.  
**WEST COVINA** 2516 E. Workman St.

**CONNECTICUT**  
**EAST HAVEN** 51 Frontage Rd.  
**FAIRFIELD** 1196 Kings Hwy. & Rt. 1  
**HARTFORD** The Richardson Bldg., 942 Main St.  
**MANCHESTER** 228 Spencer St.  
**NORWALK** Rt. 7-345 Main Ave.  
**ORANGE** Edwards Food Warehouse Shop. Ctr., 538 Boston Post Rd.  
**WATERBURY** 105 Bank St.  
**WATERFORD** 122 Boston Port Rd.  
**WEST HARTFORD** 39 S. Main St.

**COLORADO**  
**AURORA** Hoffman Heights Shop. Ctr.  
**BOULDER** Arapahoe Plaza, 3550 Arapahoe  
**COLORADO SPRINGS** 4341 N. Academy  
**DENVER** 8000 E. Quincy, Green Briar Plaza, 7075 Pecos  
**LAKEWOOD** 2099 Wadsworth Blvd.

**DELAWARE**  
**DOVER** Edgehill Shop. Ctr., Rt. #13  
**WILMINGTON** 3847 Kirkwood Hwy.

**DISTRICT OF COLUMBIA**  
**N.W. WASHINGTON** 1800 M St., 15th & L St.; Van Ness Station, 4250 Connecticut Ave.

**FLORIDA**  
**ALTAMONTE SPRINGS** 766 B. East Altamonte Dr.  
**BOCA RATON** 1662 N. Federal Hwy.  
**CLEARWATER** 2460-D US 19 North  
**DAYTONA BEACH** Volusia Plaza  
**FT. LAUDERDALE** 4368 N. Federal Hwy.  
**FORT MYERS** Edison Mall  
**FORT PIERCE** Center West Shopping Center, U.S. Hwy. 1  
**GAINESVILLE** 3315 Archer Rd.  
**HOLLYWOOD** 429 S. State Rd. #7  
**JACKSONVILLE** 8252 Arlington Expwy.; Roosevelt Mall, Roosevelt Blvd.  
**LAKELAND** 2810 S. Florida Ave.  
**LAUDERDALE LAKES** 4317-25 N. State Rd. 7  
**MERRITT ISLAND** 700 E. Merritt Island Causeway  
**MIAMI** 9459 S. Dixie Hwy., 16011 Biscayne Blvd., 15 SE. 2nd Ave., 20761 S. Dixie Hwy.  
**N. MIAMI BEACH** The Promenade, 1777 N.E. 163rd St.  
**ORLANDO** 1238 E. Colonial Dr.  
**PENSACOLA** Eastgate Shopping Center, 6925 N. 9th Ave.  
**ST. PETERSBURG** 3451 66th St. N.  
**SARASOTA** 5251 S. Tamiami Tr. (Hwy. 41)  
**TALLAHASSEE** 2529 S. Adams  
**TAMPA** 4555 W. Kennedy, 1825 E. Fowler Ave.  
**W. PALM BEACH** 2271-A Palm Beach Lakes Blvd.

**GEORGIA**  
**ATHENS** Beechwood Shop. Ctr.  
**ATLANTA** 2108 Henderson Mill; 49 W. Paces Ferry; Akers Mill S/C, 2937 Cobb Parkway NW.; 113 Peachtree St.  
**AUGUSTA** 3435 Wrightsboro Rd.  
**COLLEGE PARK** 3339 Old National Hwy.  
**COLUMBUS** Columbus Square Mall, 3050 Macon Rd.  
**DORAVILLE** 5697 Buford Hwy.  
**MACON** 1467 Eisenhower Pkwy.  
**SAVANNAH** Chatham Plaza, 7805 Abercorn St.

**HAWAII**  
**HONOLULU** Pacific Trade Center-Ground Floor, 190 South King St.

**IDAHO**  
**BOISE** 691 S. Capitol Blvd.

**ILLINOIS**  
**AURORA** 890 North Lake St.  
**BLOOMINGTON** 401 N. Veterans Parkway (Eastland Commons)

**CHICAGO** 4355 S. Archer Ave., CNA Plaza, 309 S. Wabash; 72 West Adams; 524 N. Michigan Ave.; 101 W. Washington St. at Clark  
**DEERFIELD** Deerbrook Mall  
**ELMWOOD PARK** 7212 W. Grand Ave.  
**FAIRVIEW HEIGHTS #4** Market Place  
**HOMEWOOD/GLENWOOD** 329 Glenwood Lansing; 18230 S. Halsted at 183rd  
**JOLIET** 2415 W. Jefferson Rd.  
**LA GRANGE** One S. LaGrange Rd.  
**LIBERTYVILLE** 1350 S. Milwaukee Ave.  
**LOMBARD** 4 Yorktown Center  
**MOLINE** 4401 44th Ave.  
**NILES** 8349 Golf Rd.  
**OAK LAWN** 4815 W. 95th St.  
**PEORIA** 4125 N. Sheridan Rd.  
**ROCKFORD** North Town S/C, 3600 N. Main St.  
**SCHAUMBURG** 651 Mall Dr.  
**SPRINGFIELD** Sherwood Plaza, 2482 Wabash

**INDIANA**  
**EVANSVILLE** 431 Diamond Ave.  
**FT. WAYNE** 747 Northcrest S/C  
**GRIFFITH** 308 W. Ridge Rd.  
**HOBART** Save-More Plaza, Rtes. #6 & 51  
**INDIANAPOLIS** 6242 E. 82nd St., Castleton Plz.; Speedway Plaza, 6129 B. Crawfordsville; 10013 E. Washington St.; Two W. Washington St.  
**SOUTH BEND** 1827 South Bend Ave.  
**TERRE HAUTE** 3460 U.S. Hwy. 41 S.

**IOWA**  
**CEDAR RAPIDS** 111 First Ave., S.E. (Downtown)  
**DAVENPORT** 616 E. Kimberly Rd.  
**DES MOINES** 7660 Hickman Rd., Sherwood Forest S/M

**KANSAS**  
**OVERLAND PARK** 8619 W. 95th  
**TOPEKA** White Lakes Plaza, West Tower, 3715 Plaza Dr.  
**WICHITA** 2732 Blvd. Plaza S/C

**KENTUCKY**  
**FLORENCE** 7727 Mall Rd.  
**LEXINGTON** 2909 Richmond Rd.  
**LOUISVILLE** Louisville Galleria, 4133 Shelbyville Rd.

**LOUISIANA**  
**ALEXANDRIA** 1213 Texas Ave.  
**BATON ROUGE** 7007 Florida Blvd.  
**GRETNA** 400 La Palco Blvd.  
**HOUMA** 2348 W. Park Ave. (Hwy. 24)  
**LA FAYETTE** University Square at Congress Blvd.  
**METairie** 3750 Veterans Hwy.  
**NEW ORLEANS** 327 St. Charles Ave.  
**SHREVEPORT** 1545 Line Ave.

**MAINE**  
**BANGOR** Maine Square

**MARYLAND**  
**BALTIMORE** 7942 Belair Rd., Putty Hill Plaza; 115 N. Charles St. at Lexington  
**BETHESDA** 7900 Wisconsin Ave.  
**CATONSVILLE** One Mile West S/C, 6600 B. Balt. Nat'l. Pike  
**FREDERICK** Shoppers World, Rt. 40W  
**NEW CARROLLTON-LANHAM** 7949 Annapolis Rd.  
**PASADENA** 8120 Ritchie Hwy.  
**ROCKVILLE** Congressional Plaza, 1673 Rockville Pike  
**SALISBURY** Shoppers World S/C, Rt. 50  
**TEMPLE HILLS** 4520 St. Barnabas Rd.  
**TOWSON-LUTHERVILLE** Yorktown S/C York Rd. at Ridgely Rd.

**MASSACHUSETTS**  
**BOSTON** 730 Commonwealth Ave.; 111 Summer St.; 372 Boylston  
**BOSTON** 372 Boylston  
**BRAINTREE** South Shore Plaza, 250 Granite St.  
**BROCKTON** 675 Belmont  
**BURLINGTON** Crossroads Plaza, Rt. 3 S  
**CAMBRIDGE** Harvard Square, 28 Boylston St.  
**CHESTNUT HILL** 200 Boylston St.  
**FALL RIVER** 90 William S. Canning Blvd.  
**NATICK** 1400 Worcester St.  
**SAUGUS** 343 Broadway Rd.  
**SPRINGFIELD** 1985 Main St., Northgate Plz.  
**WORCESTER** Lincoln Plaza

**MICHIGAN**  
**ANN ARBOR** 2515 Jackson Rd.  
**BIRMINGHAM** 3620 W. Maple Rd.  
**DEARBORN** Westborn Shop. Ctr., 23161 Michigan Ave.  
**DETROIT DOWNTOWN** 1559 Woodward Ave.  
**FLINT** 6328 Miller Rd., Yorkshire Plaza  
**GRAND RAPIDS** 3142 28th St. SE  
**KALAMAZOO** 25 Kalamazoo Center  
**LANSING** 2519 S. Cedar St.  
**LIVONIA** 33470 W. 7 Mile Rd.  
**PLAINFIELD** North Kent Mall  
**PONTIAC** North Oak Plaza-Pontiac Mall, 2436 Elizabeth Lake Rd.  
**ROSELVILLE** 31873 Gratiot Ave.  
**SOUTHFIELD** 17651 West 12 Mile Rd.  
**TROY** Oakland Plaza, 322 John R. Rd.  
**WARREN** 29038 Van Dyke Ave.

**MINNESOTA**  
**BLOOMINGTON** 10566 France Ave. S.  
**FRIDLEY** 7974-76 University Ave. North  
**GOLDEN VALLEY** Golden Valley S/C, 8016 Olson Memorial Hwy  
**MINNEAPOLIS** 830 Marquette Ave.  
**ST. PAUL** 6th & Wabasha

**MISSISSIPPI**  
**GULFPORNT** 516A Courthouse Rd.  
**JACKSON** 979 Ellis Ave.

**MISSOURI**  
**DES-PERES** 11960 Manchester Rd.  
**FLORENCE** 47 Florence Oaks S/C  
**INDEPENDENCE** 1325 S. Notland Rd.  
**KANSAS CITY** 4025 N. Oak Trafficway  
**ST. ANN** 10472 St. Charles Rock Rd.  
**ST. LOUIS** 500 No. Broadway (Commerce Bank Bldg., Downtown)  
**SPRINGFIELD** 2684 S. Glenstone

**NEBRASKA**  
**LINCOLN** 4601 "O" St.  
**OMAHA** 3006 Dodge St.; 1318 72nd St. at Pacific

**NEVADA**  
**LAS VEGAS** Commercial Center, 953 E. Sahara #31-B  
**RENO** 3328 Kietzke Lane

**NEW HAMPSHIRE**  
**MANCHESTER** Hampshire Plaza, 1000 Elm St.  
**NASHUA** 429 Amherst St., Rt. 101A

**NEW JERSEY**  
**BRIDGEWATER** 1472 U.S. Highway 22 East  
**E. BRUNSWICK** 595 A Rt. 18  
**E. HANOVER** Rt. 10, Hanover Plaza

# Radio Shack

TRS-80 Microcomputer News  
 P.O. Box 2910  
 Fort Worth, Texas 76113-2910

BULK RATE  
 U.S. POSTAGE  
 PAID  
 Radio Shack  
 A Div. of Tandy Corp.

## ADDRESS CHANGE

Remove from list

Change as shown

Please detach address label and mail to address shown above

**LAWRENCEVILLE** Rt. 1 & Texas Ave.  
**NEWARK** 595 Broad  
**NORTHFIELD** 322-24 Tilton Rd.  
**PARAMUS** 175 Rt. 17 S.  
**POMPTON PLAINS** Plains Plaza  
**SPRINGFIELD** Rt. #22 Center Isle  
**TOMS RIVER** 700 Rt. 37 West  
**VOORHEES** 35 Eagle Plaza

**NEW MEXICO**  
**ALBUQUERQUE** 2108 San Mateo NE.

**NEW YORK**  
**ALBANY** Shoppers Pk., Wolf Rd.  
**BAYSHORE** 1751 Sunrise Hwy.  
**BETHPAGE** 422 N. Wantagh Ave.  
**BROOKLYN** 531 86th St.  
**BUFFALO** 839 Niagara Falls Blvd.  
**FRESH MEADOWS** 187-12 Horace Harding Exp.  
**GARDEN CITY** 960 Franklin Ave.  
**JOHNSON CITY** Giant Shopping Center, Harry L. Drive  
**KINGSTON** Kings Mall, Rt. 9W  
**LAKE GROVE** 111-113 Alexander Ave.  
**MANHASSET** 1550 Northern Blvd.  
**MELVILLE** TSS Mall, Rt. 110  
**NEWBURGH** Zephyr Plaza, Rt. #17K  
**NEW ROCHELLE** 211 North Ave.  
**NEW YORK** 385 Fifth Ave., 139 E. 42nd St., 19 W. 23rd St., 347 Madison Ave., 270 Park Ave. South; 1282 Broadway, 9 Broadway  
**NIAGARA FALLS** Pine Plaza, 8351 Niagara Falls Blvd.  
**REGO PARK** 97-77 Queens Hwy.  
**ROCHESTER** 3000 Winton Rd.  
**SCARSDALE** 385 Central Park Ave.  
**SCHENECTADY** Woodlawn Plaza  
**SPRING VALLEY** White House Center, 88 W. Rt. 59  
**STATEN ISLAND** 2409 Richmond Ave.  
**SYRACUSE** 25-44 Erie Blvd., Hotel Syracuse, 510 S. Warren St.  
**UTICA** Riverside Mall  
**VALLEY STREAM** Green Acres Shop. Ctr.  
**YONKERS** Cross Country Shop. Ctr.

**NORTH CAROLINA**  
**ASHEVILLE** K-Mart Shopping Center, Tunnel Rd.  
**CHARLOTTE** 3732 Independence Blvd.; Tyvola Mall, 5401 South Blvd.  
**DURHAM** South Square Mall  
**FAYETTEVILLE** Eutaw Shopping Center, 815 Elm St.; Fayetteville Street Mall  
**GREENSBORO** 3718 High Point Rd.  
**RALEIGH** Townridge Sq., Hwy. 70 W.  
**WILMINGTON** Independence Mall  
**WINSTON-SALEM** 629 Peters Creek Pkwy.

**OHIO**  
**AKRON** Fairlawn Plaza, 2727 W. Market St.  
**BEDFORD HEIGHTS** 5217 Northfield Rd.  
**CANTON** 5248 Dressler Rd. NW.; Mellet Plaza, 3826 W.  
**DAYTON** Northwest Plaza, 3279 West Siebenthaler  
**ELRYIA** 286 Midway Blvd.  
**FAIRFIELD** 7255 Dixie Hwy. (1/4 Mi. North of I-275)  
**NORTH OLMDIST** Great Northern S/C  
**PARMA** 7551 W. Ridgewood Dr.  
**TOLEDO** 5844 W. Central Ave.; Brownstone Plaza, 1724 S. Reynolds Rd.  
**YOUNGSTOWN** Union Square Plaza, 2543 Belmont Ave.

**OKLAHOMA**  
**OKLAHOMA CITY** 4732 SE 29th St., Springdale S/C, 4469 NW 50th; 1101 SW 59th St.  
**TULSA** 7218 & 7220 E. 41st St.

**OREGON**  
**EUGENE** 390 Coburg Rd.  
**PORTLAND** 7483 SW Barbur Blvd.; 9131 SE Powell, 3rd and Washington Sts. (Downtown)  
**SALEM** Salem Plaza, 403 Center

**PENNSYLVANIA**  
**ALLENTOWN** Crest Plaza S/C, Cedar Crest Blvd. US 22  
**BALA CYNWYD** 67 E. City Line Ave.  
**EASTON** 25th St. Shopping Center  
**ELKINS PARK** Elkins Park Square, 8080 Old York Rd.  
**ERIE** 5755 Peach St.  
**HARRISBURG** Union Deposit Mall, Union Deposit Rd. #17  
**LANCASTER** Park City Plaza, US 30  
**MONROEVILLE** 3828 Wm. Penn. Hwy.  
**MONTGOMERYVILLE** Airport Sq., Rt. 309  
**PHILADELPHIA** 7542 Castor Ave.; 1002 Chestnut St.; 1801 Market St., 10 Penn Center

**PITTSBURGH** 5775 Baptist Rd., Hills Plaza, 303 Smithfield St.; 4643 Baum Blvd.; 4768 McKnight Rd.  
**SCRANTON** 206 Meadow Ave.  
**WILKES BARRE**, 23 W. Market St.  
**WYOMISSING** Berkshire Mall West, 1101 Woodland Rd.  
**YORK** York County Shopping Center

**PUERTO RICO**  
**HATO REY** 243 Franklin D. Roosevelt Ave.

**RHODE ISLAND**  
**E. PROVIDENCE** 850 Waterman Ave.  
**PROVIDENCE** 177 Union St.

**SOUTH CAROLINA**  
**COLUMBIA** Old Sears Bldg., 1001 Harden St.  
**GREENVILLE** N. Hills S/C  
**N. CHARLESTON** 5900 Rivers Ave.  
**SPARTANBURG** Hillcrest Shop. Ctr.

**SOUTH DAKOTA**  
**SIoux FALLS** 1700 S. Minnesota at 25th

**TENNESSEE**  
**CHATTANOOGA** 636 Northgate Mall  
**JOHNSON CITY** Peerless Center  
**KNOXVILLE** Cedar Blk. and #123 Executive Park Dr.  
**MEMPHIS** 4665 American Way, 1997 Union St.  
**NASHVILLE** 2115 Franklin Pike; Rivergate Plaza

**TEXAS**  
**AMARILLO** Wellington Sq. S/C, #619 S. Kentucky  
**ARLINGTON** 2500 E. Randall Mill, Suite 113  
**AUSTIN** 8764 E. Research Blvd.; Southwood Mall, 1501 Ben White Blvd.  
**BROWNSVILLE** 1639 Price Rd. (Hwy. 77)  
**BEAUMONT** 5330 Eastex Fwy.  
**COLLEGE STATION** 2414 Texas Ave., South  
**CORPUS CHRISTI** 1711 S. Staple St.  
**DALLAS** 15340 Dallas Pkwy., Suite 1100; 2930 W. Northwest Hwy.; 1517 Main St.; 2588 Royal Ln.  
**EL PASO** 9515 Gateway West, Kern Plaza Shopping Center, 3100 N. Mesa  
**FT. WORTH** 231 One Tandy Center, 2801 Alta Mere  
**GALVESTON** 5924 Broadway  
**HARLINGEN** 1514 S. Hwy 77, Sunshine Strip  
**HOUSTON** 211C-FM 1960, 10543 Gulf Fwy.; 5900 North Fwy.; 6813 SW Fwy.; 809 Dallas St.; Holland Square Center, 10920 East Freeway, (713) 453-0600; Champion Forest Plaza, Champion Forest Dr. and J.M. 1960 West, 1018 Gessner; 3278 South Loop West (So. Main at 610)  
**HUMBLE** 19300 "B" Hwy. 59 (at FM 1960)  
**HURST** Northeast Mall  
**IRVING** 2011 West Airport Fwy.  
**LAREDO** 102 East Calton Rd.  
**LUBBOCK** 3625 34th St.  
**MESQUITE** Town East Crossing Shop. Ctr.  
**ODESSA** 1613 "A" East 8th Street  
**RICHARDSON** Fleetwood Sq. S/C, 202 W. Campbell Rd.; 320 S. Central Exp.  
**SAN ANTONIO** 6018 West Ave.; 4249 Centergate; Riverbend Parking Garage, 211 W. Market St. (Downtown)  
**WICHITA FALLS** 1720-A 9th St.

**NORTH CAROLINA**  
**ASHEVILLE** K-Mart Shopping Center, Tunnel Rd.  
**CHARLOTTE** 3732 Independence Blvd.; Tyvola Mall, 5401 South Blvd.  
**DURHAM** South Square Mall  
**FAYETTEVILLE** Eutaw Shopping Center, 815 Elm St.; Fayetteville Street Mall  
**GREENSBORO** 3718 High Point Rd.  
**RALEIGH** Townridge Sq., Hwy. 70 W.  
**WILMINGTON** Independence Mall  
**WINSTON-SALEM** 629 Peters Creek Pkwy.

**OHIO**  
**AKRON** Fairlawn Plaza, 2727 W. Market St.  
**BEDFORD HEIGHTS** 5217 Northfield Rd.  
**CANTON** 5248 Dressler Rd. NW.; Mellet Plaza, 3826 W.  
**DAYTON** Northwest Plaza, 3279 West Siebenthaler  
**ELRYIA** 286 Midway Blvd.  
**FAIRFIELD** 7255 Dixie Hwy. (1/4 Mi. North of I-275)  
**NORTH OLMDIST** Great Northern S/C  
**PARMA** 7551 W. Ridgewood Dr.  
**TOLEDO** 5844 W. Central Ave.; Brownstone Plaza, 1724 S. Reynolds Rd.  
**YOUNGSTOWN** Union Square Plaza, 2543 Belmont Ave.

**OKLAHOMA**  
**OKLAHOMA CITY** 4732 SE 29th St., Springdale S/C, 4469 NW 50th; 1101 SW 59th St.  
**TULSA** 7218 & 7220 E. 41st St.

**OREGON**  
**EUGENE** 390 Coburg Rd.  
**PORTLAND** 7483 SW Barbur Blvd.; 9131 SE Powell, 3rd and Washington Sts. (Downtown)  
**SALEM** Salem Plaza, 403 Center

**PENNSYLVANIA**  
**ALLENTOWN** Crest Plaza S/C, Cedar Crest Blvd. US 22  
**BALA CYNWYD** 67 E. City Line Ave.  
**EASTON** 25th St. Shopping Center  
**ELKINS PARK** Elkins Park Square, 8080 Old York Rd.  
**ERIE** 5755 Peach St.  
**HARRISBURG** Union Deposit Mall, Union Deposit Rd. #17  
**LANCASTER** Park City Plaza, US 30  
**MONROEVILLE** 3828 Wm. Penn. Hwy.  
**MONTGOMERYVILLE** Airport Sq., Rt. 309  
**PHILADELPHIA** 7542 Castor Ave.; 1002 Chestnut St.; 1801 Market St., 10 Penn Center

**PUERTO RICO**  
**HATO REY** 243 Franklin D. Roosevelt Ave.

**RHODE ISLAND**  
**E. PROVIDENCE** 850 Waterman Ave.  
**PROVIDENCE** 177 Union St.

**SOUTH CAROLINA**  
**COLUMBIA** Old Sears Bldg., 1001 Harden St.  
**GREENVILLE** N. Hills S/C  
**N. CHARLESTON** 5900 Rivers Ave.  
**SPARTANBURG** Hillcrest Shop. Ctr.

**SOUTH DAKOTA**  
**SIoux FALLS** 1700 S. Minnesota at 25th

**TENNESSEE**  
**CHATTANOOGA** 636 Northgate Mall  
**JOHNSON CITY** Peerless Center  
**KNOXVILLE** Cedar Blk. and #123 Executive Park Dr.  
**MEMPHIS** 4665 American Way, 1997 Union St.  
**NASHVILLE** 2115 Franklin Pike; Rivergate Plaza

**TEXAS**  
**AMARILLO** Wellington Sq. S/C, #619 S. Kentucky  
**ARLINGTON** 2500 E. Randall Mill, Suite 113  
**AUSTIN** 8764 E. Research Blvd.; Southwood Mall, 1501 Ben White Blvd.  
**BROWNSVILLE** 1639 Price Rd. (Hwy. 77)  
**BEAUMONT** 5330 Eastex Fwy.  
**COLLEGE STATION** 2414 Texas Ave., South  
**CORPUS CHRISTI** 1711 S. Staple St.  
**DALLAS** 15340 Dallas Pkwy., Suite 1100; 2930 W. Northwest Hwy.; 1517 Main St.; 2588 Royal Ln.  
**EL PASO** 9515 Gateway West, Kern Plaza Shopping Center, 3100 N. Mesa  
**FT. WORTH** 231 One Tandy Center, 2801 Alta Mere  
**GALVESTON** 5924 Broadway  
**HARLINGEN** 1514 S. Hwy 77, Sunshine Strip  
**HOUSTON** 211C-FM 1960, 10543 Gulf Fwy.; 5900 North Fwy.; 6813 SW Fwy.; 809 Dallas St.; Holland Square Center, 10920 East Freeway, (713) 453-0600; Champion Forest Plaza, Champion Forest Dr. and J.M. 1960 West, 1018 Gessner; 3278 South Loop West (So. Main at 610)  
**HUMBLE** 19300 "B" Hwy. 59 (at FM 1960)  
**HURST** Northeast Mall  
**IRVING** 2011 West Airport Fwy.  
**LAREDO** 102 East Calton Rd.  
**LUBBOCK** 3625 34th St.  
**MESQUITE** Town East Crossing Shop. Ctr.  
**ODESSA** 1613 "A" East 8th Street  
**RICHARDSON** Fleetwood Sq. S/C, 202 W. Campbell Rd.; 320 S. Central Exp.  
**SAN ANTONIO** 6018 West Ave.; 4249 Centergate; Riverbend Parking Garage, 211 W. Market St. (Downtown)  
**WICHITA FALLS** 1720-A 9th St.