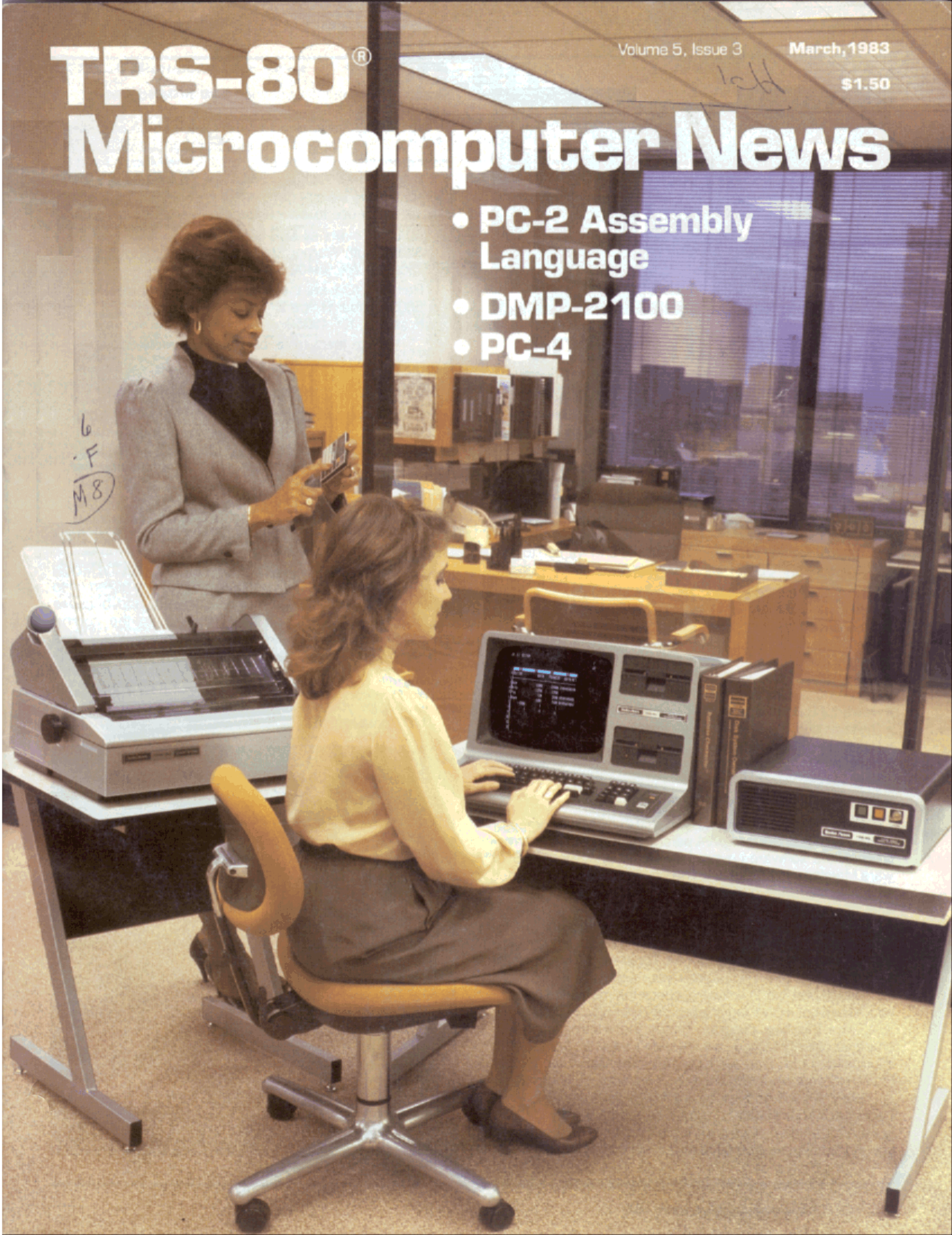
Volume 5, Issue 3 March, 1983
$1.50

TRS-80® Microcomputer News

- PC-2 Assembly Language
- DMP-2100
- PC-4



Fort Worth Scene

CP/M FOR MODELS II/12/16

As you probably know, the material for each issue of TRS-80 Microcomputer News is written two or three months before you get the magazine at home. I am writing this in early January. One of the products which Radio Shack introduced in January, and which we were not able to cover in February was CP/M Plus © Digital Research, introduced for the TRS-80 Models II/12 and 16 (operating in Model II mode). CP/M Plus is version 3.0 of the CP/M operating system. We are planning an article on CP/M Plus for the April issue and wanted to give you a few of the CP/M Plus highlights here:

- High Performance File System
- CP/M 2.2 Compatible
- Time and Date stamps on files
- Automatic Disk Login of Removable Media
- Support for Radio Shack's 8 and 12 Meg Hard Drives
- Console I/O Redirection
- Easy to use System Utilities with HELP Facility
- Powerful Batch Facility
- Systems primitives designed for applications programmers
- Resident system extensions
- Sophisticated Programmer Utilities.

BUGS, ERRORS AND FIXES

Beginning with the April issue of TRS-80 Microcomputer News, we will no longer print patches and program corrections for the software covered by Radio Shack's new Software Maintenance Program. The covered software includes all disk based, non-game software for Models I, II, III, 12, and 16.

Under the Software Maintenance Program, you will receive a letter concerning corrections or fixes for any covered software which you own and have registered. This makes it very important that you register your software. If you have disk based non-game software, check the manual to be sure you have sent in the registration card for EACH package.

Since this new program brings the information about changes directly to you, we shouldn't need to publish that information here. (You will have the information two to four MONTHS sooner under the new program!)

What we will do is publish a list of software which has been modified under the program so you can verify that you got the information. If we publish notification of a change that you did not receive, you need to contact your local Computer Center or Computer Customer Service for details on the changes.

What we will also do is to continue to publish corrections for Color Computer and Pocket Computer software, as well as for tape based or game programs for the Models I and III.

Below are two 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.

80-US (Covers all TRS-80's.)*
3838 South Warner Street
Tacoma, WA 98409

Rainbow (Covers the TRS-80 Color Computer.)
5803 Timber Ridge Dr.
Prospect, KY 40059

*Write them to receive a free sample copy.

View 7th *(From Page 4)*

products, and I have never seen a CP/M program that did not require an 80 character by 24 line screen. Most of the programs also require 52K or more memory. So while the $595 64K machine above might get its CP/M processor board someday, it still won't run any currently available software. A big name in the video game business announced a 4K, $200, CP/M based computer that is also restricted to a 40 character screen. The sad thing is that people are going to buy these products based on these crazy claims.

In case you missed it, we are offering CP/M for the Model II and 12. We are doing it because many of you asked us to support it, so you could run other software, and because Digital Research kindly said they would do the coding needed to get the new version of CP/M to run on the II. This new version is quite a product; it has many enhancements that overcome most of our earlier hesitations about CP/M, and it can address 128K if an additional 64K board is installed. I hope you Model II owners like our decision.

Until next month.

TRS-80 Microcomputer News
Volume 5 Issue 3
MARCH 1983

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 1983 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 the 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 William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars ($12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1.50 suggested retail each.

The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price $9.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007,535.

Notes to Program Users:

Programs published in the Microcomputer 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 can not assume any liability for the accuracy either of the programs themselves or of the results provided by the programs.

Further, while 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 can not be supported by our 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 and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

CompuServe™	CompuServe, Inc.
Dow Jones™ NEWS/RETRIEVAL Service®	Dow Jones & Co., Inc.
VisiCalc®	VisiCorp, Inc.
Program Pak™	Tandy Corporation
SCRIPSIT™	Tandy Corporation
TRSDOS™	Tandy Corporation
TRS-80®	Tandy Corporation

TRS-80® Microcomputer News

Contents:

Bugs, Errors, and Fixes
Color Computer
Color SCRIPSIT (26-3105) . . . 39
Monster Maze (26-3081) . . . 39
Polaris (26-3065) . . . 39
Model I/III
5 Megabyte Hard Disk (26-1130) . . . 40
RUNCOBOL (26-2203) . . . 40
Model II
Profile Plus (26-4515 Version 1.0) . . . 41
RSCOBOL Development System (26-4703) . . . 41
Peripherals
Digitizer (26-1195) . . . 41
Pocket Computer
Engineering Math II (26-3526) . . . 40
Plotter Cassette Interface Owner's Manual (26-3605) . . . 40
Madness and the Minotaur (26-3313) . . . 39
Music Pak (26-3151) . . . 39
Model III Hard Disk Operating System (26-1130) . . . 40
Model III Programming Tip . . . 39
SCRIPSIT (26-4531 Version 2.0) . . . 41
TRSDOS-HD (26-4150 Version 4.1) . . . 41
Engineering Math IV (26-3528 Version 1.0) . . . 40

Color Computer
Sands of Egypt by Bruce Elliott . . . 45
Programs
Additional PMODE3 Color Set by Joseph Kohn . . . 31
COCOTEST by David J. Stretch . . . 16
High-Res Screen Dump for CGP-115 by David Andrew Palmer . . . 45
SAVEDATA for the Color Computer by George Fraser . . . 7

Computer Clubs . . . 38

Computer Customer Service . . . 10
Installing TRS-XENIX on Your TRS-80 Model 16

Data Bases
CompuServe . . . 30
What is "Videotex"?
Profile . . . 32
Profile III+ and Your Printer: A Marriage Made in Heaven

Education
Discovering Pluto with Dr. Tombaugh and Talk/Tutor . . . 12
Exploring Computer Careers with Radio Shack's "Careers in Computing" . . . 11
Network 3 Menu-Maker Utility by Jack Tjarks . . . 13

Fort Worth Scene . . . 2

General Interest
The Software Maintenance Program by Annette Zamberlin-Main . . . 14

Model I/III
Programs
Renumbering on the Models I and III by Matthew Belmonte . . . 7
Resistor Color Code Interpreter by Roy Cartier . . . 9
Tape Debug Hint by Scott Goehring . . . 22
USR Functions from FORTRAN by John Turner . . . 43

Model II/12
Programs
Transforming FORTRAN Files from IBM Format to TRS Model II Format by George Greenwald . . . 29

Notes on Previous Newsletters . . . 37
July 1980
Machine Language sort for Model I and Model II
July/August 1982
Asteroids
September 1982
Model II Bugs, Errors, and Fixes, Accounts Payable (26-4505)
October 1982
Mailing Labels on the CoCo
November 1982
Navigation on the CC
December 1982
Christmas Card
Clock/Scheduler for the Busy Executive
Concentric Circles on the CoCo
Model II Twinkling Christmas Tree
Notes Previous

Pocket Computer
PC-2/PC-1 Program Adaptation and Incompatibilities by Peter Levy . . . 27
The PC-4 Has Arrived by Linda Miller . . . 17
Pocket Computer 2 Assembly Language by Bruce Elliott . . . 23

Peripherals
The DMP-2100 by Annette Zamberlin-Main . . . 5
The Digitizer and the Model II by Alfred Pacheco . . . 42

View from the 7th Floor by Jon Shirley . . . 4

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

View From the Seventh Floor

by Jon Shirley
Vice President
Radio Shack Computer Merchandising

It is early January, and I just returned from the Consumer Electronics Show in Las Vegas. This is a humongous show, over 70,000 flood the city to see the show, eat bad food, get jammed into the casino shows and lose money. Is it worth it? You bet. The CES show is an overview of the entire consumer electronics business. You can tell what's hot and what's not by simply looking at where the booths are, how big they are and most important how crowded they are. It was obvious from this trip that low end computers are very, very hot, as are all telephone products. What is not hot is audio, radio, video, and almost all other consumer electronics. What all this means is that you will see more TV ads next fall for computers under $300, than for aspirin.

Is this good or bad? I think it's great, not just because we will sell lots of computers, but because of the potential of this market there will soon be more software, data base services, peripherals, etc. to support the growing mass of home computers. In reading the literature, and the signs, it became clear that there are some games being played in this business and two of those games, the memory game and the CP/M game, are this month's subjects.

The Memory Game Everyone is trying to claim they have more memory than the next guy, and all of the claims get very confusing. Let's see if we can sort them out. First, all vendors, that I know of, quote memory in their specifications as the amount of total RAM in the system. (I hope you all know the difference between RAM and ROM. It is in the manual, but quickly, ROM is memory that you can't alter that is not lost when the computer is turned off and normally contains the BASIC and system software; and RAM is the memory you and the computer use whose contents are lost when the power is off.)

Thus, if a computer has 16K of RAM it is called a 16K computer. If it also has 16K of ROM almost all manufacturers would still call it a 16K computer. Does this mean that you have all 16K of RAM available for your program and data? No, it does not. Virtually every computer must use some of the available RAM, which reduces the user memory, leaving the "real" available memory in the computer. Why then do we not all quote "user" memory? Because it is like hitting a moving target . . . read on.

Let's start by looking at the two basic types of computers, those with ROM and those without. That stuff in ROM that runs the system and includes the BASIC still has to be there even if there is no ROM. Where does it go? Right, into RAM. Systems without ROM are always disk systems. In fact they do have a little ROM which sets up the computer on power up and goes looking for disk. That is called the "boot" ROM, and when it is done with its task it is switched out of the system. Consider the Model II. The boot ROM loads in the disk operating system from drive 0 into the 64K of RAM. The ROM is then switched out and the user has 55,295 bytes for user programs.

If the user is running a machine language program like SCRIPSIT that is the available memory. But if he runs BASIC, that must also be loaded into memory and user memory available for BASIC is 55,295 less the 17,920 that BASIC takes, less the buffer space for each file he wants to open. To further confuse the issue, some TRSDOS commands like DO files or SETCOM use even more memory.

Please stay with me, it gets easier. One more element I did not address above is Video memory. The screen also needs memory in all computers, and there are two ways of providing screen memory. One is to take it from user memory and the other is to provide separate memory. The Model II has another set of RAM's for the Video, and through a technique called bank switching, they occupy the same addresses within the 64K as the user memory but without taking up user memory. That's a neat trick which I will not even begin to try to explain.

Now let's look at the ROM based systems like the Color Computer, the Model III, and virtually all the low cost computers. Since all that system stuff and BASIC is in ROM, the user memory is whatever the RAM memory is, right? Wrong. First of all, computers must steal a little RAM memory for the microprocessor to read and write to while it is doing its thing. And BASIC also needs some RAM memory. So when you turn on the power, you lose a little memory. If the system also uses the main RAM for video memory, you lose even more.

In the Model III, for example, the system steals about 1000 bytes, so a 16K machine has 15384 bytes of user memory. The video memory is another 2K in its own location so it does not affect the user memory. By comparison, the Color Computer must use the user memory for video, so a 16K Color Computer on power up has only 14,631 bytes of user memory. (In case you did not read your manual, 1K is 1024 bytes. The binary logic in digital computers results in most everything being in multiples of 8's or 16's.) If you have an extended Color BASIC computer, you can set different screen resolutions with high resolution taking more user memory.

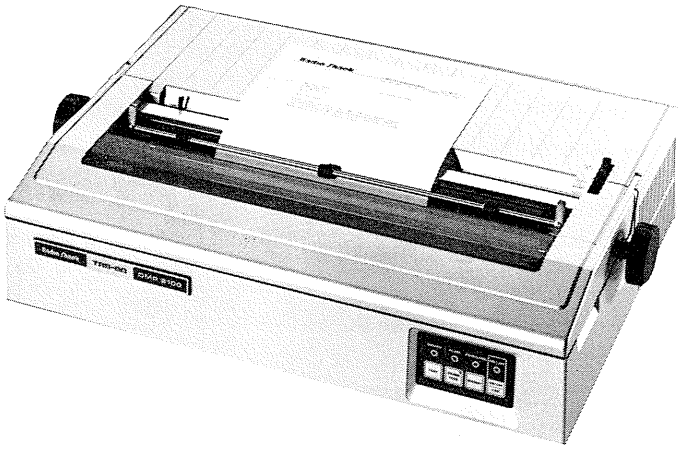
What all this means is that you must read the fine print or the manual to find out the "real" memory of any given computer. There is a widely advertised $595 64K computer on the market, BUT in BASIC the user memory is only 39K. In a 32K Color Computer the user memory is over 29K. This problem will not go away as the memory game is played by more and more people.

The CP/M Game The other popular game I saw at the show was to declare all kinds of machines able to run CP/M. Most of these machines were 32 or 40 character screen

(Continued on Page 2)

The DMP-2100

by Annette Zamberlin-Main



I like this printer. I mean I really like this printer! It has been in my office now for about two weeks, on loan from my editor. And I am still formulating a list of possible excuses, in order to somehow keep from returning it to him after I have finished writing this article. Any ideas you might have on how I could quietly pull this off, would be greatly appreciated.

Radio Shack's newest addition to the DMP-series is the TRS-80® DMP-2100 Triple-Mode Printer (26-1256). It typifies the advanced state of development that has reached the dot matrix line which includes a family of printers such as the DMP-100, DMP-200, DMP-400, and DMP-500.

I hardly know where to begin with all of this. Let's clear some of the particulars away first, and then get into the really good "stuff."

The DMP-2100 is available for $1995.00 at Radio Shack Computer Centers and participating Radio Shack stores and dealers. It rivals the Daisy Wheel printers in offering letter quality printing capabilities. It also features high-speed printing for data processing applications and high-resolution for dot-addressable graphics. The DMP-2100 character set includes 96 ASCII and 31 international characters.

At the heart of the DMP-2100 is a very versatile 24-wire print head which, in the data processing mode, is capable of printing 160 characters per second, 60 lines per minute at 132 columns. This is great for me, when I want to print out something on hard copy, just to take a look at.

```
"#$%&'()*+,-./0123456789:;<=>?@ABCDE
FGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi
{|}~´àçÉ`µ°¨†§®©¼¾½¶¥ÄÖÜ¢~äöüß™éùè¨ƒ
```

Data Processing print sample

Occasionally I want to print out a correspondence-quality document to hand into my editor. The printer has the capability of accomplishing this by printing at 100 characters per second (774 words per minute).

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
tuvwxyz{|}~´àçÉ`µ°¨†§®©¼¾½¶¥ÄÖÜ¢~äöüß™éùè¨ƒ
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
tuvwxyz{|}~´àçÉ`µ°¨†§®©¼¾½¶¥ÄÖÜç~äöüß™éùè¨ƒ
```

Correspondence print sample

If you haven't guessed it, the above is my personal favorite with this printer. You can just barely discern that this is dot matrix print. Yes, dot matrix printing has come of age.

```
Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll
Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx
Yy Zz 0 1 2 3 4 5 6 7 8 9
```

Daisy Wheel print sample

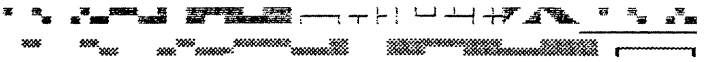
```
Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll
Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx
Yy Zz 0 1 2 3 4 5 6 7 8 9
```

Correspondence print sample

GRAPHICS MODE

As mentioned before the DMP-2100 prints on two graphic levels.

The regular, low-resolution mode is the standard Graphics Mode that other Radio Shack printers, such as the DMP-200 (26-1254), are capable of printing. Each dot that the DMP-2100 produces is actually a configuration of 9 dots created in a 3 x 3 matrix.

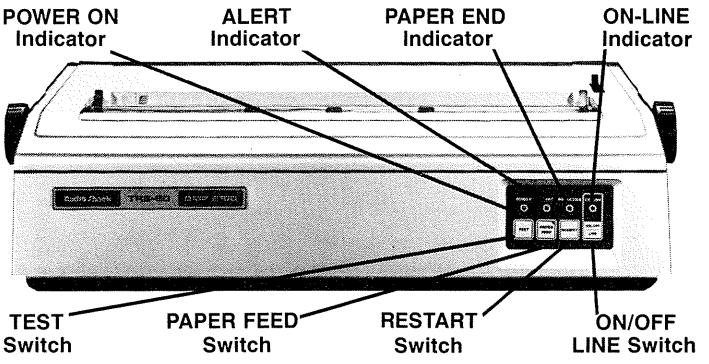


Low-Resolution graphics

The high-resolution mode of the DMP-2100 takes advantage of the printer's 24-wire print head.

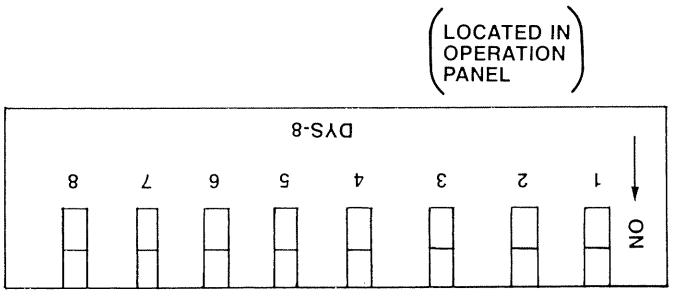
High-Resolution graphics

OUTSIDE AND INSIDE THE OPERATION PANEL



The front, right corner of the printer is a very critical area, externally and internally. For the most part, here is where you direct what is going to take place.

Located on the interior wall of the operation panel are eight switches. These switches allow you the capability to customize some of the DMP-2100 features to your own applications.



For instance, by setting the appropriate switches before turning the DMP-2100's power ON, I was able to select my favorite Correspondence-10 print style. This setting then becomes the default setting, until I select another configuration of switches.

A handy printed chart has been glued onto the interior floor (front right corner) of the printer, for quick reference of the different options available. This feature saves an incredible amount of time because you are not always trying to dig up your printer's manual for reference. It's all there, right in front of you.

FUNCTION SELECTION SWITCH

	OFF			ON
8				
7	ASCII CHARACTER			TANDY CHARACTER
6	CR = NL			CR = CR
5	LF = LF			LF = NL
4	DP MODE			WP MODE
	3	2	1	CHARACTER
	OFF	OFF	OFF	STANDARD-10
	OFF	OFF	ON	STANDARD-12
	OFF	ON	OFF	CONDENSED-16.7
	OFF	ON	ON	STANDARD-10
	ON	OFF	OFF	CORRESPONDENCE-10
	ON	OFF	ON	CORRESPONDENCE-12
	ON	ON	OFF	STANDARD-10
	ON	ON	ON	PROPORTIONAL

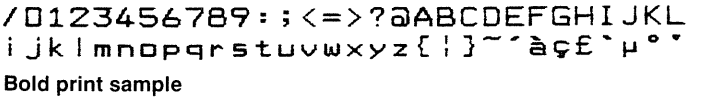
BOLD AND UNDERLINE

Two of my other favorites with the DMP-2100 are its Bold and Underline capabilities. Each of these can be accessed through BASIC program LPRINT commands in decimal values.

For example, in BASIC a Start Bold would read:
LPRINT CHR$(27);CHR$(31)
An End Bold would read:
LPRINT CHR$(27);CHR$(32)
In BASIC, a Start Underline would read:
LPRINT CHR$(15)
A Stop Underline would read:
LPRINT CHR$(14)

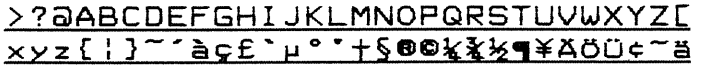
And in SCRIPSIT for the Model II, the Printer Control Codes, located in the Disk Utilities, are accessed via hexadecimal values. This technique will save a great deal of time. You are no longer waiting for the SCRIPSIT software to painstakingly direct the printer to type and retype over each character (one-by-one) in a given segment of text which you desire to be in boldface; or type (one-by-one) under each character you desired to be underlined. You would go directly into the Printer Control Codes, so that the printer can control the boldface or underline procedures within an entire segment of print.

A Start Bold would read:
2 0 1B1F
An End Bold would read:
2 0 1B20



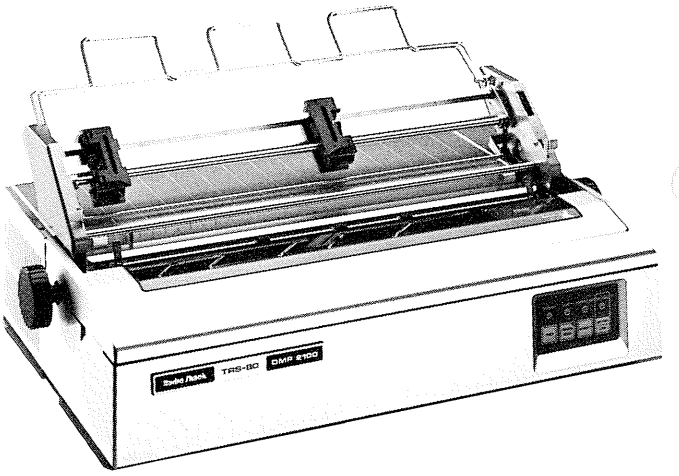
Bold print sample

A Start Underline would read:
1 0 0F
A Stop Underline would read:
1 0 0E

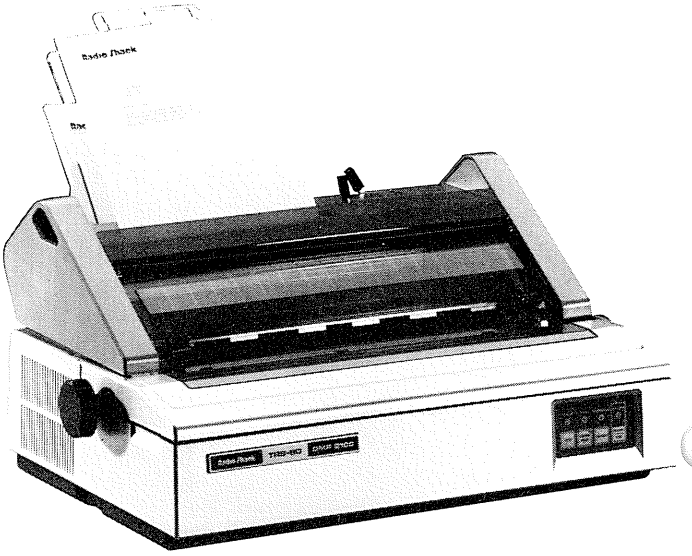


Underline print sample

PAPER HANDLING AND OPTIONAL EQUIPMENT



Radio Shack also offers optional equipment for the DMP-2100. A Bi-Directional Tractor (26-1441) for use with



fanfold paper, labels, and multi-part forms, is available separately for $169.95.

You can also manually feed a single sheet of paper into the printer. For larger single sheet jobs a Sheet Feeder (26-1440) for automatic paper insertion and stacking is recommended. The Sheet Feeder is available separately for $995.00 at Radio Shack Computer Centers and participating Radio Shack stores and dealers.

CONCLUSION

Well there it is. Some of the reasons why I think the DMP-2100 is so special. I am sure when you see it demonstrated for the first time, you will think it's equally as great. Now back to my current problem. How do I keep this new printer in my office . . .

Renumbering on the Models I and III

Matthew Belmonte
2120 Marlboro Drive
Alexandria, VA 22304

Add these lines to a program to be renumbered and type RUN63996. NEWLINE is the new line number of the first line to be renumbered. STARTLINE is the line in the program where renumbering is to begin. INCREMENT is the increment between each renumbered line.

To begin with, the program gets the start address from the beginning of BASIC program pointer at addresses 16548 and 16549. It then POKES the new line number in LSB, MSB sequence into the third and fourth bytes of the line. After renumbering of the line is complete, the program gets the start address of the next line from the first and second bytes of the line.

After renumbering is complete, check to see the new line numbers of the renumbering program and delete them.

NOTE: This program only renumbers lines, and does not update GOTO, GOSUB and other line references.

```
63996 W=Ø
      : INPUT "NEWLINE"; Y
      : INPUT "STARTLINE"; S
      : INPUT "INCREMENT"; I
      : X = PEEK(16548) + 256 * PEEK(16549)
      : A = Y / 255
      : IF A >= 1 THEN W = INT(A)
63997 Y = Y - W
      : FOR Z = W TO 255
      : FOR Y = Y TO 255 STEP I
      : IF Y > 255 THEN Y = Y - 255
      : NEXT Z
63998 IF PEEK(X+2) + 256 * PEEK(X+3) < S THEN NEXT Y
      : NEXT Z
63999 IF PEEK(X) > Ø OR PEEK(X+1) > Ø THEN POKE X +
      2, Y
      : POKE X + 3, Z
      : X = PEEK(X) + 256 * PEEK(X+1)
      : NEXT Y
      : NEXT Z
```

SAVEDATA for the Color Computer

George Fraser
3403 Bear Creek Drive
Newbury Park, CA 91320

It has happened to me too often. I am running a program on my Color Computer which requires a lot of string data to be tabled, when suddenly my program blows up with an "FC" or "OS" error. Or I realize I am going to have to break into the program to change a line with Edit. Or I break into the program to do a "SKIPF" to position the cassette for output, and I get an I/O error. In all of these cases, BASIC will not let me continue the program with "CONT." The best I can do is run it again from the top, and all my string data is lost!

To avoid this disaster, I wrote this machine language user subroutine. It will store my string data in my own well protected area of memory. With it, I can edit the BASIC calling program as much as I want, then run it from the top, and all of my string data is still intact.

Interested in seeing how it works? If so, play around with the BASIC program. Key it in carefully and make sure you have a good copy CSAVEd on tape before you try to run it for the first time. A mistake in a machine language routine can easily blow your whole system and destroy your BASIC program in memory.

This demo program may look quite complex, but it is mostly comments. The USR routine substitutes for the "DIM A$(N)" function of Color BASIC. The size (N) of the USR table (array) is never defined and is limited only by the amount of memory you reserve for it in lines 80-90. S is the starting address, and E is the address of the last byte.

You can make the table as long or short as you wish by changing E. In the demo program, I am using the first two graphics pages for the table. This works fine if you don't do any graphics commands or a "PCLS." You may relocate it to any protected area of memory by changing S and E. The first 178 bytes of the table area you define with S and E are occupied by the machine language program code. Immediately following that is the data area for string storage.

The program will store and retrieve strings of any length (including 0), limited by the size of L$ in line 70. A string may also be updated to a longer or shorter length after it has been initially stored (Lines 490-580).

Another advantage of the USR routine is that if you are using a really large array, say 100 or more entries, you will be able to pack more string data into a given amount of memory than if you let BASIC store the strings for you. BASIC sets up a five byte string descriptor for each string in its array. The machine language program uses only one byte of "overhead" per string (plus the 178 bytes of machine language code), which is a length code of Hex '00' to 'FF' at the beginning of each string stored.

Here is how to include the USR routine in a "real" BASIC program of your choice. After you are sure the demo program is working correctly, save the machine language code on tape with the command:

```
CSAVEM "NAME", 1536, 1713, 35Ø
```

Include up front in your program lines 40, 70-140, and 160-200. Do not include 150 and 810 - 1070. Instead, you will do a "CLOADM" with the machine language tape you created. Lines 100 - 140 define the five entry points to the machine language code. Line 160 will pass the end address to the machine language program, which stores it. Lines 170 - 200 are crucial, for this allows you to run the program from the top without wiping out your previous data, as when you stopped to edit your BASIC program.

When you clear the table (line 200), the program moves hex '00' to the whole data area and sets its current position pointer to the top of the data area (table entry zero).

Lines 360-410 show how to load the table sequentially (from top to bottom, in order). This could be from the keyboard or from cassette or disk input data files. A$ is loaded at the current position pointer, which is then automatically advanced to the next entry in the table. USR0 will return X = – 1 to BASIC when your table is full (line 760).

A call to USR3 (line 640) will move the current position pointer to whatever table entry you wish. The top entry is zero. This will be followed by a call to USR0 (load an entry) or USR1 (retrieve an entry).

Lines 360 - 420 illustrate sequential retrieval. This could be used to print part or all of your table to the printer, disk, or cassette output data file. USR1 moves the table entry of the current position pointer to BASIC's L$ and then advances the pointer to the next entry. It does not shorten L$, but does return X = the length of the string in the table. This is why it is necessary to use the LEFT$ function in line 700. It will return X = – 1 if L$ is not long enough, and move nothing to L$.

I have been using routines similar to this in home-brew word processing and 6809 assembler programs with excellent results. Such programs require a large array of string data to be stored, usually thousands of bytes. My early versions of these programs used BASIC string arrays, but once I converted to the machine language USR routines, I was sure glad I did.

```
10 ' "SAVEDATA" FOR THE TRS-80 COLOR COMPUTER.
     G.FRASER.  FEB 1982.
20 ' USR ROUTINE TO STORE AND RETRIEVE STRING DATA.
30 '
40 CLS
     : PCLEAR 4
     : CLEAR 350
50 PRINT "USR DATA STORAGE ROUTINE"
60 PRINT "USING MEMORY 1536-4607 FOR TABLE"
70 L$=STRING$(100," ")  'LENGTHEN OR SHORTEN THIS
     STRING AS NEEDED.
80 S=1536  'START OF TABLE AREA
90 E=4607  'END OF TABLE
100 DEFUSR0 = S       'LOAD STRING
110 DEFUSR1 = S + 102  'GET STRING
120 DEFUSR2 = S + 134  'CLEAR TABLE
130 DEFUSR3 = S + 148  'REPOSITION
140 DEFUSR4 = S + 171  'END VALUE
150 GOSUB 830       'LOAD ML CODE
160 X = USR4(E)       'SET END VALUE
170 INPUT "CONTINUE OR NEW <C,N>"; A$
180 IF A$ = "C" THEN GOSUB 790
     : GOTO 310
190 INPUT "PRESS ENTER TO CLEAR TABLE"; A$
200 X = USR2(0)     'CLEAR TABLE
210 C = -1          'COUNTER
220 '
230 PRINT "LOAD THE TABLE SEQUENTIALLY"
240 PRINT "ENTER SLASH </> WHEN ALL DATA IS LOADED."
250 LINEINPUT A$
260 IF A$ = "/" THEN 310
270 GOSUB 750      'LOAD A$
280 C = C + 1          'COUNTER
290 GOTO 250
300 '
310 PRINT
320 INPUT "SEQUENTIAL RETRIEVE, RANDOM RETRIEVE, OR
     RANDOM UPDATE <S,R,U>"; A$
330 IF A$ = "R" THEN 440
340 IF A$ = "U" THEN 490
350 '
360 PRINT "SEQUENTIAL RETRIEVAL"
370 GOSUB 610  'SET TABLE POSITION FOR START
380 FOR D = A TO C
390 PRINT D;
400 GOSUB 680  'GET AND PRINT STRING
410 NEXT D
420 GOTO 310
430 '
440 PRINT "RANDOM RETRIEVAL"
450 GOSUB 610  'REPOSITION TABLE
460 GOSUB 680  'GET AND PRINT THE STRING
470 GOTO 310
480 '
490 PRINT "RANDOM UPDATE"
500 GOSUB 610  'REPOSITION TABLE
510 PRINT "OLD=";
520 GOSUB 680  'GET AND PRINT OLD STRING
530 PRINT "NEW=";
540 LINEINPUT A$
550 'THE NEW STRING MAY BE LONGER OR SHORTER
560 X = USR3(A)  'REPOSITION TO ENTRY A
570 GOSUB 750    'LOAD UPDATED STRING TO TABLE
580 GOTO 310
590 '
600 'REPOSITION TABLE FOR RETRIEVAL OR UPDATE
610 PRINT "ENTER A NUMBER FROM 0 TO"; C
620 INPUT A
630 IF A < 0 OR A > C THEN 610
640 X = USR3(A)
650 RETURN
660 '
670 'GET STRING AT POSITION SET BY PREVIOUS CALL TO
     USR1 OR USR3
680 X = USR1(VARPTR(L$))
690 IF X < 0 THEN PRINT
     : PRINT "L$ IS NOT LONG ENOUGH"
     : SOUND 1, 1
     : STOP
700 PRINT LEFT$(L$,X)
710 RETURN
720 '
730 'LOAD A$ TO THE TABLE
740 'POSITION WAS SET BY A PREVIOUS CALL TO USR0,
     USR2, OR USR3
750 X = USR0(VARPTR(A$))
760 IF X = 0 THEN RETURN ELSE PRINT
     : PRINT "OUT OF MEMORY--TABLE IS FULL"
     : SOUND 1,1
     : STOP
770 '
780 'THIS IS A CONTINUATION (RESTART)
790 INPUT "# OF TABLE ENTRIES"; C
800 RETURN
810 '
820 'LOAD M/L CODE
830 C=0
840 FOR X = S TO S+177
850 READ A
     : POKE X, A
     : C = C + A
860 NEXT X
870 IF C = 18989 THEN RETURN ELSE PRINT C
     : STOP
900 DATA 141, 87, 172, 140, 75, 36, 57, 52, 20, 224
910 DATA 132, 39, 49, 42, 20, 31, 18, 83, 92, 58
```

```
92Ø DATA 166, 128, 167, 16Ø, 172, 14Ø, 53, 37, 247,
     31
93Ø DATA 33, 141, 1Ø3, 32, 27, 174, 14Ø, 42, 31, 18
94Ø DATA 58, 172, 14Ø, 99, 35, 4, 53, 2Ø, 32, 78
95Ø DATA 175, 14Ø, 27, 166, 162, 167, 13Ø, 16, 172,
     97
96Ø DATA 34, 247, 53, 2Ø, 31, 152, 92, 172, 14Ø, 73
97Ø DATA 34, 56, 167, 128, 166, 192, 9Ø, 38, 244, 14Ø
98Ø DATA Ø, Ø, 37, 39, 175, 14Ø, 249, 32, 34, 189
99Ø DATA 179, 237, 31, 1, 238, 2, 23Ø, 132, 142, Ø
1ØØØ DATA Ø, 57, 141, 241, 225, 132, 37, 2Ø, 23Ø, 128
1Ø1Ø DATA 39, 11, 52, 4, 166, 128, 167, 192, 9Ø, 38
1Ø2Ø DATA 249, 53, 4, 79, 141, 41, 32, 3, 2Ø4, 255
1Ø3Ø DATA 255, 126, 18Ø, 244, 141, 12, 175, 14Ø, 197,
     79
1Ø4Ø DAIA 167, 128, 14Ø, Ø, Ø, 35, 249, 57, 189, 179
1Ø5Ø DATA 237, 31, 2, 49, 33, 48, 14Ø, 2Ø, 32, 3
1Ø6Ø DATA 23Ø, 128, 58, 49, 63, 38, 249, 175, 14Ø,
     185
1Ø7Ø DATA 57, 189, 179, 237, 237, 14Ø, 222, 57
```

Resistor Color Code Interpreter

Roy A. Cartier
800 National Ave.
Winchester, VA 22601

Most resistors that have a dissipation rating of 2 watts or less and a tolerance of 5% or higher have their resistance indicated by means of a color code. This color code consists of three or four bands of color around the resistor. In some instances, colored dots are used. The first three bands, beginning with the band closest to the end, indicate the resistance. The fourth band, when used, indicates the tolerance of the resistor. This Model I/III program will interpret this color code.

The colored bands or dots are located closer to one end of the resistor. Beginning with the band closest to the end, type in the color of the band and ⟨ENTER⟩. Enter the color of the second, third, and fourth band in the same manner. If the resistor has no fourth band or only three, just press ⟨ENTER⟩ when the fourth color is requested. The Computer will then interpret the color code and print out the resistance of the resistor and its tolerance. If the resistor does not have four colored bands or dots, its tolerance is 20% or more.

The names for the colors used by this program are the standard color code names which are: black, brown, red, orange, yellow, green, blue, violet, gray, white, silver, and gold. Do not use purple for violet.

If the portion of the program that interprets the tolerance is not needed or wanted, delete lines: 180, 490, 640, 650, 660, 670, 680, and 690.

```
1Ø 'RESISTOR CALCULATOR, NAME IS < RESISTOR >
2Ø CLEAR 1ØØØ
     : DEFSTR A
     : DEFINT B-Z
     : DEFDBL G,H
3Ø AA = "#####.##"
4Ø CLS
     : PRINT@192,""
5Ø PRINT
     : INPUT "1ST. BAND "; A
6Ø IF LEFT$(A,3) = "SIL" PRINT "WRONG END"
     : GOTO 5Ø
7Ø IF LEFT$(A,3) = "GOL" PRINT "WRONG END"
     : GOTO 5Ø
8Ø IF LEFT$(A,3) = "BLA" GOTO 42Ø
9Ø GOSUB 27Ø
1ØØ E = D * 1Ø
11Ø INPUT "2ND BAND "; A
12Ø GOSUB 27Ø
13Ø F = D
     : T = E + F
14Ø INPUT "3RD BAND "; A
15Ø GOSUB 27Ø
16Ø GOSUB 52Ø
17Ø H = T * G
18Ø GOSUB 64Ø
19Ø IF H > 1E5 THEN H = H / 1E6 ELSE GOTO 22Ø
2ØØ PRINT
     : PRINT USING AA; H;
     : PRINT " MEGOHMS, "; AB
21Ø GOTO 5Ø
22Ø IF H > 1ØØØ LET H = H / 1ØØØ
     : GOTO 25Ø
23Ø PRINT
     : PRINT USING AA; H;
     : PRINT " OHMS, "; AB
24Ø GOTO 5Ø
25Ø PRINT
     : PRINT USING AA; H;
     : PRINT " K OHMS, "; AB
26Ø GOTO 5Ø
27Ø D = 2Ø
28Ø IF LEFT$(A,3) = "BLA" D = Ø
29Ø IF LEFT$(A,3) = "BRO" D = 1
3ØØ IF LEFT$(A,3) = "RED" D = 2
31Ø IF LEFT$(A,3) = "ORA" D = 3
32Ø IF LEFT$(A,3) = "YEL" D = 4
33Ø IF LEFT$(A,3) = "GRE" D = 5
34Ø IF LEFT$(A,3) = "BLU" D = 6
35Ø IF LEFT$(A,3) = "VIO" D = 7
36Ø IF LEFT$(A,3) = "GRA" D = 8
37Ø IF LEFT$(A,3) = "WHI" D = 9
38Ø IF LEFT$(A,3) = "SIL" D = 11
39Ø IF LEFT$(A,3) = "GOL" D = 12
4ØØ IF D = 2Ø PRINT "WRONG COLOR"
     : GOTO 5Ø
41Ø RETURN
42Ø D = Ø
     : INPUT "2ND BAND "; A
43Ø GOSUB 27Ø
44Ø E = D
45Ø INPUT "3RD BAND "; A
46Ø GOSUB 27Ø
47Ø GOSUB 52Ø
48Ø H = E * G
49Ø GOSUB 64Ø
5ØØ PRINT
     : PRINT USING AA; H;
     : PRINT " OHMS, "; AB
51Ø GOTO 5Ø
52Ø IF D = Ø THEN G = 1
53Ø IF D = 1 THEN G = 1Ø
54Ø IF D = 2 THEN G = 1ØØ
55Ø IF D = 3 THEN G = 1E3
56Ø IF D = 4 THEN G = 1E4
57Ø IF D = 5 THEN G = 1E5
58Ø IF D = 6 THEN G = 1E6
59Ø  IF D > 6  AND  D < 11  PRINT "LAST  COLOR
    WRONG"
     : GOTO  5Ø
```

Installing TRS-XENIX On Your TRS-80 Model 16

Last month Radio Shack released the TRS-XENIX multi-user operating system. A product overview appeared in the February issue of the Microcomputer News, and this month we in Computer Customer Service decided to discuss the installation of the operating system.

The first thing that must be considered in implementing the TRS-XENIX operating system is the configuration of equipment required. The TRS-XENIX operating system is designed to work on a Model 16, enhanced Model 12, or enhanced Model II with a hard disk unit, and one or two terminals. (A floppy disk version will be available at a later date, but will be much more limited in scope.) The printer is connected to the computer for all printed output. Memory required to run the system is only 256 KB, however, 384 or 512 KB will allow the system to operate noticeably faster.

Before installing the TRS-XENIX operating system some consideration should be given to the choice of a system manager or administrator. One person in your company or organization (or yourself) should be designated as the system manager. The system manager is the person who will control the overall usage of the system. It is extremely important that one person be delegated this responsibility. The system will function much more smoothly with one person overseeing its operation. The system manager will be responsible for the installation of the TRS-XENIX system; adding and removing user accounts, passwords, and files; ensuring that system resources are used efficiently; and protecting the system's data with proper security and backup procedures. This responsibility may seem awesome, but it shouldn't. Every system needs someone to oversee its day to day activity. This is the system manager.

After you have the necessary equipment the next step is to initialize the hard disk drive with the TRS-XENIX Multi-User Operating System Boot Disk. We have tried to make the installation as simple as possible, without a lot of computer mumbo-jumbo, but some technical information will be required. You will need to have your hard disk's "Media Error Map," and you will be asked the number of cylinders and number of heads. If you have an eight megabyte hard drive, there are 256 cylinders and 4 heads. A twelve megabyte hard drive has 230 cylinders and 6 heads. The media map is located in the plastic envelope on the underside of the hard drive.

To start, press the "reset" switch on the front panel of your computer, and press the ⟨**BREAK**⟩ key, followed by the ⟨**REPEAT**⟩ key, and press the ⟨**BREAK**⟩ key again. The message "insert diskette" should appear on the screen. At this point, insert the Boot Disk in drive 0 and close the disk drive latch. The computer will "boot" from the floppy diskette and the screen will now display the message "XENIX Boot" with a ":" appearing on the next line. Type the program name "diskutil" and press ⟨**ENTER**⟩. The following questions will appear on the screen. Answer each of them in turn, as indicated below.

"Diskutil: hard or floppy disk (h or f)?"

Answer "h" to indicate that you want to format the hard disk. The computer will then prompt you:

"Copy or format (c or f)?"

Answer "f" for format.

"Hard disk unit number (0 . . . 3)?"

Answer "0" to indicate the first hard disk. The computer will now prompt you:

"consult your Hard Disk Owner's Manual
for the following . . .
How many cylinders?
How many heads?"

Answer these questions with the appropriate information (which was discussed above). The computer will then ask you to refer to your "Media Error Map" and enter the track and head numbers of any flawed tracks. Type "done" when you have finished.

The system will respond:

"About to format hard disk drive 0"

A message will inform you how long the formatting process will take for your hard disk. You will then see the number of the cylinder and head change as they are formatted.

"Formatting cylinder xx, head xx"

When the formatting is completed, you will see the message:

"Hard disk drive successfully formatted.
Drive parameters and MEDIA ERROR MAP
successfully written. Your hard disk is
ready for the XENIX initialization."

The next step in the installation process is to run a program called "hdinit". This program copies the boot track to the hard disk, copies the rest of the boot floppy to the hard disk, and then automatically shuts the system down.

To run "hdinit", simply boot the system from the floppy disk again. When the boot prompt appears, press the ⟨**ENTER**⟩ key.

"XENIX
:⟨**ENTER**⟩"

The system will respond as if you had typed in the word "xenix" after the colon. Hdinit will begin automatically asking the following questions.

"Do you wish to initialize your hard
disk?"

Answer "y" for yes. You will then be asked:

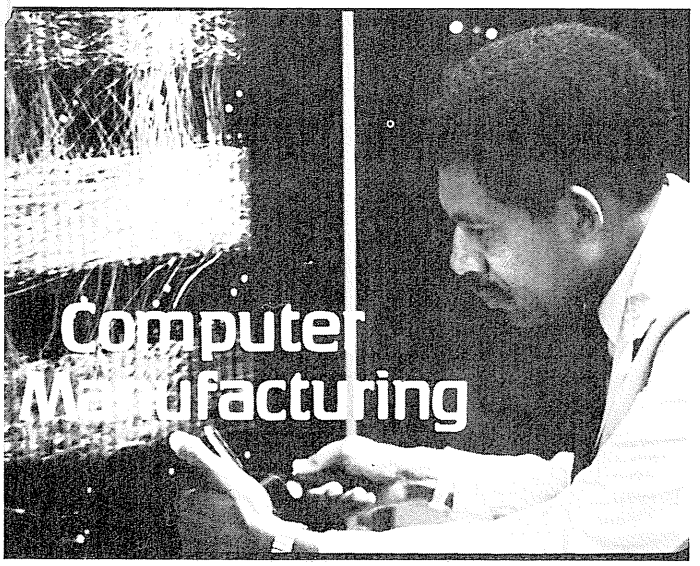
"Has your hard disk been formatted with
diskutil?"

(Continued on Page 28)

Exploring Computer Careers with Radio Shack's *Careers in Computing*

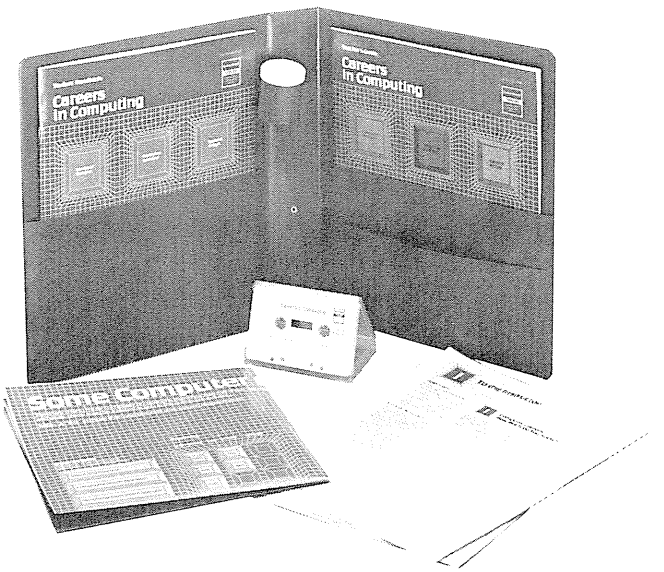
Radio Shack's new *Careers in Computing* package, Catalog Number 26-2758, was released earlier this year. The multi-media instructional kit is designed to provide secondary students with the information they need to begin considering a computer-related career. *Careers in Computing* presents a broad look at the field, and makes the point that no matter what your interests, there can be a computer career for you.

The narrated film strip in *Careers in Computing* tells the story of two high school students (a male and a female) who are assigned to report on computer careers for a career day at their school. Guided by the uncle of one of the students, they take a tour of a computer center and discover that you don't have to be a math whiz to work with computers. Data entry, programming, management, and computer manufacturing careers are illustrated as the tour progresses.



The Teacher's Guide and the 30 Student Handbooks in *Careers in Computing* present a close look at the major areas of computer careers and at specific jobs within each area. Discussions of each job include the job's general responsibilities; the education, experience, and personal characteristics recommended; the future outlook; and related job titles. Among the many non-programming jobs featured are: computer center librarian, technical writer, data entry operator, production controller, service technician, and sales representative. Careers requiring different levels of education (high school only, college, or graduate school) are highlighted. The student book also provides some hints as to how, and with what kinds of companies, you can find a computer career.

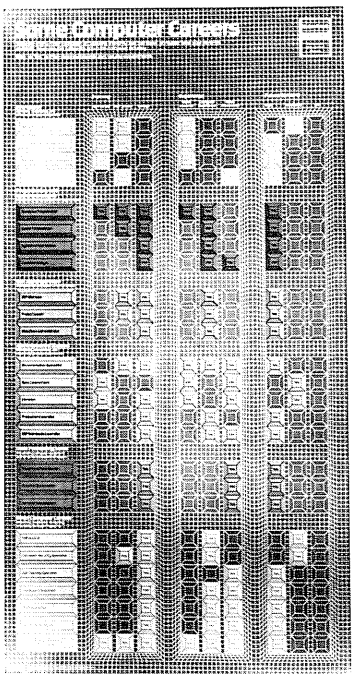
The information in the student book is specific enough to provide a real sense of representative computer careers and how various computer careers fit together. But at the same time it is general enough to be useful in dealing with many kinds of companies in all parts of the country.



An important part of *Careers in Computing* is the set of eight spirit masters for student activities. The first activity is a true-false pre-quiz entitled "How Much Do You Know?" One of the objectives of this quiz is to get the student to reject myths like: "There's no place for an athlete, teacher, or musician in the computer industry."

Another activity master asks students to look at their own skills—do they like working with details?, do they think they have leadership skills?, etc. Activities like this one, like activity number 5 (which gives students practice in resume writing), and like activity number 6 (an exercise in completing an application), are usefully thought-provoking no matter what career field a student chooses. Activity number 7 presents a script for an interview—recommended use is for the class to role-play the interview, then discuss strengths and weaknesses. Other activities ask the student to match his or her skills to computer careers, match student profiles to computer careers, and test their understanding of the future outlook for particular computer careers.

A colorful, eye-catching wall chart for the classroom provides graphical reinforcement on the training, experience, and future demand associated with the computer careers discussed in the handbook.



HOW YOU CAN USE *CAREERS IN COMPUTING*

Careers in Computing can be used in many ways, including the following:

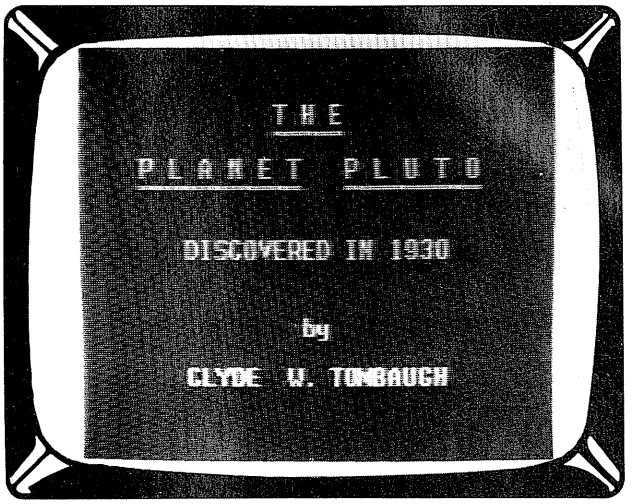
- As a unit on career opportunities, within a secondary computer science program
- As part of a secondary business class or vocational education class
- As a workshop for interested students, as part of the secondary school's or college's vocational counseling program
- As part of an adult education or continuing education program

Radio Shack's *Careers in Computing* provides an interesting, informative way to help students explore careers in today's computer society. The instructional materials cut through the mystery and mythology that surround computer careers and make students aware of some of the options available to them. Whether or not the students decide that they want to work directly with computers, they gain a better understanding of the world in which they will be seeking a job and of job-seeking methods.

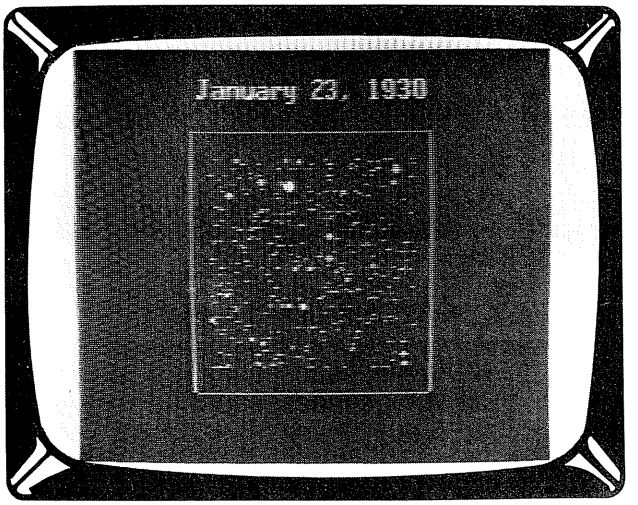
To find out more about the *Careers in Computing* kit, contact your local Radio Shack store or Computer Center, or contact the Radio Shack Regional Educational Sales Coordinator in your area. The complete *Careers in Computing* package, including Teacher's Guide, film strip, 30 Student Handbooks, wall chart, and spirit activity masters (Cat. No. 26-2758) is listed at $130.00. Extra Student Handbooks are available separately (Cat. No. 26-2759) at $1.00 each. (Prices may vary at individual stores and dealers.)

Discovering Pluto with Dr. Tombaugh and Talk/Tutor

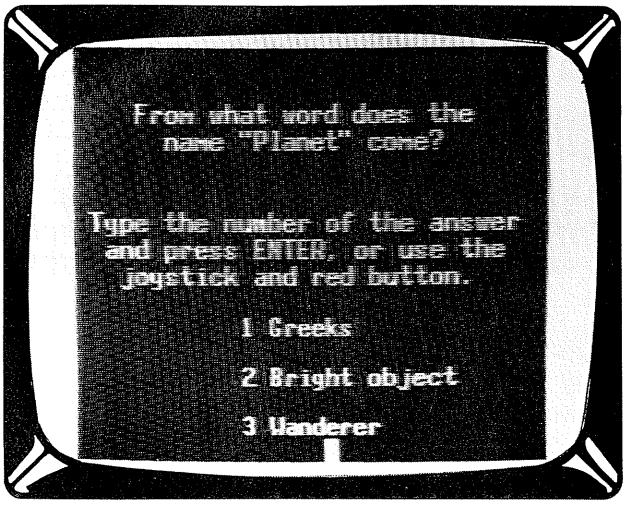
In February, 1930, Dr. Clyde W. Tombaugh used a "blink comparator" to compare two astronomical photographs, and he found what he was looking for—he had discovered the planet Pluto.



Radio Shack's new Color Computer program "The Solar System: Featuring the Discovery of the Planet Pluto" presents an informative audio-visual introduction to our neighboring planets that recreates some of the excitement of that discovery. Designed for the 16K or 32K TRS-80® Color Computer tape system, the program was created using the Talk/Tutor development system. The program features high-resolution graphics, display of upper- and lower-case text, and recorded speech. Dr. Tombaugh, now a Professor Emeritus of astronomy, narrates part of the program himself. He tells how he searched for a trans-Neptunian planet and finally captured the planet photographically.

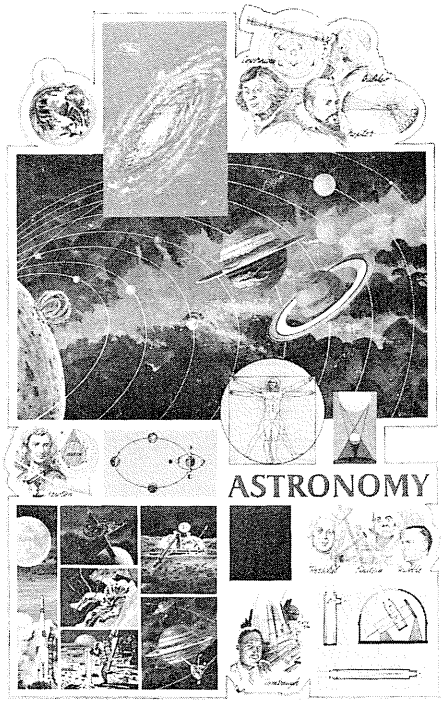


Questions over the lesson material are presented during the program. Students have the option of either typing answers at the Color Computer keyboard or using joysticks to answer. Immediate feedback is given; students receive reinforcement for correct answers and are informed of incorrect answers. At the end of the lesson, a report screen displays the number of questions that the student answered correctly.



A book by Professor Tombaugh and the British astronomer Patrick Moore is included in the Radio Shack package. *Out of the Darkness: The Discovery of the Planet Pluto* provides further information on how astronomers observe planets and how Pluto was discovered. The user's manual for the Radio Shack program includes a "Suggested Activities" section that presents ideas for individual and group activities based on the program and the book. (26-2647).

A colorful wall poster in the package highlights milestones in astronomy. The people, events, and tools of astronomy that are featured on the poster are ideal subjects for reports to the class or other special projects.



Network 3 Menu-Maker Utility

by Jack Tjarks

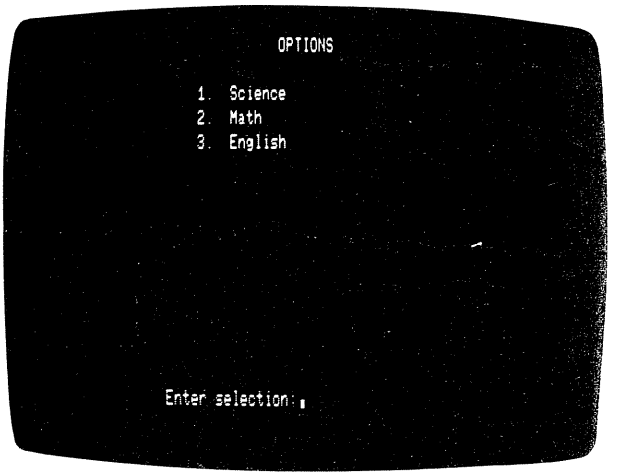
The Menu-Maker Utility is part of Radio Shack's Network 3 Operating Software package (Catalog Number 26-2775). It was included as a utility in the package to give the teacher a powerful tool to use in making a variety of BASIC programs easily available to students. Using menus generated with this utility, students who are not experienced in using computers can have access to many CAI programs and to many BASIC programs which help them begin learning how computers work.

The utility was written for teachers who have no experience programming in BASIC. A knowledge of programming is not necessary to use the Menu Maker. In fact, the only information necessary to use this program is an understanding of what a menu is (a list of choices) and a plan that you want to put into menu form.

Menu programs tell the student what his or her options are and let the student communicate choices to the computer. This requires that a list exist (displayed on the screen) for the student to choose from. A second list should inform the computer what filename is associated with the student's choice. (The name used on the menu does not have to be the same as the filename of the program.) This use of two lists allows the teacher to use appropriate cue words or phrases when listing the choices for the student, and the language of the computer does not interfere with the learning process.

An important function of menu design is control of the learning process. The menu allows the student to direct his or her own learning, while it allows the teacher to decide which programs the student may choose from. This control is powerful and at the same time low key.

The teacher uses the MENU program to create a program which we will call "CHOICES." This program contains a list of up to nine choices available to the student. When the CHOICES program is called, the student might see a screen like this one:



(Continued on Page 22)

The Software Maintenance Program

by Annette Zamberlin-Main

Editor's Note: Because of the importance of the new Software Maintenance Program, and in case you may have missed the first printing or would like to review it again, we felt that the reprinting of this article from the February 1983 issue would be advisable.

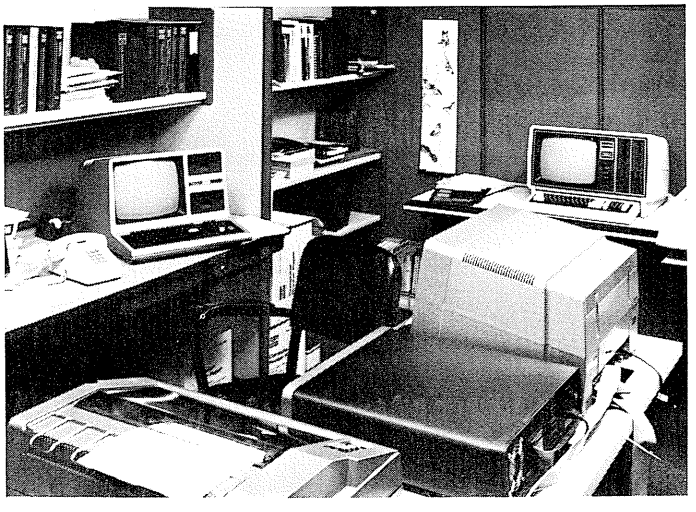
In the January 1983 issue of the Microcomputer News, Mr. Jon Shirley made an announcement in "View from the 7th Floor," that we thought you might like some additional background and information about.

Mr. Shirley dedicated two paragraphs to a new policy for software support, which began here on the January 1, 1983 and is quite exciting when you consider all the ramifications.

PRESENT SYSTEM

Presently, the means employed for assisting customers (i.e. answering questions and generally assisting during the implementation phase) works very well, as far as it goes. The problem is, it doesn't go quite far enough. Radio Shack not only wants to help by providing answers to your questions (which we presently do), we also want your software to be maintained and kept up-to-date. By keeping your software updated with the latest improvements, you will need less and less "one-on-one" assistance and that is good for both of us.

But, back for a moment to the present system. Up until now you have probably depended upon the Radio Shack Computer Center, Computer Departments or participating Radio Shack dealers to answer your questions, to keep you informed of changes to your software and otherwise assist you in using your computer system.



The personnel in these stores are further backed by a very competent group of individuals at Computer Customer Services in Fort Worth. The customer can call direct to Computer Customer Services about any problems he is experiencing. (See Computer Customer Service article in this issue for appropriate telephone numbers.)

And last, but surely not least, we (the *Microcomputer News*) would publish entire sections in our newsletter dedicated solely to the bugs, errors and fixes for each software package, under its respective model as we became aware of them during the course of a given month. As an aside, you may have noticed, with the January issue that the various "Bugs, Errors and Fixes" sections have now become one section in the News, subdivided by model.

An additional level of support is being added to this existing system which is not matched by any other manufacturer/retailer of microcomputers. That additional level of support is to ensure that each user of a specific software package is kept current on all changes, revisions, etc. for their particular package.

With the advent of our April issue, a listing of letters sent to specific software package owners will appear along with any detailed corrections to games and/or color software.

This additional level of support is based upon a highly accurate mailing list, which in turn is based upon the completed and returned software registration cards, which the customer finds contained in all disk-based software binders (games and color software excluded).

Software Registration Card

Cat. No. _____

Version _____

SAMPLE

Name _____

Company _____

Address _____

City _____ Phone (___) ___ - ____

State _____ Zip _____

CUSTOMERS' RESPONSIBILITIES

If you already own a disk-based (non-game) software package and have not sent in the software registration card included within, do so at once.

R E A D M E F I R S T

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.

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.

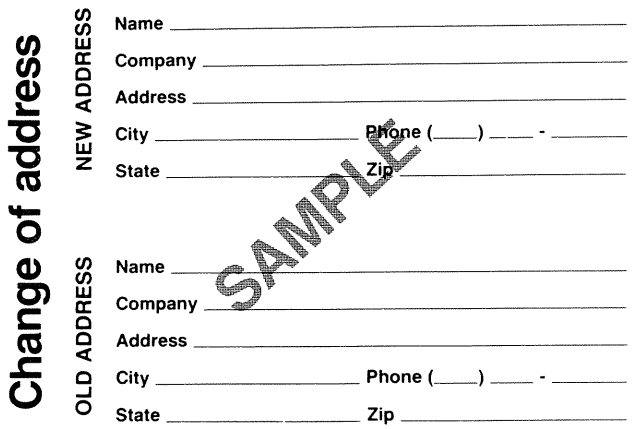
3) As we furnish updates or "patches", and you update your software, you must keep an accurate record of the current version numbers on the logs below. (The version number will be furnished with each update.)

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
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

You might also want to contact your local Radio Shack store where you purchased the package or telephone Computer Customer Services in Fort Worth to find out if there have been any revisions, corrections, etc. to the software since your date of purchase, which you might have missed. What you are attempting to do here is to insure that your software is as up-to-date as possible, so that any future corrections will fit right into place. Be sure as well, to note these revisions somewhere in the package for later reference.

We have included a sample of our new Version Log with this article. This will be placed in the front of all disk-based, non-game software packages that are produced after January 1 of 1983. You may want to take a minute to copy this sample log and place it in the front of all software packages that you presently own. The Version Log will be handy for keeping track of revisions that are made under the auspices of this program. The only time that you should make an entry in this log is when we notify you by mail of an update. In the Customer Service Bulletin itself, we will provide you with a new version number to enter into your log, if applicable. By using this method, we can try to keep track of your version number in Fort Worth, knowledge of this number can be very helpful when inquiring about your software.



Change of address cards have also been included. The currency of the mailing list will be the primary responsibility of the customer. We need to know where you are in order to insure that you receive the information in a timely fashion.

WHAT TO EXPECT

And what exactly will be coming in the mail? A first class letter will be sent to you, as the need arises, which will inform you of all PATCHes and re-releases (you may know them as Radio Shack's 700 numbers).

The types of changes involved include four major categories:

(1) A serious problem fix.
(2) A simple necessary PATCH.
(3) A low level obscure fix.
(4) A special situation PATCH.

If a re-release of a program is made, information on the new "700-XXXX" number will be contained in the letter. No disks will be sent thru the mail. It will be the customers' responsibility to go to his local Radio Shack Store, Computer Center, or participating dealer to pick up the disk and any additional new documentation. All detail information will be included in the letter.

Radio Shack in Fort Worth will also send the same letters to all stores and computer centers.

So please, send those software registration cards in. You need to be kept up-to-date on all corrections or revisions that might be made to the software package(s) which you own and Radio Shack has the mechanism in place to keep you current.

COCOTEST

David J. Stretch
Box 408, U.S.S.A.H.
Washington, D.C. 20317

Here is a program I use to check the color, alignment and tone on my TV set. It is also an interesting random display.

```
9Ø REM "COCOTEST"
92 REM FOR TRS-8Ø COLOR COMPUTER
1ØØ CLSØ
1Ø2 FOR H=14 TO 49
1Ø4 FOR V=6 TO 25
1Ø6 SET(H, V, 7)
     : NEXT V, H
1Ø8 C=RND(8)
11Ø A=RND(3Ø)
112 B=RND(16)
114 XX=RND(5)
116 YY=RND(3)
118 X=A+13
12Ø Y=B+5
122 FOR H=X TO X+XX
124 FOR V=Y TO Y+YY
126 SET(H,V,C)
     : NEXT V, H
128 S=A + B + C + X + Y + XX + YY
13Ø IF T=1 THEN 132 ELSE 138
132 FOR SO=S TO S+S STEP A+B+C
134 SOUND SO, 1
     : NEXT SO
136 T=Ø
     : GOTO 1Ø8
138 FOR SO=S+S TO S STEP -(A+B+C)
14Ø SOUND SO, 1
     : NEXT SO
142 T=T+1
     : GOTO 1Ø8
```

Computer Customer Service Address and Phone Numbers

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

Model I/III Business Group	(817) 390-3939
Model II/16 Business Group	(817) 390-3935
Languages and Compilers	(817) 390-3946
Color and Pocket Computer Group	(817) 390-3944
Hardware and Communications Group	(817) 390-2140
Educational Software	(817) 390-3302
Games, Books, and New Products	(817) 390-2133
Newsletter Subscription Problems	(817) 870-0407

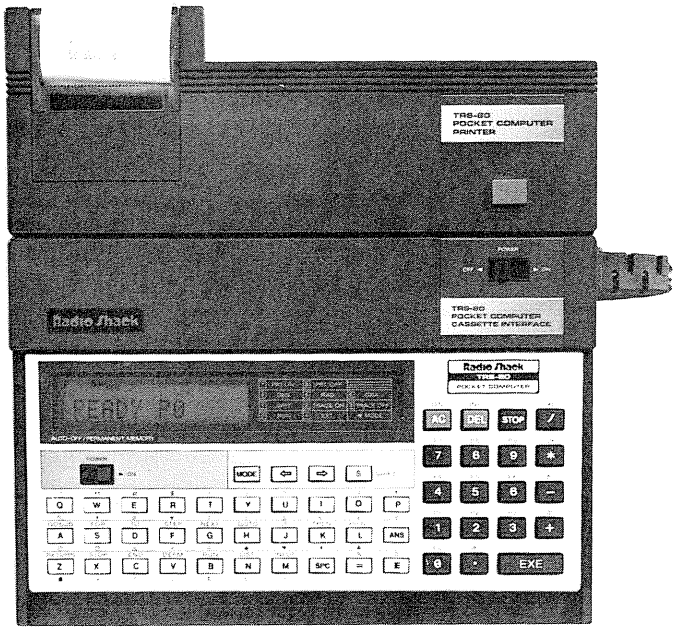
The PC-4 Has Arrived!

by Linda Miller

The Pocket Computer 4 is an inexpensive and versatile computer for anyone wanting portability and programming power in a small unit. It has capabilities that are unique to it as well as many that are similar to the PC-1 or the PC-2.

JUST HOW BIG IS IT?

There are three separate components that make up a complete PC-4 system: the PC-4 pocket computer (26-3650), the cassette interface (26-3651), and the printer (26-3652). The PC-4 is 3/8" high by 6 1/2" wide by 2 3/4" deep. The cassette interface is 1 1/4" high by 6 7/8" wide by 4 1/2" deep. The printer dimensions are 1 3/8" high by 6 7/8" wide by 1 1/2" deep. The total dimensions can be misleading because all three components fit together to form one unit that is approximately 1 7/16" high by 6 3/4" wide by 7" deep.



Even when all these components are connected together, the whole PC-4 system will fit into anything from a medium sized purse to a briefcase. It really is compact.

THE DISPLAY

Up to 12 characters can be viewed at one time on the display. Each character is displayed within a 5 horizontal by 7 vertical dot matrix. If the 12th character is entered, all the characters on the display shift to the left one character. As additional characters are entered the display will continue to scroll to the left to allow a total of 62 characters to be input on a single line. To forewarn the user that the total number of characters allowed per line has almost been reached, the cursor changes from the blinking "_" character to a blinking "■" character after the 55th character is input.

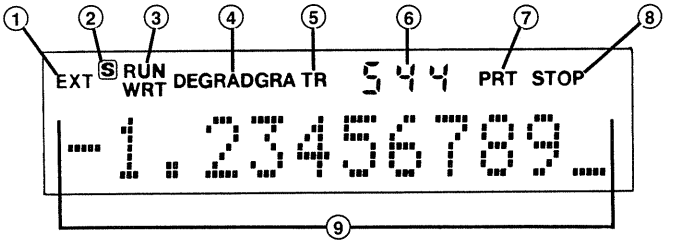


Figure 1. A representation of the different information that can be shown on the display.

1. Extension mode
2. Shift in/out
3. Run/write mode
4. Angular units
5. Trace on/off
6. Number of program steps remaining
7. Print mode on/off
8. Stop mode
9. Twelve character display

THE KEYBOARD

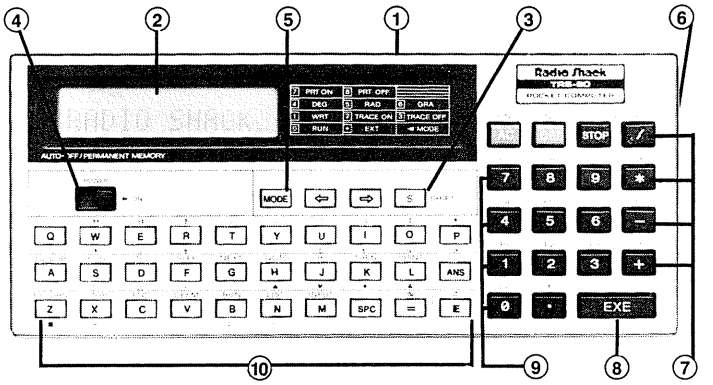


Figure 2. The PC-4 Keyboard

1 Adapter connector
2 Display window
3 Shift key
4 Power switch
5 Mode key
6 Display contrast control
7 Calculation instruction keys
8 Execute key
9 Numerical keys and decimal point key
10 Alphabet keys

The display contrast control (6) is located on the right side of the computer, and the adapter connector is located on the top back side of the computer so neither is visible in Figure 2.

KEY OPERATIONS

Some keys such as ⟨SPC⟩ have only one operation while other keys have two. Each alphabetic key has four

different operations. There's more than meets the eye on this little computer.

Ⓢ (Shift)—This key is the only red letter key on the keyboard and should be easily distinguished from the letter "S" key. When the shift mode is selected by pressing this key, a Ⓢ appears on the display to indicate that shift in is engaged. By pressing the Ⓢ key again, Ⓢ disappears from the display, indicating that shift in is not engaged.

← → Cursor Keys—Pressing the right or left arrow key moves the cursor to the right or left.

〈AC/ON〉 Key—When 〈AC/ON〉 is pressed one of four things may happen.

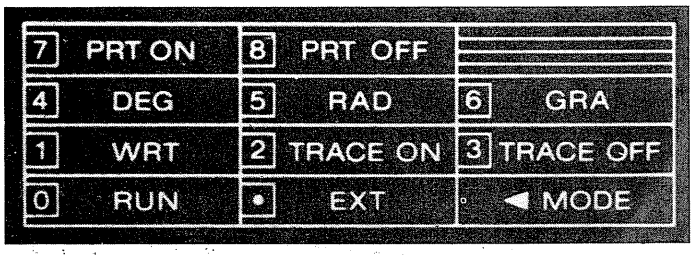
1. The display will be cleared.
2. When a program is running, the 〈AC/ON〉 can be used as a 〈BREAK〉 key to stop program execution.
3. If an error message appears, then pressing this key will clear the display.

〈STOP〉 Key—If pressed during program execution, "STOP" will be displayed, and program execution will stop at the end of the program line currently being executed. During program execution, pressing the stop key will cause the computer to display the currently selected program area (P0 - P9) and the program line number just executed.

〈EXE〉 Key—In manual calculation, the EXEcute key is pressed instead of the = to get the result. When entering program lines in the WRT mode, pressing 〈EXE〉 stores the program line in memory. If it is not pressed, the line will be lost. In the RUN mode, 〈EXE〉 is pressed after inputting data or to continue program execution after the STOP message is displayed.

〈ANS〉 Key—This 〈ANS〉 key is used to recall the result of the most recent calculation performed. To add the quotient of 100/.6 to the quotient of 33/.5, perform the first calculation 100/.6 〈EXE〉 which results in 166.6666667. Next divide 33 by .5 and add 166.6666667 to the result by pressing the 〈ANS〉 key which recalls the result of the first calculation. The key sequence 33/.5+〈ANS〉 〈EXE〉 results in 232.6666667.

〈MODE〉 Key—By pressing the 〈MODE〉 key with the 〈.〉 and 〈0〉 through 〈8〉 keys the computer's condition or angular unit can be designated. Figure 3 is a representation of the rectangle which appears to the right of the PC-4 display window.



〈MODE〉 〈.〉—EXT is displayed on the display window indicating that the extension mode has been designated and lower case or special symbols may be selected. Pressing 〈MODE〉 〈.〉 again releases the extension mode.

〈MODE〉 〈0〉—RUN is displayed and manual calculation and program execution can be performed.

〈MODE〉 〈1〉—WRT is displayed and program write-in and checking/editing can be performed.

〈MODE〉 〈2〉—TR is displayed. A trace of program execution can be performed.

〈MODE〉 〈3〉—TR disappears from the display and trace is turned off.

〈MODE〉 〈4〉—DEG is displayed thus designating the angular unit as degree.

〈MODE〉 〈5〉—RAD is displayed, and the angular unit is designated as radian.

〈MODE〉 〈6〉—GRA is displayed, and the angular unit is designated as gradient.

〈MODE〉 〈7〉—PRT is displayed and printing is enabled.

〈MODE〉 〈8〉—PRT disappears from the display and printing is disabled.

SHIFT IN OR SHIFT OUT

Some operations are accomplished depending on whether the Ⓢ (shift) key is in or out. "Shift out" means that the shift key has not been pressed while "Shift in" means that the Shift key has been pressed. For example, notice the GOSUB that is located over the letter A. Pressing the A key causes a capital letter A to appear on the display. If, before pressing the A key, the Ⓢ key is pressed, then instead of an A, the word GOSUB appears.

There are eighteen BASIC words that appear above the alphabetic characters in the bottom two rows. The ability to print these commonly used BASIC words with only two keystrokes can be time saving when writing a BASIC program.

All the alphabetic keys are shifted in to get the character or word that appears above the key. Other keys that are affected by the shift in mode are the 〈DEL〉 (as is to delete a character, shifted to insert a character), numeric keys (0-9), calculation instruction keys (/, *, -, +), the decimal key (.), the equal key (=), and the exponent key (Ⓔ).

THE EXTENSION MODE

But there's only an upper case alphabet, right? Lower case can only be found on more expensive machines. Not true! Lower case letters are available on the PC-4. How does one get lower case characters, and what about all those other characters that appear below the alphabetic keys—the heart, club, diamond, and spade for instance? By using what is called the Extension Mode the lower case and special characters can be accessed.

The Extension Mode is entered by pressing the 〈MODE〉 key and then the 〈.〉 (decimal) key which is located between the 〈0〉 and the 〈EXE〉 keys on the lower right hand side of the numeric keypad. After pressing 〈MODE〉 〈.〉, EXT appears in the upper left hand corner of the display. There is a shift in and a shift out operation in the Extension Mode also. When in shift out which is operative unless the Ⓢ key is pressed, pressing any alphabetic key will result in the lower case letter. By pressing the Ⓢ key before an alphabetic key is pressed, the twenty-six special characters (one for each letter of the alphabet) will be displayed.

The small size of the keyboard may be a little misleading when one stops to realize that with the twenty-six alphabetic keys alone there are a possible 104 different characters including BASIC words and upper and lower case, and special characters.

MEMORY

How much memory is there? The PC-4 comes with a 544 step, 26 variable memory (plus an exclusive character variable memory) RAM which may be increased to a maximum

of 1568 possible steps or up to 222 memories with the addition of the optional, user installable RAM Expansion Pak (26-3653).

POWER

The PC-4 requires two batteries (Cat. No. 23-162) with a battery life of approximately 360 hours of continuous use. The PC-4 does have an auto power-off feature so that if the user forgets to turn the power off, the computer will automatically shut off after 7 minutes of non-use. When the computer is turned back on, the program and variable contents will still be intact.

CALCULATING ON THE PC-4

Manual calculations are performed in the RUN mode, and calculations are done in a priority sequence. The Calculation priority sequence is as follows:

1. Functions (SIN, COS, TAN, etc.)
2. Power
3. Multiplication and division (* and /)
4. Addition and subtraction (+ and −)

When priorities are the same, calculation will begin from the left. Calculations enclosed within parentheses get the highest priority.

The calculation range for the PC-4 is from 1×10^{-99} to $+9.99999999999 \times 10^{+99}$ and 0.

The functions available on the PC-4 are:

Function Name		Form
Trigonometric functions	sin x	SIN x
	cos x	COS x
	tan x	TAN x
Inverse Trigonometric functions	$\sin^{-1}x$	ASN x
	$\cos^{-1}x$	ACS x
	$\tan^{-1}x$	ATN x
Square root	$\sqrt{x}$	SQR x
Exponential function	e	EXP 1*
Natural logarithm	ln x	LN x
Common logarithm	log x	LOG x
Change to integer	INT x	INT x
Deletes integer portion leaving only the fraction	FRAC x	FRAC x
Change to absolute value	|x|	ABS x
Signum function	positive number →1 0→0 negative number → −1	SGN x
Round off	(round off x at 10^{y})	RND(x,y)†
Random number		RAN#

*EXP is a command to call out the numerical value of the exponential table.

† In the case of the RND function, the argument must be enclosed in parentheses.

THREE VARIABLE TYPES

The PC-4 supports 3 variable types: numeric, character, and exclusive character.

A numeric variable is designated by a single capital letter, A - Z which may or may not be subscripted (followed by a number in parentheses). A numeric variable is assigned a value such as 1.59, 33.45, 10, etc. This value may be assigned directly as W and X are below, or the value may be derived as the result of a mathematical calculation as are Y and Z below.

```
W=45
X=A(1Ø)//@
Y=34/6.7*6
Z=INT(W)
```

A string (character) variable is designated by a single capital letter followed by a $ (dollar sign). String variables also may or may not be subscripted. A$ (pronounced A string), C$, and Z$ are all character variables which represent a group or string of characters such as 123, HELLO, or $23.50. A string variable can be assigned by a statement like T$ = "TIME".

Notice that the string of characters (TIME in this example) being assigned to a string variable is always enclosed in quotation marks during the assignment process unless the string is being entered directly as it would be in an INPUT statement. When T$ is printed, only the characters TIME will be printed. String variables on the PC-4 can be up to 7 characters long.

The third type of variable supported by the PC-4 is the exclusive character variable. There is only one exclusive character variable, and it can contain up to 30 characters and is designated by the "$" character.

```
$="Captain John Smith"
```

Only the exclusive character variable can be used with the MID function.

ARRAYS

The PC-4 can support single dimension arrays which are designated by subscripted variables. In the variable A(5), A is the variable name and (5) is the subscript. A(6) is a different variable than A(5).

The normal or default number of variable memories is 26, but by using the DEFM statement, this number can be expanded according to the amount of memory available. DEFM 15 would increase the number of variable memories from 26 (the default number) to 26 + 15 or 41.

Each variable memory added decreases the number of program steps available by eight steps. When there are 26 memories, there are 544 program steps (in a non expanded memory) available. If the number of variable memories is increased to 27, then the number of program steps available is decreased to 536 which is a reduction by eight steps.

It is important to remember when using arrays that the particular memory used by a variable depends on whether the variable is one of the first 26 variables (A-Z) or the argument of the variable. Note the following:

```
A=A(Ø)
B=A(1)=B(Ø)
C=A(2)=B(1)=C(Ø)
```

If B(1) is assigned the value of 10, then variables C, A(2), and C(0) will all have the value 10. If a program contains a variable A which is equal to 25 and during program execution the variable A(0) is assigned the value 30, then A will also have the value of 30.

PROGRAM AREAS

By entering the WRT mode (⟨MODE⟩ ⟨1⟩), a program can be written in one of the 10 separate program areas which are indicated by P0, P1, P2, . . . , P9 above the 0-9 keys on the numeric key pad. This unique memory arrangement makes it possible to define a variable in one program and use it in another or to access a subroutine stored in a program

area completely different from the one currently designated. For example, program area P0 might contain the one line program 10 X=25, and program area P1 contains the program 10 Y=X/2:PRINT Y. By first running P0 and then running P1, the value of Y will be printed as 12.5.

By using the different program areas, it is possible with the PC-4 to have several different programs in memory at the same time or a single larger program or whatever combination the operator desires and has room for.

ERROR MESSAGES AND EDITING

When errors occur on the PC-4, error messages regarding everything from syntax error to nesting errors will be displayed. There are nine separate error codes numbered from 1 to 9.

Characters may be changed by moving the cursor by using the right and left arrow keys so that it is positioned over the offending character and keying in the correct character. Characters can be deleted by moving the cursor to the position where the character-to-be-deleted is and pressing the ⟨DEL⟩ key to delete the character. If instead of deleting, it is necessary to insert a character, then by positioning the cursor on the character that would follow the inserted character and pressing (S) ⟨DEL⟩, a space opens up so that the character can be entered.

Programs may be examined by listing them in either the RUN or WRT mode. In the RUN mode after LIST is typed, each line will be displayed for approximately 2 seconds. By pressing the ⟨STOP⟩ key, the listing will stop until the ⟨EXE⟩ key is pressed. In the WRT mode, each line is listed, and the next line will not be displayed until the ⟨EXE⟩ key is pressed. Insertions and deletions to program lines may be made using the ⟨DEL⟩ key as described above. The ⟨EXE⟩ key should always be pressed at the end of editing to store any line changes.

PROGRAM DEBUG

Debugging capabilities can be divided into two categories: table top debugging which is done during the writing and logic checking of the program by the operator and conversation type debug where the computer displays error messages during program execution. For instance, if program line 10 contained a syntax error then the message ERR 2 P0-10 would appear on the display. This message tells the user that a syntax error occurred in line 10 of the program in program area P0.

Still another feature of the PC-4 is the ability to follow program execution one line at a time in the trace (TR) mode. Program errors can be isolated to a single line in this mode. Additionally, the STOP command can be used to stop the progress of program execution.

THE PC-4 SPEAKS BASIC

Below is a list of the BASIC language commands included in the BASIC language of the PC-4.

INPUT is used to input either numeric or string data into a variable.

```
10 INPUT "ENTER NUMBR", N
10 INPUT "ENTER NAME", N$
```

KEY is used to read one character into a character variable by pressing one key during program execution. KEY differs from INPUT in that it does not wait for operator input. Program execution will continue. (KEY is similar to INKEY$ in other TRS-80 BASICs.)

PRINT is used to display calculation results, data, prompts, or instructions on the display. Some examples of print statements are:

```
PRINT N          'Prints the value of N
PRINT 123        'Prints the number 123
PRINT X/4*5      'Prints the calculation result of X/4*5
PRINT N$         'Prints the string stored in the
                  string variable N$
PRINT "TIP"      'Prints the word TIP
PRINT "A"+"B" 'Joins (concantenates) and prints AB
```

CSR is used with the PRINT statement to position data on the 12 character position (0-11) display or on the printer output. The format for CSR is:

```
PRINT CSR numeric expression; numeric expression
                              character expression
```

The first numeric expression is the position on the display or the printout (indicated by numbers 0-11). The second numeric expression, or the character expression, represents the numeric or string data to be printed.

Where N equals 12345

```
PRINT CSR 5; N
```

will print 12345 beginning in the seventh character position of the display. Seventh? Yes. Since 0 is the first position, CSR 5 actually designates the sixth position and because 12345 is a positive number, it is printed with a leading space for the sign of the number. Therefore 12345 will be printed with a leading space beginning at display character position 7.

GOTO n causes program execution to unconditionally jump to program line number n, while GOTO #n causes program execution to jump to program area n (0-9).

IF, THEN, and ; are available to set up conditional jumps. The IF statement establishes the condition which must be met before execution goes on to what follows the THEN or the ; (IF A = B, IF A/12*5 =20, etc.). THEN n is used to jump to program line n. THEN # n causes a jump to program area n. A ; is used for other instructions such as ;PRINT "THIS IS THE END" or ;X=10*Y.

FOR-NEXT loops can be nested up to four loops. **STEP**is also available for incrementing the loop.

GOSUB and **RETURN** are also included in the PC-4's BASIC. With GOSUB jumps can be made to specific program lines with a statement such as GOSUB 100 or a jump can be made to a program area with a statement such as GOSUB #5. The RETURN statement which will be included at the end of the subroutine causes a return to the calling program. GOSUB statements can be stacked up to 8 levels.

Multistatements on a single program line are possible by inserting a : (colon) between the statements. For example:

```
10 PRINT "TRY THIS ONE.":PRINT "YOU LUCKY DEVIL."
```

The **STOP** command stops execution temporarily. Pressing the ⟨EXE⟩ key will resume execution.

The **END** statement terminates program execution.

RUN is used to execute a program. RUN will execute a program beginning at the first program line, while RUN 50 will execute a program beginning at line 50.

LIST will list the program lines on the screen or printer (if the PRT mode is on). LIST will cause the program to be listed from beginning to end. LIST 50 will begin the program listing

at line 50 and continue through the end of the program. LIST A will list all the programs in all the program areas beginning with P0.

MODE n (4-8) is used to designate the angular unit or printer output condition in a program.

MODE 4 "DEG" designation
MODE 5 "RAD" designation
MODE 6 "GRA" designation
MODE 7 PRINT mode designation
MODE 8 PRINT mode release

These mode commands written in a program can achieve the same effect as manually pressing the MODE key and the appropriate number (4 - 8).

The **SET** command is used to determine the display output format. It can be used to designate the number of effective positions (numbers to the left of the decimal) from 0 to 9 positions and the number of decimal positions (numbers to the right of the decimal) from 0 to 9.

If SET E 5 is performed either manually or in a program then when a number is printed it will have five effective positions. Consider the number 123456 which has six effective positions. After SET E 5 is performed, 123456 ⟨**ENTER**⟩ results in the number 1.2346E05 being printed. 123456 is printed in exponential notation because after SET E 5 is executed, the numbers printed can have only five effective positions, and 123456 has 6.

SET F n (where n is a number from 0 to 9) designates the number of decimal positions that will be printed. If SET F 2 is performed, and the number 1.23456 is entered, then when it is printed, the result is 1.23.

LEN is a function which returns the length of a character variable. When A$ equals "ABCDE", then LEN(A$) is 5.

The **MID** function is only used with the exclusive character variable ($). The format for MID is MID (m,n) where m and n are numerical expressions between 1 and 30, and n is the number of characters to be duplicated, and m is the position number of the character where the duplication is to begin.

If $="ABCDEFGHIJ" then the statement PRINT MID(2,5) will result in the string BCDEF being printed.

If a value for n is not used, then beginning at position n, the character at that position and every character to the end of $ will be printed.

The **VAL** function gives the numeric value of a string of numbers. If A$="123" then PRINT VAL(A$) would yield the number 123.

CLEAR statements are executed in the WRT mode to clear the program in the currently designated program area (P0-P9). CLEAR A clears all programs in all program areas from P0 through P9.

WRITING PROGRAMS AND DATA TO TAPE

SAVE is used to save the program in the currently designated program area to tape. The format for SAVE is: SAVE "filename". ("filename" may be omitted.) The filename must be enclosed in quotation marks and may contain up to 8 characters including alphabetical characters, numbers, and symbols.

SAVE A "filename" ("filename" is optional) will write all the programs in program areas from P0 to P9 to tape.

LOAD "filename" ("filename" may be omitted.) will cause the program with the specified file name to be loaded into the currently designated program area.

LOAD A "filename" ("filename" is optional) loads into program areas P0-P9 programs which were previously written to tape with a SAVE A instruction.

PUT is used to write data to tape. The format for PUT is PUT "filename" variable 1, variable 2 where "filename" and ", variable 2" are optional. Variable 1 and variable 2 set up a range of variables to be written to tape. For instance:

```
PUT "DT"$,A,Z(1Ø)
```

creates a data file called DT, writes the exclusive character variable $ to tape, and writes the variables A through Z(10) to tape.

VER is used to verify that the programs or data have been recorded on the tape properly.

PRINTING

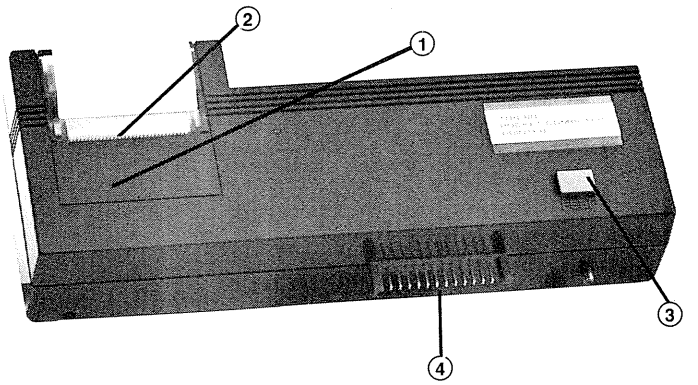
By designating MODE 7 (print mode), program listings may be obtained and anything that normally would be printed on the display will also be printed on the printer.

MODE 7 (printer on) and MODE 8 (printer off) may be written in program lines as in the following example program.

```
1ØØ MODE 7
11Ø PRINT A
12Ø MODE 8
```

When listing a program to the printer in the RUN mode, the listing will be printed continuously without stopping. When listing a program in the WRT mode to the printer, printing will stop after one line has been printed and wait until the ⟨**EXE**⟩ key is pressed.

THE PC-4 PRINTER



1 Printer Cover
2 Paper Cutter
3 Feed Key
4 Connector

The printer operates on a built-in rechargeable Ni-Cad battery. The battery is charged by plugging the charger into an AC outlet.

The PC-4 printer uses electro-thermal roll paper (38mmW × 16mm0) for its special electro-thermal (non-impact) printing system. The paper is easily loaded into the printer by feeding the leading edge of the paper roll into the feed slot and pressing the FEED key to advance the paper.

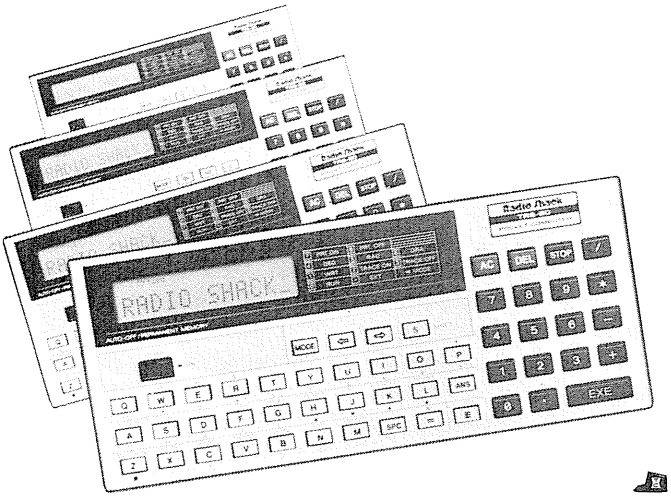
The printer can be connected directly to the computer by placing the computer in the spacer which comes with the printer and connecting the printer to the computer or the printer can be connected to the cassette interface which is in turn connected to the printer.

There are 20 character print positions per printed line and the printing speed is one line per second. The characters are formed in a 5 × 7 dot matrix.

THE PC-4 CASSETTE INTERFACE

With the additions of the PC-4 cassette interface and a compatible cassette tape recorder, the PC-4 has the ability to save and load programs and data to tape.

The PC-4 cassette interface uses two AA size batteries for remote control.



Tape DEBUG HINT

Scott Goehring
5835 East 19th Street
Indianapolis, IN 46218

Although I have found Tape Debug to be a useful program, it has an error in it. The Q command will often cause a BASIC OM error or a system reboot. The following procedure will correct the problem.

1. Load in Debug as shown on the top of Page 2.
2. Type in the following sequence exactly and ignore the machine's responses: M 47CD (space) CC (space) 06 ⟨ENTER⟩ ⟨ENTER⟩
3. Record the corrected copy as shown on the bottom of Page 2.

NOTE: Page numbers refer to the Debug manual. Also, this patch is only for Level II or Model III BASIC versions.

I have a Model I Level II computer with 16K of RAM, although we intend to buy a Model III disk system in April and a LP 8 in July. So far, I program in BASIC, Assembly, and PASCAL. I am 13 years old, and get a little upset when I see 12 year-olds write in about how they know 10 commands.

Network 3 *(From Page 13)*

By typing a number, the student indicates which of the choices he or she prefers. Once a choice has been made, the computer finds and begins the chosen program without further input by the student. The chosen program takes over the control of the computer until the program ends. If the teacher has modified the program's end statement to run the menu program, then the student will return to the list of choices without having to do anything that requires use of computer commands or a knowledge of BASIC, and without having any other special knowledge (like the name of the menu).

If there are more than nine programs in your software library for any particular group of students, the menu utility can still work for you. First, divide your programs into logical groups, or if there are no natural groupings, list them in the order you like. If you have groupings, give each group of programs a name like Science, Math, and English. Next, create three menus named Science, Math, and English, listing each program in the appropriate menu. Last, we create another menu with 3 choices: Science, Math, and English. This main menu gives the student the opportunity to choose first by category, and the next menu gives a specific choice.

Main Menu
1. Science
2. Math
3. English

Science	Math	English
1. Weather	1. Addition	1. Spelling
2. Erosion	2. Subtraction	2. Vocabulary
3. Planets	3. Sets	3. Sentences
4. Volcanoes	4. Fractions	4. Story #1
5. Another Subject	5. Mixed Problems	5. Another Subject
	6. Another Subject	

The last item in each of the subject menus allows the student to return to the main menu if he changes his mind about the subject he wants to pursue. By cascading menus in this way, any number of programs can be included in the list of options. If your programs do not group naturally, begin listing them in order, and at number 9 of the first menu, type 'More Programs' and give the filename of the next menu you will create. Do this as many times as necessary and as the last item of the last menu type 'More Programs' and give the filename of the first menu you created. Your menus now run in a circle which the student can follow to see all of the choices.

Another interesting way to use menus allows you to reward good performance in certain programs by access to other programs. This works especially well in programs produced by you or your students since they are easy to modify. After the student has achieved a certain level of performance in a certain program, a branch can be inserted which sends the student to one menu for a high score and to another menu for a lower one. These menus can list "rewards" activities or can be menus that send the student to more or less difficult activities.

The Menu-Maker utility provides convenience and flexibility for teachers using a Radio Shack Network 3 system. For more information on how you can use the Network 3 Controller and the Network 3 Operating Software, contact your local Radio Shack store or Computer Center, or contact the Radio Shack Educational Sales Coordinator for your region.

PC-2 Assembly Language

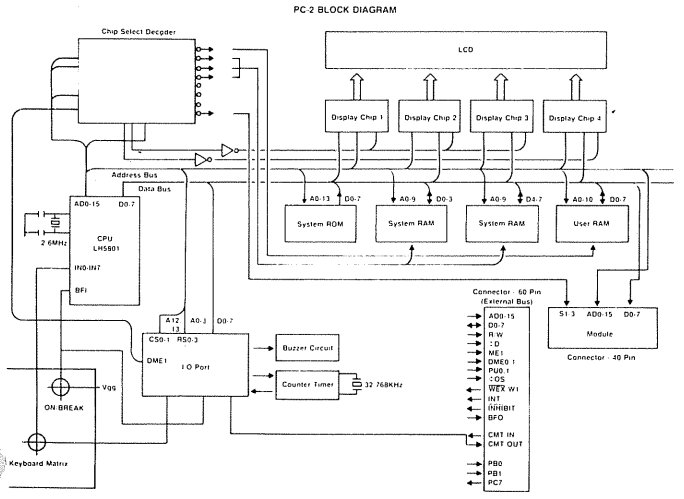
Article by Bruce Elliott

This is the first in a series of articles which will describe the MPU (microprocessor unit) used in the Radio Shack PC-2 pocket computer. It is our intention to include specific information about the 8-bit CMOS microprocessor, the machine code used by the microprocessor, as well as information about the PC-2 memory map and certain ROM calls which are available. Please realize that much of what we are talking about refers to the overall capabilities of the MPU and does not imply that all of these things can be done with a PC-2. Some known precautions when working with the PC-2 include:

- Po—This signal is not supplied to an external output pin on the PC-2.
- TI—The Timer Interrupt service routine is not available on the PC-2. If a Timer Interrupt occurs, an RTI is immediately executed.
- NMI—The Non-Maskable Interrupt is not available to the programmer on the PC-2.
- The MPU signals BRQ and BAK are not supplied to the external output pins.
- Though ME0 is available as an output from the MPU, DME0 (from one of the support chips) performs a similar function and should be used.

Please understand that the information provided in these articles is the only information which is available. We will try to clarify any ambiguities which occur in the articles, but can not reply to questions outside the scope of these articles. Further, published copies of *TRS-80 Microcomputer News* are the only source of this information, and we will not be maintaining back-issues.

PC-2 BLOCK DIAGRAM



OUTLINE OF THE 8-BIT CMOS MPU

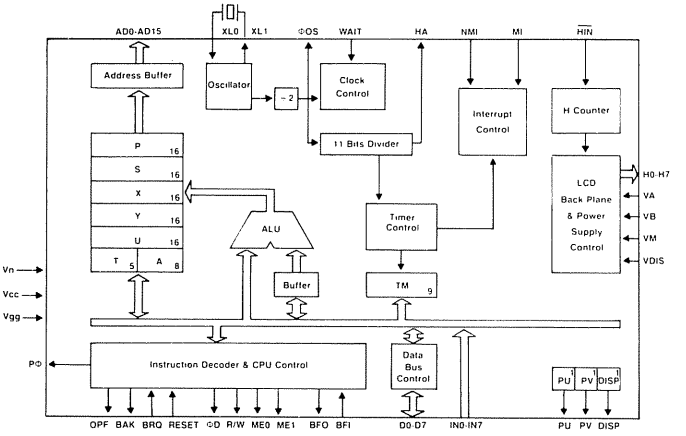
The 8-bit MPU chip (LH5801) uses CMOS static technology. This gives the MPU the low power dissipation inherent to CMOS technology. The MPU incorporates the LCD backplane signal generator, input port, external latch clock and the timer.

The MPU features:

- 16 bit address bus
- 8 bit data bus
- 8 bit input port
- DMA and multiprocessor capabilities
- Contains a WAIT function for memory access control
- LCD backplane control
- Clock frequency of 2.6 MHz.
  a. Internal machine cycle of 1.3MHz.
  b. Minimum instruction execution time of 1.3 microseconds.

In the PC-2, the MPU performs the following functions:

- Key input routine
- Acknowledges remaining program lines
- Interprets program execution statements
- Interprets cassette control statements
- Interprets printer control statements
- Interprets command statements
- Display processing routine
- Arithmetic routines
- Print routine
- Instructs I/O chip to perform serial communications, sound buzzer, and control counter/timer



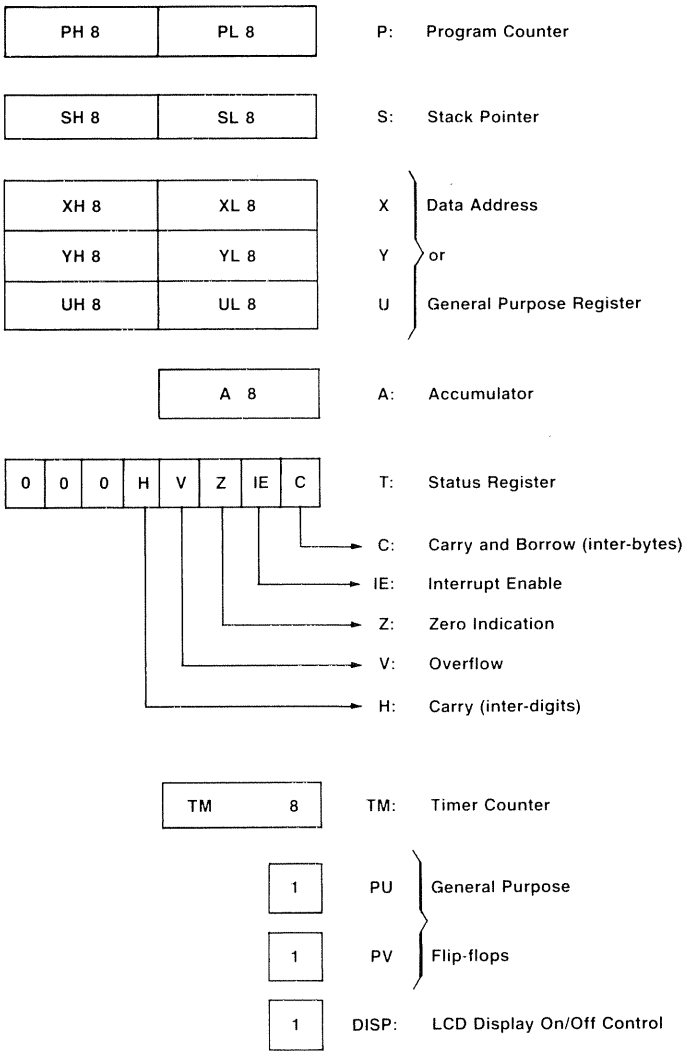
8 Bit CMOS MPU Block Diagram

MPU SIGNALS

ΦD—Output disable signal, when this signal is active, the data bus is in the output mode.

ΦOS—This clock signal is in phase with the internal basic clock and is supplied to the outside system. 2MHz of the clock frequency is supplied when a 4MHz crystal is being used between XL0 and XL1. Since PC-2 uses a 2.6MHz chip, the clock frequency is 1.3MHz.

AD0-AD15—Address bus. The address bus is tri-state and goes into the high impedance state when a Bus Request, BRQ, is issued.



MPU Internal Registers and Flip-flops

BAK—The BAK output is synchronized with the internal clock. When BAK goes high, the Address Bus, Data Bus, ME0, ME1, R/W, and ΦD all turn to the high impedance state. Not used in PC-2.

BFO, BFI—BFO is an output of the BF flip-flop and BFI is an input to the BF flip-flop. The BF flip-flop is normally used for the memory backup system. In the PC-2, BFI is connected to the ⟨**BREAK**⟩ key, and goes "high" when the ⟨**BREAK**⟩ key is depressed. BFO, in the PC-2, is connected to the Chip Select Circuit and the Expansion Port.

BRQ—Bus Request. The MPU responds to the BRQ by turning BAK (Bus Acknowledge) high. Not used in PC-2. Tied to GND.

D0-D7—Bidirectional data bus through which data is written to or read from external memory.

DISP—A flip-flop which is used to control the on and off action of the L.C.D. Instructions are provided to set and reset this flip-flop.

GND—Ground

H0-H7—These are the LCD backplane signals.

HA—Output of the MPU internal driver. Divider output of 625 Hz in the PC-2. Used by the display chips.

HIN—LCD backplane signal and an input to the counter that generates H0-H7. This is connected to HA in the PC-2.

IN0-IN7—This is the input port which the MPU uses to bring 8-bit data into the internal accumulator. Internal pull-up resistance is present. In the PC-2, the input port is connected to the keyboard.

ME0, ME1—The Memory Enable signals used by the MPU to directly access a maximum of 128K bytes in external memory. In the PC-2, ME0 is connected to the chip select circuit and to the ME1 input of the I/O chip. In the PC-2, ME1 is connected to the ME0 input of the I/O chip and the expansion port.

MI—The Maskable Interrupt Input signal. The MPU will respond to this interrupt request when the Interrupt Enable flag (IE) is on. Interrupt processing will begin at the address indicated by FFF8 and FFF9. In the PC-2 this is connected to the INT output of the I/O Chip.

NMI—The Non-Maskable Interrupt Input. The MPU will respond unconditionally, and interrupt processing will begin at the address indicated by the contents of FFFC and FFFD. Not used in the PC-2, tied to GND.

OPF—Operation Code Fetch. Allows the MPU to fetch an operation (instruction) code. OPF appears when an instruction code is fetched, during address data and immediate data operations, and when the second byte of a two step instruction is being fetched. Not used in the PC-2.

PΦ—External latch clock. The contents of the accumulator is transferred on the data bus when this clock is in the high state, and can be used as an output port when an external latch IC is present. Not used in the PC-2.

PU, PV—These are MPU internal flip-flops. Set and reset instructions are provided for both PU and PV. In the PC-2, both PU and PV are connected to the expansion port. PU is one of the enable signals for the printer ROM.

R/W—Memory Read/Write Signal.

RESET—MPU reset input which causes the MPU to reset when a high signal is received. Program execution begins at the memory address pointed to by the contents of FFFE (low order) and FFFF (high order.) Execution begins at the indicated address when the RESET input changes from a high to a low state. On the PC-2 this is connected to the All Reset Switch.

VA—Power Supply to the LCD. High voltage for segment signals, 1.2—2.2 volts.

VB—Power Supply to the LCD. Low voltage for segment signals, .2—1.2 volts.

Vcc—+4.7 volts

VDIS—Power Supply to the LCD. +3.7 volts.

Vgg—+4.7 volts

VM—Power Supply to the LCD. An intermediate voltage used for the common and segment signals. .8—1.6 volts.

WAIT—When the MPU receives a high signal at the WAIT input, the MPU internal clock is halted to stop microprogram execution inside the MPU. WA is an internal flip-flop which accepts the WAIT input at the falling edge of the clock oOS and stops the MPU clock when it is in a high state. Connected to the WAIT output of the I/O chip in the PC-2. This informs the CPU when memory or an I/O device is not ready.

XL0, XL1—Crystal connection pins. PC-2 uses a 2.6MHz crystal which operates the MPU at a 1.3MHz clock frequency. XL0—Input, XL1—Output

MPU DESIGNATIONS

A : "A" represents the 8-bit register (accumulator) used for retention of arithmetical results or for data transfer with external (non-MPU) memory.

DISP: LCD display on/off control

P : "P" represents the 16-bit register (program counter) that indicates the next address that follows the currently executing instruction, and is automatically incremented by one when the next instruction is fetched. The maximum 64K bytes addressed by ME0 is addressable by P and constitutes the program area.

PH: High order 8 bits of the program counter

PL: Low order 8 bits of the program counter

PU: General purpose flip-flop

PV: General purpose flip-flop

R : represents any one of the X, Y, or U 16-bit registers. These registers can also be used as data pointers. When X, Y, or U are used as data pointers, it becomes possible to issue Memory Enable signals, ME0 and ME1, independently. A maximum of 128K bytes of memory area is available to X, Y, and U (a maximum of 64K bytes in the memory area accessed by ME0 and another 64K bytes in the memory area accessed by ME1.)

RH: represents any one of the high order XH, YH, or UH 8-bit registers

RL: represents any one of the low order XL, YL, or UL 8-bit registers.

S : "S" represents the 16-bit register (stack pointer) that indicates the next available stack address for the push-down or pop-up stack in memory. The maximum 64K bytes addressed by ME0 is available as the stack area.

SH: High order 8 bits of the stack pointer

SL: Low order 8 bits of the stack pointer

T : "T" represents the 5-bit register (status register or flags) designed to hold status information such as: carry (C), borrow (H), zero (Z), overflow (V), and interrupt enable (IE). The flags (C, H, Z, V), other than the interrupt enable, can be tested by the conditional branch or conditional subroutine jump instructions.

TM: "TM" is the 9-bit polynomial counter (timer counter.)

U : 16-bit register

UH: High order 8 bits of register U

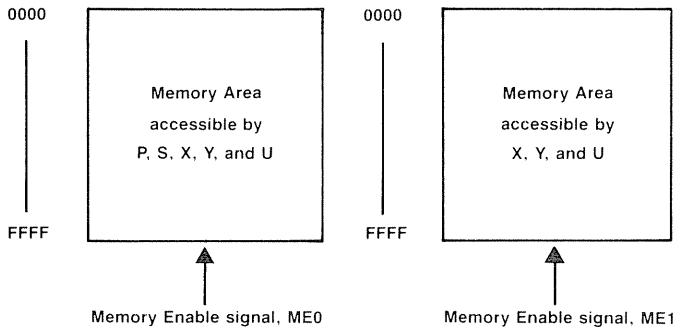
UL: Low order 8 bits of register U
X : 16-bit register
XH: High order 8 bits of register X.
XL: Low order 8 bits of register X.
Y : 16-bit register
YH: High order 8 bits of register Y
YL: Low order 8 bits of register Y

OPERATIONAL SYMBOLS

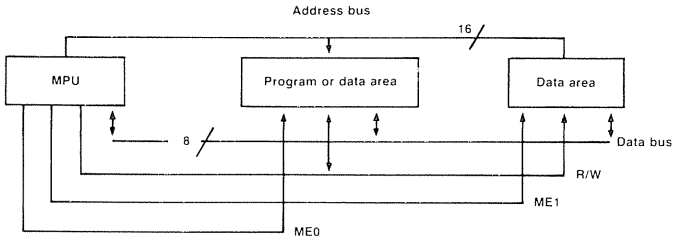
→ : Signal or data flow
← : Signal or data flow
• : Logical AND
v : Logical OR
⊕ : Exclusive OR
+ : Arithmetic addition
– : Arithmetic subtraction

MEMORY AND ADDRESS REPRESENTATION

Since the Memory Enable signals, ME0 and ME1, are output from the MPU, the PC-2 microprocessor can directly access any area within 128K bytes. ME0 takes care of one 64K byte memory area and ME1 another 64K byte memory area. However, ME0 is dedicated to program or data areas and ME1 to data area only.



Memory Area accessible by MPU

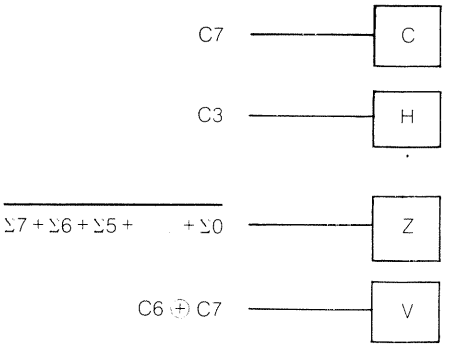


(R) : The contents of the ME0 accessible memory that can be specified by the register R.
#(R): The contents of the ME1 accessible memory that can be specified by the register R.
(ab): "a" is a number that represents the high order 8 bits of the address and "b" low order 8 bits of the address. Together, they indicate the contents of the memory that can be represented by the 16 combined bits of a and b (ME0 accessible).
#(ab): Same as the above, except that it can be accessed by ME1.
ab : used in defining the conditional jumps and subroutine calls to designate the two hex digits which comprise a single byte immediate value "i".

STATUS FLAGS

The status flags, C, V, H, Z, and IE are contained in the 5-bit status register. The contents of C, V, H, and Z may change upon completion of an arithmetic instruction.

Assume that the added results of each bit of the 9-bit full adder are as follows:
Σ7, Σ6, Σ5, Σ4, Σ3, Σ2, Σ1, Σ0, with carry of C7, C6, C5, C4, C3, C2, C1, C0.
The input conditions for each of flags shall be as described below:



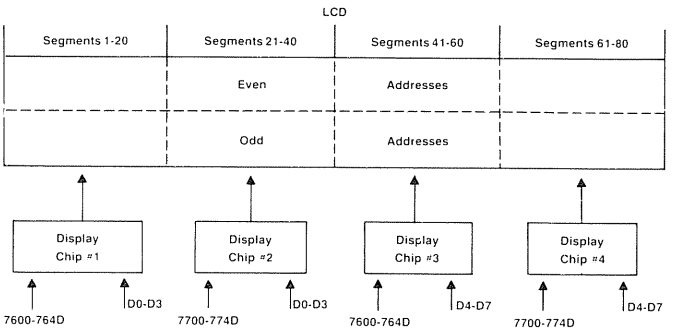
(1) Carry flag C—The carry flag C is either set or reset depending on the presence of a carry in C7 (8th bit).
(2) Half carry flag H—The half carry flag H is either set or reset depending on the presence of a carry in C3.
(3) Zero flag Z—The zero flag Z is dependent on the arithmetic results, it will be set when the result is zero, otherwise, it will be reset.
(4) Overflow flag V—The overflow flag V is set when the arithmetic results of one byte is in overflow, provided that the 8th bit is used for a sign with rest of the 7 bits for used for numeric representation.

I/O PORT CHIP

Contains:
- two 8 bit bi-directional ports, labeled PA and PB. Each bit in these two ports can be programmed as either an input or an output. The CPU can access PA or PB as one location in memory. PA is used for the keyboard strobe and PB is used for cassette, counter/timer, and as an interrupt input.
- one 8 bit output port labeled PC. PC can be accessed as one location in memory and is used for counter/timer control and to sound the buzzer.
- Two interrupt request inputs, used with ⟨BREAK⟩ and IRQ inputs from the expansion port.
- one interrupt request output connected to the CPU.
- CPU WAIT control output. Outputs two memory enable signals, DME0 and DME1, which are used with memories that have slow access times.
- Controls serial communications. The two wait input lines, W0 and W1, are used in serial communications.

LCD DISPLAY CHIPS

Four display chips used for displaying information on the LCD, and as memory space for fixed memories E$ - Z$. Display chips 1 and 3 are used for the LCD display, indicators, and fixed memories E$ - O$. Display chips 2 and 4 are used for the LCD display and for fixed memories P$ - Z$.



OTHER PARTS OF THE PC-2 SYSTEM

- Chip Select Decoder Circuit
- 16K System ROM
- 1K System RAM (two 5514 RAM chips). This RAM is used for fixed memories A$ - D$, fixed memories A - Z, stack space, the 80 character input buffer, and is used by FOR-NEXT statements.
- 2K User RAM (one 6116 RAM chip). This RAM is used for fixed memories A27 or A$27 and above as well as being used for Reserve, Program and Variable memory.
- Buzzer circuit
- Counter/Timer circuit
- Module port
- Expansion port
- Keyboard

Memory Map:

0000 - 3FFF Module ROM - 16K

4000 - 47FF User RAM - 2K

- 4000 - 4007 Reserve Memory pointers
- 4008 - 4021 Menu 1
- 4022 - 403B Menu 2
- 403C - 4055 Menu 3
- 4056 - 40C3 Function Key Definitions
- 40C4 0 to mark end of function key definitions
- 40C5 - 47FF Program (Variable) Memory

4800 - 6FFF Module RAM

7000 - 75FF Duplicate of 7600 - 7BFF

7600 - 76FF Display Chip 1 & 3

- 7600 - 764D LCD Display - Sections 1 & 3
- 764E Indicator
  - Bit 0 - Busy
  - Bit 1 - Shift
  - Bit 2 - Japanese
  - Bit 3 - Small
  - Bit 4 - III
  - Bit 5 - II
  - Bit 6 - I
  - Bit 7 - Def
- 764F Indicator
  - Bit 0 - De
  - Bit 1 - G
  - Bit 2 - Rad
  - Bit 3 -
  - Bit 4 - Reserve
  - Bit 5 - Pro
  - Bit 6 - Run
  - Bit 7 -
- 7650 - 765F E$
- 7660 - 766F F$
- 7670 - 767F G$
- 7680 - 768F H$
- 7690 - 769F I$
- 76A0 - 76AF J$
- 76B0 - 76BF K$
- 76C0 - 76CF L$
- 76D0 - 76DF M$
- 76E0 - 76EF N$
- 76F0 - 76FF O$

7700 - 77FF Display Chips 2 & 4

- 7700 - 774D LCD Display - Sections 2 & 4
- 774E - 774F Not used
- 7750 - 775F P$
- 7760 - 776F Q$
- 7770 - 777F R$
- 7780 - 778F S$
- 7790 - 779F T$
- 77A0 - 77AF U$
- 77B0 - 77BF V$
- 77C0 - 77CF W$
- 77D0 - 77DF X$
- 77E0 - 77EF Y$
- 77F0 - 77FF Z$

7800 - 7BFF System Memory - 1K

- 7800 - 78BF System Memory - 192 Bytes
- 7863 RAM top - High order 8 bits
- 7864 RAM bottom - High order 8 bits
- 7865 - 7866 Beginning of BASIC program
- 7867 - 7868 End of BASIC program
- 7869 - 786A Head address of a BASIC program to perform editing based on keyboard entries.
- 786B Beep On/Beep off
- 7875 LCD Cursor Position
- 7879 Cassette parameter F/F
- 7880 LCD display parameter F/F
- 7890 - 7893 Used by RIGHT$, LEFT$, MID$
- 7894 String Buffer Pointer 7894 = 10H
- 7899 - 789A Start of variable storage area
- 789B Error Code = ERR/2 + 1
- 78C0 - 78CF A$
- 78D0 - 78DF B$
- 78E0 - 78EF C$
- 78F0 - 78FF D$
- 7900 - 7907 A
- 7908 - 790F B
- 7910 - 7917 C
- 7918 - 791F D
- 7920 - 7927 E
- 7928 - 792F F
- 7930 - 7937 G
- 7938 - 793F H
- 7940 - 7947 I
- 7948 - 794F J
- 7950 - 7957 K
- 7958 - 795F L
- 7960 - 7967 M
- 7968 - 796F N
- 7970 - 7977 O
- 7978 - 797F P
- 7980 - 7987 Q
- 7988 - 798F R
- 7990 - 7997 S
- 7998 - 799F T
- 79A0 - 79A7 U
- 79A8 - 79AF V
- 79B0 - 79B7 W
- 79B8 - 79BF X
- 79C0 - 79C7 Y
- 79C8 - 79CF Z

79D0 - 7BFF System Memory - 560 Bytes

- 79E0 - 79E1 Printer X-axis position relative to origin
- 79E2 - 79E3 Printer Y-axis position relative to origin
- 79E4 - 79E5
- 79E6 Printer HCURSOR value
- 79E7 - 79E8
- 79E9 Printer pen up/down
- 79EA Printer line type
- 79EB - 79EF
- 79F0 Printer Text/Graphic mode
- 79F1
- 79F2 Printer ROTATE value
- 79F3 Printer pen color
- 79F4 Printer CSIZE
- 7A00 - 7A07 Numeric Data Buffer or String pointer
- 7A10 - 7A17 Numeric Data Buffer or String pointer
- 7B10 - 7B4F String Buffer
- 7B60 - 7B67 Tape out Synchronization header
- 7B68 - Tape out file mode
- 7B69 - 7B78 Tape out file name
- 7B79 - 7B84 Tape out header (available to user)
- 7B85 - 7B86 Tape out # bytes in BASIC file -1
- 7B87 - 7B88 Tape out end header
- 7B91 - 7BA0 Tape in file name
- 7BA1 - 7BAB Tape in user header
- 7BAC - 7BAD Tape in # bytes in BASIC file -1
- 7BAE - 7BAF Tape in end header
- 7BB0 - 7BFF 80 Character Display Buffer

7C00 - 7FFF Duplicate of 7800 - 7BFF

8000 - BFFF Expansion ROM - 16K

- A519 Change printer pen color
- A769 Printer motor off
- A781 Send ASCII character to printer (no LF)
- A8DD Move pen
- A9F1 Send line feed (LF) to printer
- AA04 Send n line feeds to printer
- AAD9 Pen Up/Down
- ABCB Switch printer from graphic to text mode
- ABEF Switch printer from text to graphic mode
- BBD6 Write tape synchronization header
- BBF5 Finalization of tape I/O control
- BCE8 Read tape synchronization header/search for filename
- BD3C Read/Write file to tape
- BDCC Send a character to tape
- BDF0 Read a character from tape
- BF11 Turn tape drive on
- BF43 Turn tape drive off

C000 - FFFF System Program ROM - 16K

- D0D2 Magnitude Comparison for Numeric Values
- D0F9 Magnitude Comparison for Character Strings
- D2EA Search for program line number
- D461 Find address of variable
- D925 String concatenation
- D9B1 CHR$
- D9CF STR$
- D9D7 VAL
- D9DD ASC if YL = 60H, LEN if YL = 64H
- D9F3 RIGHT$, LEFT$, MID$
- E243 Keyboard Scan - wait for character
- E33F Auto Power Off
- E42C Keyboard Scan - no wait
- E8CA Display contents of display buffer
- ED00 Output n characters to LCD using current cursor location
- ED3B Output n characters to LCD beginning at cursor = 0
- ED4D Output one char to LCD and increment cursor position by one
- ED57 Output one character to LCD
- ED95 Convert two bytes of ASCII code (0-9,A-F) into one byte of hex data
- EDEF Output one graphic column to current cursor position
- EFB6 X - Y → X
- EFBA X + Y → X
- F00B I/OP Flag 2
- F01A X * Y → X
- F084 X / Y → X
- F0E9 SQR X → X
- F161 LN X → X
- F165 LOG X → X
- F1CB EXP X → X
- F1D4 10 ^ X → X
- F391 COS X → X
- F39E TAN X → X
- F3A2 SIN X → X
- F492 ACS X → X
- F496 ATN X → X
- F49A ASN X → X
- F531 DEG X → X
- F564 DMS X → X
- F597 ABS X → X
- F59D SGN X → X
- F5BE INT X → X
- F89C Exponentiation (X ^ Y → X)

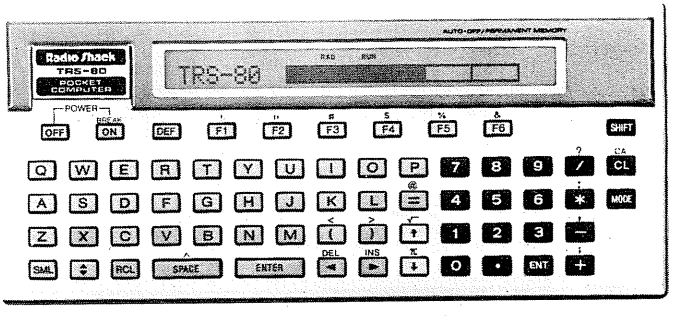
FF00 - FFF6 Vectors for jumps and calls

FFF8 - FFF9 Start Address for MI routine

FFFA - FFFB Start Address for the Internal Timer

FFFC - FFFD Start Address for the NMI routine

FFFE - FFFF Start address for the RESET routine



PC-2/PC-1 Program Adaptation and Incompatibilities

by Peter Levy

This article will explain what must be done to convert an existing PC-1 program to run on a PC-2. Of course, the adapted program won't work right on the PC-2 unless it already works correctly on the PC-1.

The PC-2 implementation of BASIC includes all of PC-1 BASIC. Every operation possible on the PC-1 can be done with the PC-2. In fact, in many cases the PC-2 will run a PC-1 program exactly as-is. This article will deal with the cases where it won't, and will explain the differences between the two pocket computers and how to program around them.

Just to start off with, a PC-2 cannot read a PC-1 tape, nor can a PC-1 read a PC-2 tape.

ARRAYS

The single largest difference between the two pocket computers is array handling. The PC-1's array handling is limited to its use of A(i) to address specific variables—A(1) is A, A(4) is D, and so on. The PC-1 can also use A(27) and up to address extra (unused) memory as additional variables.

The PC-2 uses "real" arrays, complete with DIMension statements so, in PC-2 BASIC, A(2) would refer to the third (counting from zero) element of a dimensioned array A—not to variable B. For this reason, to address A through Z (or A$ through Z$) as array elements, the PC-2 uses @(i) where the PC-1 would use A(i).

Use of @(i) in the PC-2 is limited to i from 1 to 26, inclusive. You can't use @(27) and up to reach unused memory in the PC-2, since its variable management is very different from the PC-1.

So, where a PC-1 program refers to A(i), substitute @(i) if i will always be from 1 to 26. If i may exceed 26, dimension an array A (or A$) of the needed size. The second way is easier, but uses a lot of extra memory which is usually at a premium in a PC application.

CLOAD1 AND MERGE

The PC-1 can load additional program(s) into memory from tape with the CLOAD1 statement. The PC-2 uses the different word MERGE for the same purpose. Any CLOAD1 statements in PC-1 programs must be changed to MERGE to run on a PC-2.

DEBUG AND TRON/TROFF

The PC-1 enters its debugging mode upon receipt of the direct command DEBUG, which is treated much like RUN. The PC-2 uses the command TRON for the same purpose.

Where a PC-1 program would be debugged by typing DEBUG, the PC-2 accomplishes the same thing with consecutive commands TRON and RUN (or GOTO or a DEF-key). TRON sets the debugging state as "on," TROFF sets it to "off." These changes allow a PC-2 to start and stop the debugging process in mid-run.

FOR/NEXT LOOPS

The PC-1's internal definition of when to close a FOR-NEXT loop is slightly different from the industry standard for BASIC. The PC-1 executes a FOR-NEXT loop until after the counter variable is equal to or greater than the end (TO) value and exits the loop AFTER the first iteration with the counter equal to or higher than the upper counter limit.

The PC-2 conforms to the BASIC convention and stops a FOR-NEXT loop BEFORE the first iteration in which the counter variable is greater than (not equal to) the TO value.

The effect here is that if the STEP increment doesn't divide into the difference between the FOR and TO values evenly, a PC-1 will do the FOR-NEXT sequence one more time than a PC-2.

The best way around this is to use good programming practices in designing FOR-NEXT loops—use a STEP which divides into the loop range (FOR minus TO) evenly.

If you have a special case in which you just have to use an odd STEP increment, just add enough to the TO limit to ensure the loop is executed an extra time.

IF AND THEN

The PC-1's BASIC differs again from the common BASIC convention in the area of IF-THEN evaluation. The PC-1 treats any positive number as logically TRUE, and zero or negative numbers as FALSE.

The PC-2 conforms to the BASIC convention, in which zero is FALSE and all nonzero numbers are TRUE.

The effect here is that the PC-1 treats a negative number as FALSE, the PC-2 as TRUE. This can cause trouble in a program which uses IF to branch on a value and that value is not compared to anything in the IF evaluation.

For example, if variable X equals −1 and the PC-1 executes the statement IF X THEN 100, it will not branch because −1 is FALSE to the PC-1. The PC-2, executing the same statement, would branch to line 100 because −1 is TRUE to the PC-2.

The best way around this is to avoid using IF without a numeric comparison. Where the PC-1 used IF X THEN, use IF X>0 THEN on the PC-2.

PRINT AND LPRINT

The only way to make the PC-1 print output to a printer is to reset the computer with the printer active. The PC-2 has no

equivalent to this; the only way to make a PC-2 print (rather than display) results is to substitute LPRINT for PRINT.

SUMMARY OF COMPATIBILITY PROBLEMS AND CURES

Problem	Likely Cause/Cure
ERROR 1	(a) Untranslated CLOAD1 statement. Change to MERGE. (b) Tried to change PRINT to LPRINT by insertion of a leading "L". Retype entire word as LPRINT.
ERROR 6	(a) Untranslated A(i). Change to @(i). (b) Didn't DIMension array A. DIM A(x) at start of program.
ERROR 9	(a) Changed A(i) to @(i), but i exceeded 26. Change back to A(i) and DIMension A at start of program.
ERROR 20	(a) Typo in translation of A(i) to @(i).
FOR-NEXT loop stops early	(a) STEP doesn't divide into FOR-TO evenly. Adjust STEP if possible, otherwise alter TO as necessary.
IF branches unexpectedly	(a) Evaluated a negative expression as TRUE which is FALSE to a PC-1. Translate IF X to IF X>0.
No printer output	(a) Untranslated PRINT. Change to LPRINT.

Installing TRS-XENIX *(From Page 10)*

If you have been following these instructions you will answer "y" for yes to this prompt as well. You will again be asked for the number of tracks and heads (but not the Media Error Map information). The hdinit program will now continue automatically to initialize the hard disk. Do not touch the system until the message

"*NORMAL SYSTEM SHUTDOWN*"

appears on your screen.

In the final phase of the installation you will run the program "firsttime". Again, as in the last step, the program will run automatically. To start this phase, reboot the system, this time from the hard disk. At the XENIX prompt ":" press ⟨**ENTER**⟩. This time the message should read

"TRS-XENIX Hard Disk
Basic System
File System Installation"

The firsttime program will give you instructions to insert and remove the remaining "Install Disks" into drive 0 as they are copied to the hard disk.

"For each floppy in the Distribution set, insert the floppy into drive 0, and hit the ⟨**ENTER**⟩ key, or type the letter 'y'. Should you see the message 'tar:please mount volume.', insert the next floppy into drive 0, and hit the ⟨**ENTER**⟩ key. Do not interrupt the installation of the distribution at this point."

At this point you will be prompted with the question:

"First Floppy?"

Answer with "y" followed by ⟨**ENTER**⟩ after you have inserted the floppy disk in drive 0 and the system will tell you that it is:

"Extracting files from floppy . . . "

When the process is completed, you will be prompted again with:

"Next floppy?"

Once again, answer "y" and the system will continue to copy files to the hard disk. When all of your "Install Disks" have been copied to the hard drive, answer "n" when it asks for the next floppy. You will then see the message:

"Setting up directories and permissions.
Installation Complete."

The installation of the TRS-XENIX is now complete. The next message will be:

"Type control-d to proceed with normal startup (or give root password for system maintenance):"

This is the normal boot you will use every time you start your system. At this point type "control-d". The system will ask you for the date and time. You may press ⟨**ENTER**⟩, but we recommend that you enter the date and time as they will be used on all reports generated by the system as well as on all file creation dates. (The system will only be as accurate as you make it.) After the date and time are entered you will see:

"login:"

Since you are using the system for the first time, answer with "root" and press ⟨**ENTER**⟩. XENIX will welcome you to the system and return with the XENIX prompt which is a "#" for the "root."

Now that the TRS-XENIX operating system is installed on the hard disk, the next step is to learn about the system. At this point we suggest (if you have not already done so) that you thoroughly read the "TRS-XENIX Operations Guide" to familiarize yourself with the multi-user operating system. One of the first things you will need to learn is how to properly shut down the system.

Currently we are logged in as the "root" and our prompt appears as a "#". At the prompt, type:

/etc/haltsys ⟨**ENTER**⟩

you will see the message:

****NORMAL SYSTEM SHUTDOWN****

When you have familiarized yourself with the TRS-XENIX Multi-User Operating System, you will then be ready to add user accounts to the system and begin using the system for applications software. Many of the Radio Shack accounting software packages are already available for installation on the multi-user system. Of course your local Radio Shack Computer Center will have a complete, up-to-date list of the software available.

We have tried to make the installation of the TRS-XENIX as simple as possible, and we in Computer Customer Service hope that your installation has gone well. The TRS-XENIX Multi-User Operating System is a very powerful system and we hope that you enjoy using it.

Transforming FORTRAN Files from IBM Format to TRSDOS Model II Format

George Greenwald
12122 Western Ave.
Garden Grove, CA 92645

I wanted to convert several FORTRAN source codes on an IBM System 34 computer for use on a TRS-80 Model II. The process seemed at first to be easy enough. Our company purchased a copy of Radio Shack's "ReformaTTer™," and I went through the conversion procedure outlined in that manual. After using ReformaTTer, the IBM 80 character EBCDIC files were converted to 80 byte ASCII format, but the FORTRAN source codes were still not in a form that could be used by the Model II FORTRAN; the Model II expects to see single byte records packed end-to-end into the TRS-80, 256 byte records. Model II FORTRAN lines begin with a five digit line number, terminated by a tab character. The text follows, with the end of the line marked by a carriage return character. The conversion was still missing the line numbers, the tabs, and the carriage returns at the end of the lines. Though not a serious problem, the new code also had many individual spaces where the Model II FORTRAN would normally use tabs, thus occupying more storage and memory than necessary.

I developed a BASIC program to complete the process of converting the reformatted file to one byte records and to add to the required line numbers, carriage returns, and tabs where appropriate. That program is shown below. I have used this program successfully to convert several programs from IBM to TRSDOS.

Some words of caution and specific changes that the user will need to be aware of:

- In the ReformaTTer program, I have had the best results using the "A" option, which formats each record identically the way the IBM file was formatted.
- In the program below, the user will need to insert in lines 60 and 110 the appropriate file names of the original and final programs.
- Line 130 sets R% = 6 because in all of our files, the IBM codes had five control lines that we did not want copied over. That value could be changed to 1, and the unwanted lines eliminated later by use of the FORTRAN editor.
- In the event that the FORTRAN program has regions that have more than 8 spaces imbedded in any line, a minor modification is needed in line 460, since the function of that line is to determine when the end of a FORTRAN line occurs and terminate the line. The only consequence of either eliminating the line 460 or using a number larger than 8 is that additional spaces would be imbedded in the source code.
- Additional changes to the converted FORTRAN code may be required before compilation using the FORTRAN editor; there may be certain commands which are not available in the TRSDOS Model II FORTRAN but were originally available on the IBM. For example, free-formatted input from the keyboard is not allowed on the Model II.

If you have an interest in making an IBM to TRSDOS FORTRAN conversion, I think you will find that with a few minor adjustments, the program below will do the job for you.

*ReformaTTer/TM MicroTech Exports

```
10 '** Program TRANFOR/BAS Converts IBM FORTRAN Code
    to TRS-80 Model II Format.
20 '** Use on files that have been converted by
    "ReformaTTer."
30 'Created by GEORGE F. GREENWALD   6/03/82
40 DIM D$(80), E$(80)
50 'OPEN FILES
60 OPEN "D",1,"USER INPUT FILE NAME",80 ' NOTE THAT
    CHANGE BY USER IS REQUIRED
70 FIELD 1, 1 AS C$, 79 AS D$(80)
80 FOR L% = 1 TO 79
90      FIELD 1, L% AS D$(L%), 1 AS E$(L%)
100 NEXT L%
110 OPEN "D",2,"USER OUTPUT FILE NAME",1 ' NOTE THAT
    CHANGE BY USER IS REQUIRED
120 FIELD 2, 1 AS F$
130 RE% = 0
    : J%(1) = 0
    : J%(2) = 0
    : J%(3) = 1
    : J%(4) = 0
    : J%(5) = 0
    : R% = 6
140 IF R% > LOF(1) THEN CLOSE
    : END
150 GET 1,R%
160 'ADD LINE NUMBERS
170 FOR I = 1 TO 5
180     FR$ = STR$(J%(I))
190     FG$ = "A" + FR$
200     FT$ = RIGHT$(FG$,1)
210     GOSUB 560
220 NEXT I
230 'ADD TAB CHARACTER AFTER THE LINE NUMBER
240 FT$ = CHR$(09)
    : GOSUB 560
250 'PRINT LINE NUMBERS ON SCREEN TO VIEW PROGRESS OF
    THE CONVERSION.
260 PRINT J%(1); J%(2); J%(3); J%(4); J%(5)
270 'INCREMENT LINE NUMBER COUNTER
280 IF J%(3) = 9 THEN J%(3) = 0 ELSE J%(3) = J%(3) +
    1
    : GOTO 320
290 IF J%(2) = 9 THEN J%(2) = 0 ELSE J%(2) = J%(2) +
    1
    : GOTO 320
300 J%(1) = J%(1) + 1
310 'CHECK FOR CHARACTER IN COLUMN ONE.
320 IF C$ <> "" THEN FT$ = C$
    : GOSUB 560
    : K = 5 ELSE K = 6
330 'CHECK FOR CHARACTER IN COLUMNS 2 THROUGH 6
340 'IN NONE FOUND, INSERT A TAB.
350 KK = 0
360 FOR I = 1 TO K
370 IF E$(I) = CHR$(32) THEN 380 ELSE I = K + 1
    : KT = 1
    : KK = 1
    : GOTO 410
380 NEXT I
390 FT$ = CHR$(09)
    : GOSUB 560
400 KT = K + 1
```

(Continued on Page 36)

WHAT IS "VIDEOTEX"?

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

Videotex is not missing pieces from your video recorder, a piece of equipment from Texas or a nickname for the nerd next door. Videotex is—generic. Yes, that's right. The truth is out. Videotex is generic just like "Kleenex," "Graham crackers," and "Quick Oats." The true identity of videotex is as a two-way, interactive system that transmits electronic information (text and graphics) directly to you, the mass medium consumers. Videotex users and their computers or computer terminals interact with a main computer by using a key pad, push button console, or full typewriter-like keyboard to transmit signals to the main computer via cable, phone, lines. Information is sent to your computer (your equipment at your home or office) and is displayed by a TV screen or other video display unit.

As in any videotex operation, CompuServe's information is prepared by writers, designers, programmers, and research specialists who gather and format the text and graphics in conjunction with outside information providers. They put it up on the main system which you are then able to access, or in some cases CompuServe will switch the customer to a computer in another location. The efforts of CompuServe's staff allow consumers to search a database, send and receive inquiries, shop or bank at home.

The costs involved in being a user include cost of communications, the telephone charges (mainly for those using long distance), the computer connect time and any extra charges for premium products.

Some people wonder what the difference is between videotex and videotext. Videotex and videotex(t) are one in the same. If you were in Britain, you'd be using the Prestel videotex system operated by the government, postal, and telephone service. In France, Telematique, also operated by the government, postal, and telephone service, would be the service where you'd plug in your computer. And if you were in the Far East you'd be using the Captain system.

So far, we've generalized the generic. We have the user, a computer, and a communications system. Specifically, how does all this mix? With technology.

The technology helping to provide flexible, reliable, and cost effective data communications. And CompuServe is just one videotex service which accomplishes this.

In a dramatization, CompuServe customer Moe Bedink dials the local access number in his area. He is linked to CompuServe via leased telephone lines and satellite circuits. Moe is then able to retrieve the information he's interested in through his personal computer or terminal.

When Moe's computer links with the CompuServe mainframe equipment, it becomes part of a complex, yet well-managed data communications network. The network uses a ring configuration which chooses the best routes for the data between his site and the host computer with almost no geographical or topological limitations.

Built into this system is an automatic alternate routing feature. This, along with an automatic reconnect feature, routes his data to another line should the one he was using become temporarily unavailable. CompuServe's 24-hour staff monitors the network for problems, corrects them and continues to provide uninterrupted service. You enjoy worry-free communication, we enjoy some of today's best technology at work.

The CompuServe network extends across the continental United States and Canada. Customers in cities without local CompuServe numbers use the supplemental networks of Tymnet.

Tymnet is a communications network service comprised of telephone lines. Like CompuServe's network, it uses equipment designed to improve the efficiency of data transmission in order to achieve a logical association between stations and the customer's data terminal equipment.

Telenet is a public network consisting of Telenet access locations around the world from 110 to 56,000 bps.

So, now that Mr. Bedink and his family know the how and whys of receiving their videotex information, they should also know about videotex equipment requirements.

There are many types of computers on the market today and CompuServe is compatible with most all of them. CompuServe provides a list of computer equipment, their requirements, and their compatibility.

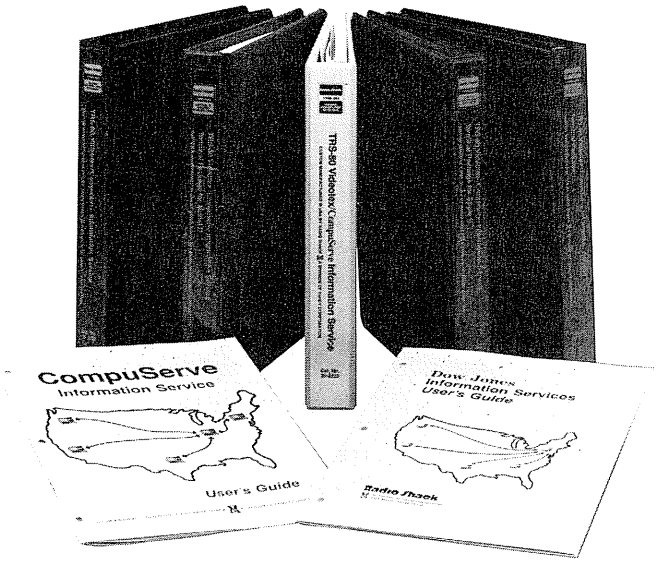
There are some technical requirements to follow in order to communicate with CompuServe. These include setting your system for 110, 300, or 1200 baud; full or half duplex;

one stop bit for 300 and 1200 baud and two stop bits for 110; ASCII code and seven data bits with an even parity bit and eight data bits with no parity bit.

One of the most important components for communicating is a modem. A modem is a device which allows data communications over phone lines. The word is a combination of "modulate" and "demodulate." A modem translates digital signals which won't travel through telephone lines into audio signals which are suitable for telephone transmission (modulate). It also retrieves digital signals from audio signals from the telephone lines (demodulate).

These signals are converted to tones for transmission over phone lines and received by a CompuServe modem. It is then modulated, transmitted back to the customer's modem and demodulated to signals the terminal can read.

Modems come in different categories. Some connect directly to the telephone by wires; some use acoustic couplers to transfer the signals; some communicate only with answering modems and are called originate only (they can't communicate with each other); and some communicate with both originate only and answering modems and are called originate/answer modems.



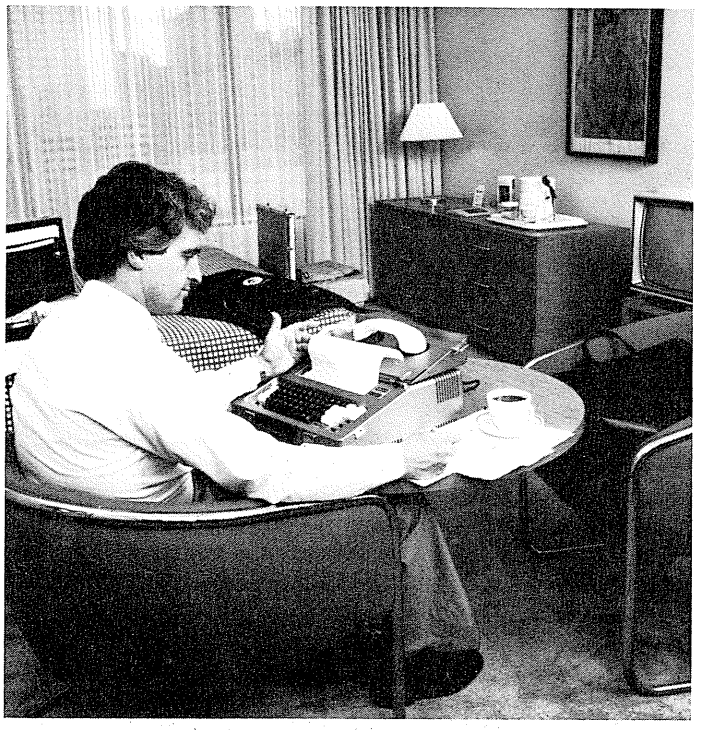
Besides the personal computer and the modems, there's the host computer. CompuServe's mainframes are Digital Equipment Corporation's KL-10 and KL-20 central processors. They are considered by experts the most effective in interfacing with remote terminals. These mainframes incorporate the most cost effective and reliable memory units and disk storage equipment in delivering high performance time-sharing to our customers.

The future of our generic, videotex, is unlimited. Videotex supporters and researchers hope to soon find this new medium taking its place among the home's most useful appliances. The TV, radio, and microwave are commonplace today, and in the future, videotex will be also.

As for CompuServe, we will continue to provide new services and products, focusing on areas such as travel and entertainment. We are also heavily involved in the development of graphics. Today we have limited graphic applications in our business services, and others will follow in our games area.

When will today's favorite generic reach future stages of perfection? It's anybody's guess.

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, or Jacqueline A. Farthing, assistant editor, CompuServe Information Service, 5000 Arlington Centre Boulevard, P. O. Box 20212, Columbus, Ohio 43320 or through Feedback, main menu item 5, CompuServe User Information.



Additional PMODE3 Color Set

Joseph Kohn
1343 Blossom Avenue
Redlands, CA 92373

In the Extended Color BASIC Color Computers, it appears that complete switching of PMODEs requires both the PMODE and SCREEN commands. By using the following program sequence, an additional PMODE3 color set can be accessed.

The color set is:
5 = Black
6 = Blue
7 = Orange
8 = Buff

The program sequence to set up this mode is:

```
10 PMODE4: SCREEN 1, 1
20 PMODE3
```

The only restriction appears to be to not use additional SCREEN commands as this would force a reversion to the standard color set.

PROFILE III+ and Your Printer: A Marriage Made in Heaven

The small Computer Company
P.O. Box 2910
Fort Worth, TX 76113-2910
Profile III+ Section By Ivan Sygoda, Director, Pentacle

If you've been using Profile III+, or at least reading these articles about it over the past few months, you know what an incredible amount of information this data base management system can keep in reserve for you—information about clients, about products, about any of the people or things you deal with in your business and personal life. You've also seen Profile III+'s power to manipulate that information so that you can use it efficiently and effectively.

It's all there in the TRS-80 Model III computer. But to be truly useful, that information must be brought out of the computer and into your daily routine. Not many of us like to spend 24 hours a day in front of the screen. Most want to get on with business, and that's what Profile III+ is all about—generating output you can use. Most of that output comes off your printer, to be distributed to colleagues, taken to meetings, included in presentations, mailed to clients.

If you think Profile III+'s printed output is limited to five types of reports and five types of mailing labels, then you've barely scratched the surface of all the program can do for you. This month we'll examine the many ways Profile III+ can put your data on paper.

REPORT FOR DUTY

Let's begin with the most obvious. First of all, Profile III+ allows you to create up to five report formats (Userfile/PR1 . . . 5). Each format can include up to two title lines, three header lines, and up to 66 lines worth of one- or two-line detail lines. (The default value is 60 so there will be margins around the perforations of fan-fold paper.) If you only request one detail line for each record, the program prints these lines consecutively. If you request two detail lines, the program skips a line between each record for increased readability. Finally, reports can be up to 132 columns wide.

This is all rather standard stuff. But Profile III+ gives you ways to make your output especially flexible, focused, and clear. First of all, don't forget that most 80-column dot matrix printers have a 16.7 c.p.i. (characters per inch) mode as well as the standard 10 c.p.i. mode. This means you can print 132-column reports on 8 1/2" paper. Give it a try; you'll find the condensed font surprisingly readable. You can automatically set your printer to this mode with a DO file called by your user menu. (See the December 1982 Microcomputer News and your printer manual for more details on creating a user menu tailored to your printer.)

Reports can be dated and paginated easily. Use the "/" indicator in a title or heading line for the date, leaving eight spaces. The program automatically gets the log-in date in MM/DD/YY format. Similarly, use the "#" indicator, leaving five spaces, for the page number. You must leave five spaces because the architecture of the system allows the page counter to reach 65,536, even though you're unlikely to generate a report that long. If you don't leave five spaces, the page number field overwrites the following field, eradicating the field indicator. (This happens whenever you fail to leave the allotted space for a field. The other result of careless formatting—over-allotment—causes the end-of-field marker on the formatting screen, which is also "/", to be left floating. And if it's in a title or heading line, it calls today's date where you weren't expecting it to appear.)

By the way, "/" and "#" can only be used once in a report format. The manual tells you this so that you avoid the following surprise: You might want to use the abbreviation "A/C" for area code in a column heading. It's a common abbreviation, and it fits perfectly. If you use it, however, your report will print "A03/14/83" and may overwrite the following characters as well. In the same way, if you use "Rec.#" for record number, you'll get the page number instead of the record number, causing a great deal of confusion. This happens because the program scans the title and heading lines from right to left, searching for the first instance of one of these special field indicators. Only if you use the date and page number indicators at the top right corner of your report do the symbols also become available for normal use below and to the left of that position.

A QUESTION OF POWER

One other field indicator is quite helpful when used in titles and headings: the powerful "?" field. This character alone can make one report format do the work of many.

When placed in a title line, the "?" indicator prints the contents of the sort field for the first record that appears on a particular page. Thus, it functions like the reference word that appears on the top of each page in a dictionary, helping to guide you to the information you seek. And it does more than that. It also prints the sort field data even if the sort field is not part of your detail line. For instance, you can sort a report by area code, having only "name," "address," and "sales representative" in your detail lines. The "?" indicator prints the area code of the first entry on each page. If you find it useful to sort alphabetically by sales representative later on, you won't have to change your format. "?" can be made to print the last name of the first sales representative on each page. If you are using associated fields, the "?" indicator is your only way of

knowing the sort basis of the records on a particular page of your report.

By the way, the "?" indicator prints as many characters of the sort data as you choose to include in the sort field length. This may be less than the length of the sort field (only the five critical digits in a ten-digit zip-code, for example) or more than the length of the sort field (to sort by first name within last names, which you can do if the first name field immediately follows the last name field on your "Define Files" screen).

ASSOCIATED FIELDS

Here's an example of using the "?" indicator with associated fields: At Pentacle, one of our priorities is keeping track of the presenters that are interested in our dance companies. Each "presenter" record has eight associated fields into which I enter the three-character codes for the various companies. My report detail lines tell me the information that I need about the presenters—name, address, phones, name of theatre, etc. The companies constantly ask me about potential bookings; and I periodically generate reports summarizing their prospects, using my interest field as a control break. The "?" field in the title of each page identifies the company that might be hired by the presenters listed on that page. Each company gets its own page (or two or three or four, if they're "hot"). I get a total for each company and a grand total at the end. Each presenter appears on as many pages as there are companies in which they've expressed an interest. Again, if you're using associated fields, this is the only way to identify the subject of each page.

If I change the control break to the field in which I store the state code, I can use the exact same format to generate a listing of presenters state by state. In this case, the "?" indicator prints the state on each page. If, at another time, I delete the control break entirely and sort by zip code, I get a continuous listing of presenters in zip order. This is invaluable for targeted marketing, as when I have to "fill out" a tour. The "?" prints the beginning of the selected zip code range.

It works like a charm. One word of caution though: Allow enough spaces after the "?" indicator for any of your likely sort choices, lest you overwrite other text or indicators.

HEADING THE LIST

The "?" field can also be placed in heading lines. In this case, it prints the heading you gave the sort fields when you originally defined your files. This prompts another word of caution: The number of characters available for field headings is 201 per segment, and each field used requires two of them for housekeeping. Segments that approach the limit of 36 fields have, of necessity, abbreviated or even cryptic headings. Try to anticipate which are most likely to be used with the "?" indicator in reports and make them more understandable. A colleague consulting your report might be hard-pressed to guess that "Sort by:I" means "alphabetically by name of institution."

I use the "?" field in heading lines in the following way. The fastest way to access your records is by record number, but of course these numbers are arbitrary, especially after a few months of additions and deletions. I found a way to get around this: First I defined a one-line report format that includes name of institution, state, zip, name of contact person, and record number. I used the "@" indicator for record number, leaving five spaces—it's a five-character field like page-number. (By the way, "@" only works on the first detail line of a two-line report.) My heading begins: "Sort by:?". I run three reports, one sorted by "Zip," one by "Name," and one by "Inst"; and then I use them as an index to my data base. I can locate any record in the time it takes to put the phone to my ear.

DETAILS, DETAILS

In a detail line, "?" prints the data in the sort field for each record. This can be useful when you're not using the control break to identify your sort criteria. It also prevents you from having to create several report formats when one will do. This is because "?" calls up the contents of your sort field, whatever it is to be at any one time. This means that you don't need to include all your likely sort fields in your format or create a different format for each one.

FIELDS OF BATTLE

There are a few other kinds of fields you might want to include in your data base because of their usefulness in organizing and generating output. Some have been mentioned in previous articles, but are worth reviewing. We've just discussed the convenience of including the record number ("@") in detail lines of reports. Another useful field is "Date of last update" ("&" or "@" on your screen formats). If you use the "@" option (YY/MM/DD), you can sort on this field and identify inactive clients, products that aren't moving, collectibles you've held for X years—whatever time-related attribute is important in your application.

Another extremely useful field is one I call the "marker." It's a one- or two-character flag, a sort of electronic "dog-ear" marking a record for special action of some sort. It can be alphanumeric or exclusively numeric—whatever is best suited to your needs. It enables me to browse through my records and decide, for instance, to send certain presenters a recent press clipping about one of my dancers. My selection process may not necessarily be logical enough to allow for a regular sort operation. Why not leave a little room for intuition? So I tag certain files in the marker field with any character I happen to select—a "P" for example. The next day my assistant sends copies of the clipping to all the records containing Ps. It's easy to print mailing labels and then a report listing for them.

Codes, of course, are more permanent markers. If you remember, we discussed the use of codes in last month's column. Read it if you haven't already; it should help you in designing reports.

SOME REPORTS ARE MORE EQUAL THAN OTHERS

Before leaving the subject of reports, here are a few tips on making them more readable and presentable. Profile III + automatically skips a line between title(s), heading line(s), and the body of the report. You might experiment with making these parts stand out even more by trying various types of lines and underlines. My own preference is to use a string of "=" signs, but try other signs—*, -, ., +,—and use those you find most appropriate for your reports.

Graphics are not only decorative but provide important practical applications as well. Using the graphic resources at your disposal to clarify heading lines makes your reports clearer, which, in turn, makes them more effective. A report is inherently a columnar document. If a field such as "Name"

varies greatly in length from one record to the next, it can be quite difficult to read because the eye has such irregular jumps to make. I've found it helps to define the total field length by centering the heading within equal signs, then skipping a space between fields. Look at figure 1:

Now compare it with figure 2—the same report made clearer with graphics:

As the example shows, it also helps to group fields logically; fields placed one above the other seem more closely related than fields placed side by side. Separate columns; don't hesitate to leave some blank parts. It will help you organize and clarify your information. I'm not a behavioral psychologist, but I suspect the blank part of the page is almost as important in certain respects as the inked part. True or not, it is definitely worth spending some time fussing with your report layouts. Ask some people who use your reports which they find easiest to read.

YOU'LL LIKE THESE LABELS

Your five label formats (Userfile/LB1 . . . 5) are even more flexible than your report formats, unless you insist on defining a label as a mailing label. Use your imagination; I'm sure you'll come up with clever uses for labels. For example, a label can also be a Rolodex card. Stationers supply these in a number of sizes. (Some form houses even sell perforated Rolodex cards on fan-fold card stock.) Match the size you prefer with the appropriate Radio Shack self-adhesive labels. Your labels can be sorted by any criteria you wish. The sort field doesn't necessarily have to be included in your label format, but if you want the field identified on the label, the way it is in the public library card catalogue, then use the "?" field on the top line. It prints the sort field data for the current record. Incidentally, the program truncates the "?" if it would otherwise run off the end of your label format.

If you need a lot of information on your "card," try this:

```
*Pentacle: Sponsor Interest Report for GRD   Region: ( )I   ( )II  ( )III ( )IV  ( )Foreign      Date of Report: 10/29/82 page   13
================================================================================================================================


==========Institution========== ========Contact person========= =========Address=========  =======Facility======== ====Source====

Coastal Carolina College         Marianna H.  Fox               PO Box 1954, College Rd.  Wheelright Auditor'm    Mode:SAE82
Conway              SC 29526     Chairman   Cultural Arts Comm.  Office phone:803 448-8419  W=    D=    seats=  820 Date:82/10/07

University of Iowa               Jim         Wockenfuss         Hancher Auditorium        Hancher Auditorium      Mode:ACUCMW82
Iowa City           IO 52242                                    Office phone:319 353-6251  W=    D=    seats= 2684 Date:82/09/23


RECORDS SELECTED    2
```

Figure 1.

```
*Pentacle: Sponsor Interest Report for GRD   ( )Region I (Northeast) ( )Region II (Rest of US)  Date of Report: 10/20/82 page    8
--------------------------------------------------------------------------------------------------------------------------------

CONTACT PERSON               TITLE       DEPARTMENT          INSTITUTION                FACILITY           SOURCE OF INFORMATION

Ann          Cochran         Asst. Dir.  Student Activities   Univ. of New Hampshire                        Convention ACUCNE82
Memorial Union Building  Durham              NH 03824        Office phone: 603 862-1524    W x    D, seats       Date 82/09/12

Douglas      Brown           Asst. Dir.  Student Activities   Monroe Community College                      Convention NACL82
1000 East Henrietta Road  Rochester           NY 14623        Office phone: 716 424-5200    W x    D, seats  600  Date 82/10/01


RECORDS SELECTED    2
```

Figure 2.

Define a one-up label format, which can be as much as 132 columns wide, but use two-up or three-up labels. Position your fields carefully so you don't overstep the label boundaries. You can print all the information you need from one record on two or three labels across; once printed, stick the labels one above the other on whatever cards you choose.

Deletion is easy. Toss out-of-date cards into your circular file. If you use the "date of last update" field ("@"), you can periodically run replacement labels for only those records with newly revised information.

Then there are file folder labels, tags for your pre-Columbian pottery, stickers for your inventory of plumbing supplies, titles for your videotapes . . .

LOOKIN' GOOD

Profile III + offers you two ways to make your labels look good. The "push = left field" ("〈") closes up extra spaces between fields. This means that if a name or title doesn't take all the spaces you've allotted for it, you have a way of seeing to it that the following field comes right after, closing in the extra space. For example, if the city field is 10, the state field is 11, and the zip field is 12, then you should format as follows:

```
*10      /〈11〈12
```

It's good to print mailing labels using your printer's correspondence quality proportional space font. The only trouble is, the program loses track of the left margin of the right-hand labels. The solution is simple. Use a one-up label format but two-up labels. Make a note of the number of records selected for your label run, and pause printing (press 〈BREAK〉 once) when half of them have been printed. Reverse the label sheets and go back the other way. (Press any other key to resume printing.)

EIGHT-LINE REPORTS

Do you absolutely need a report format longer than two lines? Well, you've got it. Define a one-up label format using the whole width of your printer. It can be up to eight lines long. Print your "labels" on fan-fold paper. The only thing you'll be missing are titles, headers and page numbers. But you can enter text, if you wish, to identify fields. Adjust your format so that a whole number of records fits on each 66-line page. Eleven six-line records fit perfectly. Use "push = left fields" so that first and last names aren't separated by too many spaces.

HARDCOPY IS EASY

There are two ways of making a hardcopy of anything that appears on the screen.

The first uses Profile III + 's "vidscan" feature. At any point in the program in which the program recognizes 〈BREAK〉, press 〈BREAK〉 once. A flashing prompt appears at the bottom of the screen. If you press 〈BREAK〉 again, you will quit the current operation. If you press any other key, you will cancel the break and resume what you were doing. But if you press 〈SHIFT〉 and 〈@〉 at the same time, the program will do a "screen dump" to your printer. This is useful both for printing information from a particular record when it doesn't make sense to run a whole report and also, for documenting your formats for reference. This way, you can easily assemble a user manual for your application.

You can also do a Mass Hardcopy, using any screen format, of any and all records you choose. Sort and select as usual. When the first sorted record appears on the screen, press 〈SPACE BAR〉 once to call up the Mass operation command line. The two Mass operations that involve your printer are "hardcopy" and "purge." ("Purge" prints out each record before deleting it. "Delete" eradicates records without printing them first.) Select 〈H〉, and the program transfers the contents of each record to your printer.

At Pentacle, we use this feature to print out simple invoices and form letters. Many of our dancers belong to our group health insurance plan. We designed a screen format to remind them when their quarterly premiums are due, positioning the names and addresses to show through a window envelope. Quarterly billing that used to take hours now takes only minutes, and we don't even have to address envelopes.

WAIT! THERE'S MORE!

Profile III + can interface with Radio Shack's SuperSCRIPSIT word processing program (26-1590). This means you can perform merge operations that generate personalized form letters as a result. I'll describe this in detail in an upcoming issue of Microcomputer News.

PROFILE Editor's Note: This is Mr. Sygoda's sixth article in a series of 'how-to' Profile III + articles. Other articles in the series will be published over the next few issues in this column. We hope that you enjoy this feature, and we look forward to your comments and questions on Profile III +.

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

KEEP THOSE LETTERS COMING IN

Readers of the Microcomputer News have been writing in with exemplary questions about Profile III + . From time to time, we try to answer some of them in this column.

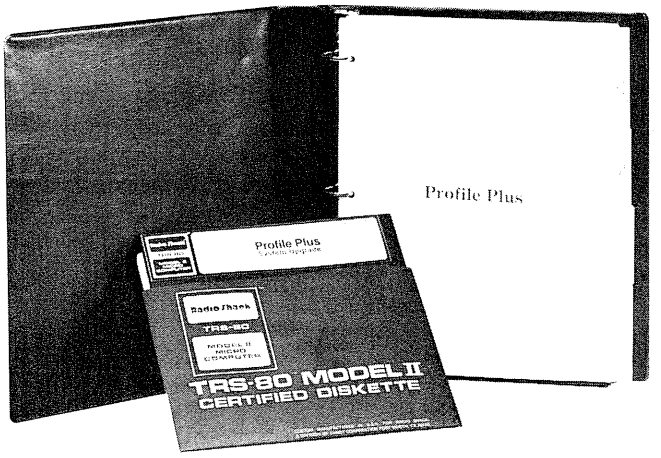
SORT DIVISION

Bill Pickard wrote in from Harrison, Arkansas, to ask how to divide a data base, as we suggested in the November 1982 article, if it is already defined and filled with data. You should consider doing this when the number of records in your file approaches your system's capacity to store them on disk. The procedure is really quite simple. "Clone" your data disks and then delete complementary records.

For the purposes of this example, let's say that you have a two-drive system and have placed your Profile III + runtime and formatting programs on the disk in drive 0. All your data segments are on the drive 1 disk, and you have entered the 1,000 or so records you have room for. Make an extra backup of this data disk. Then decide by which criteria to separate your data base. If your data base is a mailing list like Mr. Pickard's, geographical division is the logical choice because you'll most often be sorting by zip code. The criteria you select can involve one or more of the key fields.

Next, determine the zip code of the person halfway down the zip-sorted list—72601, for example. About half of the records have zip codes less than or equal to 72601 and the other half greater than or equal to 72602. Take one of the two identical data disks and perform a Mass Delete of all records involving zip codes less than or equal to 72601. To do this, sort by "zip", and select "LE 72601". When the first record appears on the screen, press the 〈SPACE〉 bar once to get into Mass Mode. The Mass Operation command line appears at the bottom of the screen. Select "Mass Delete." The pro-

gram deletes the appropriate records, leaving the half with zip codes 72602-99999. Label the disk accordingly, and from now on only use it to enter records that go on that half of your data base. On the other disk, delete all records with zips greater than or equal to (GE) 72602. After you do this, of course, each disk will have the same number of records it had before, but half of them will now be empty. As you enter new records, the program begins to fill the empty slots. Be sure to make backup copies of each half of your data base.



HAPPY APPENDING

Did the Easter Bunny bring you an expansion drive? It's easy to "unclone" your data base and reverse what we just did to it. From TRSDOS, use the ⟨**APPEND**⟩ library command to recombine data segments, segment by segment. For example, COPY MAILLIST/KEY:1 :2 from the data disk holding zip codes 00000-72601 in drive 1 to a formatted disk in your new drive 2. Now place the data disk holding zip codes 72602-99999 into drive 1. APPEND MAILLIST/KEY :1 :2. TRSDOS adds the records in the drive 1 key segment at the end of the records already in the drive 2 key segment. Repeat these steps for MAILLIST/DAT, MAILLIST/DAT2, etc. You can move these files around from TRSDOS at any time and to your heart's content, shuttling segments from disk to disk as a function of available space and relative segment size. Profile III + will find any segment it needs automatically.

OUT OF SORTS

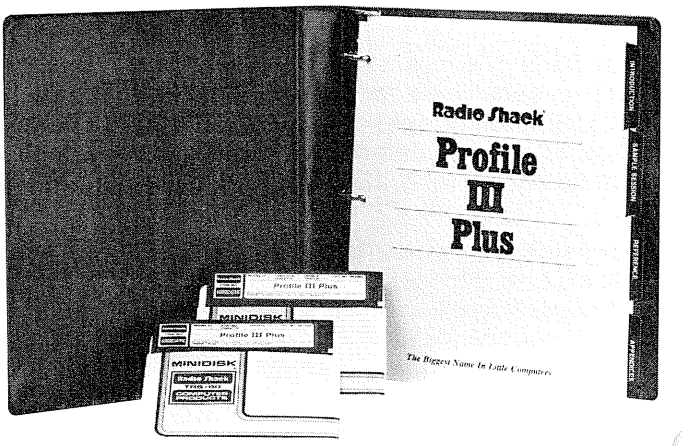
Joel Schwartz wrote from White Plains, New York, to observe that Profile III + 's alphabetical sort does not follow all the rules of alphabetization. In particular, the sort does not ignore spaces, hyphens, and apostrophes the way the phone directory does. As a result, Profile III + places O'Hara before Ohana.

This problem is not really Profile's but ASCII's (American Standard Code for Information Interchange). This standard, used by almost all computer manufacturers, assigns a code number to each letter, numeral, and punctuation mark. Your BASIC Operation Manual has a list of these codes. Alphabetization really involves sorting in ASCII order, code number by code number, which is why spaces and punctuation marks are not ignored. Therefore:

O'Hara (79-39-72-97-114-97)
comes before
Ohana (79-104-97-110-97)

Very expensive software such as the phone company's has a "filter" to catch punctuation that the user wishes to ignore. Profile III + doesn't, but there is an easy trick if your application demands the sort of alphabetizing Mr. Schwartz does.

When you define your files, create a second last name field that parallels the "real" last name field. Call it "alpha." After you enter O-'-H-a-r-a in the real name field (#1), enter O-h-a-r-a in the alpha name field (#2). Whenever you sort by name, sort on field 2, but print field 1 in your formats. (The sort field does not have to appear in the screen, label, or report format you are using. It just has to exist as a key field in your file.) If you do this, your listings will be alphabetized the way you learned in school.



FORTRAN Files *(From Page 29)*

```
41Ø KO = Ø
42Ø 'READ EACH CHARACTER FORM 8Ø BYTE RECORD & PUT
     OUT TO ONE BYTE RECORD.
43Ø 'IF MORE THAN 8 BLANKS ARE ENCOUNTERED, ASSUME
     END OF LINE.
44Ø FOR I = KT TO 79
45Ø     IF I = 6 AND KK = 1 THEN FOR J = 1 TO 2
            : FT$ = CHR$(32)
            : GOSUB 56Ø
            : NEXT J
46Ø IF KO > 8 THEN 52Ø
47Ø IF E$(I) = CHR$(32) THEN KO = KO + 1 ELSE KO = Ø
48Ø FT$ = E$(I)
49Ø GOSUB 56Ø
5ØØ NEXT I
51Ø 'PUT OUT CARRIAGE RETURN AT LINE END, AND RETURN
     FOR NEXT LINE.
52Ø I = 8Ø
     : FT$ = CHR$(13)
     : GOSUB 56Ø
53Ø R% = R% + 1
54Ø GOTO 14Ø
55Ø 'SUBROUTINE TO PUT ONE BYTE CHARACTERS OUT TO NEW
     FORTRAN FILE.
56Ø LSET F$ = FT$
57Ø RE% = RE% + 1
58Ø PUT 2, RE%
59Ø RETURN
```

Notes Previous

JULY 1980
MACHINE LANGUAGE SORT FOR MODEL I AND MODEL II

Hoyt W. Huggins
2605 W. 120th Terrace
Leawood, KS 66209

I have a Model I disk system, 48K. I just purchased all the back issues of TRS-80 Microcomputer News and am intrigued by all the articles, letters, and follow-ups on sorting. I want to add my two-cents worth to see if I have something useful to offer or if my idea is full of holes.

I use the embedded machine language sort presented by Allan Emert of Odessa, Texas. I select a key element of the record or key portion of the multi-dimensioned array (or single dimensioned array, for that matter) and, if it is numeric, I convert it to a uniform size string. I use either single or double precision numbers, depending of course, on the degree of accuracy I want, and I multiply the value, say VAL#, by 100 or 1000, this time determined by the required accuracy of the sort (in the event of near duplicate values). This moves the decimal to the right and then, in the same program line, I truncate the decimals of VAL# and FIX and convert the result to an unsigned string with A$(I) = RIGHT$ (STR$ (VAL#), LEN (STR$(VAL#)) – 1) Also, in the same program line, I find and retain the value of the longest string (L%) by: IF L% < LEN(A$(I)) THEN L% = LEN(A$(I)). When the A$(I) array is loaded, I run it through a quick loop to widen each element to a uniform length and tack on a pointer to be stripped off later for the print routine, thus:

```
200 FOR I=FIRST TO LAST
210 IF LEN(A$(I) < L% THEN A$(I) = "0" + A$(I) GOTO
    210 'FRONT LOADS THE ELEMENT WITH ZEROES SO THE
    SORT WILL WORK
220 A$(I)=A$(I)+"/"+STR$(I) 'ADDS THE POINTER BY /
230 NEXT I
```

For the print routine this program portion would apply:

```
500 FOR I = FIRST TO LAST
510 L1 = INSTR$(A$(I),"/")  'FIND THE POINTER BARRIER
    SYMBOL
520 L2 = LEN (A$(I)) 'FIND THE OVERALL LENGTH OF THE
    ARRAY ELEMENT
530 P(I)=VAL(RIGHT$(A$(X),L1-L2) 'STRIP OUT THE
    POINTER CONVERTING IT TO A NUMERIC
540 NEXT I
```

The print routine might continue:

```
600 FOR I = FIRST TO LAST
    :GOSUB 1000 'FIELD BUFFER FOR SUB-RECORDS
610 GET 1, P(I) 'PULL UP THE NEXT RECORD "POINTED TO"
620 ... convert field strings to variable strings and
    numerics, print the record as required
630 NEXT I
```

Of course, adjustments are necessary for sorting sequential files and other arrays, but two things must be emphasized: 1) Using the pointer method along with the Emert Sort reduces the processing and sorting time from long boring minutes to mere seconds, sometimes, a mere flash; and (2) the thing just will not work unless the programmer follows Emert's instructions to the letter.

JULY/AUGUST 1982
ASTEROIDS

Kenneth Baker, Jr.
R.R. 1 Box 106C
No. Manchester, IN 46962

In the game by Richard Zepp, I reached a score of 6982 on Screen 110. I did not change anything as Joseph Prisco did. I took it directly from the magazine.

SEPTEMBER 1982
MODEL II
BUGS, ERRORS, AND FIXES
ACCOUNTS PAYABLE (26-4505)

The final part of Line 224 should read:

```
:AB$=V$ELSE...
```

OCTOBER 1982
MAILING LABELS ON THE COCO

Line corrections are as follows:

```
220  eliminate space in STZP*ACPHCODE
1265 LN$(1) = LN$(2) + " " + LN$(1)....
1281 PRINT#-U, TAB(1) LN$(1); TAB(41) LN$(1)
```

NOVEMBER 1982
NAVIGATION WITH A CC

Garrett E. McGowan
3207 Evergreen Avenue
Hopewell, VA 23860

I'm not sure of the sailing possibilities in Tucson, but if you can catch the Commander before he puts to sea with his "Navigation" program, please do so, because undoubtedly, he will become lost at sea. I can well understand why he would be concerned with Latitudes North and South and Longitudes West and East as he inputs in lines 155,160,165, and 170, but he doesn't do anything with the input within the program. Likewise his conversion to Kilometers in line 325 is incorrect. His way makes less Kilometers than Miles. If he'll change line 325 as it reads "DK = DM*KM" to read "DK = DM/KM", this will get the proper conversion to Kilometers when printed in Line 410. I have neither the time nor inclination to debug his failure to use N,S,E,W in the program.

If you can catch him before he's lost at sea, advise him to use the Shortwave Program on pages 36-38 of the January 1982 issue of TRS-80 Microcomputer News. I don't wish to see a fellow Retiree lost. Secondly, we need all the CoCo programmers we can get!

DECEMBER 1982
CHRISTMAS CARD

Mary Bevere
171 North Hills Drive
Parkersburg, WV 26101

Regarding the program by G. Pigounias of Savannah, Georgia, it came to my attention that there are three mistakes in line 100. One may have been typographical. The other two were probably overlooked by the programmer. The lines should read as follows:

```
100  DRAW
     "C0;BM16,12;N;U10;N;R10;N;D15;N;L10;N;E5;N;G5;N;H
     5;N;F5"
100  DRAW "M16,12;C0;N;V10;...F5;"
```

I thought this should be brought to your attention. I also changed C0 (white) to C2 (yellow) because it fit the season and was attractive.

My father owns a TRS-80 Color Computer and my family enjoys it immensely (I am fifteen). I enjoy the hi-resolution graphics, my father, the financial aspects, and my mother, the Bridge Tutor Cartridge.

CLOCK/SCHEDULER FOR THE BUSY EXECUTIVE

There are two corrections to the program:

```
21Ø CURSOR 12
     :PRINT "The time is"...
22Ø RESTORE 2
     :FOR I=1 TO 6 ...
```

CONCENTRIC CIRCLES ON THE COCO

Henry Nielson
9001 Gray Ave.
Ocean Springs, MS 39564

The program will play just as well on a 4K Non-Extended Color Computer with these changes:

Add line:

```
165 M=7Ø.7-D
```

Change lines:

```
18Ø IF F=8 THEN F=1
2ØØ J=16+(.6*(A*SIN(M)))
```

There is also an error in Line 190 in the original. It has too many))) after the D.

I am sure there are other ways to make it work in Non-Extended, but this is at least one way of doing it.

MODEL II TWINKLING CHRISTMAS TREE

Joe Enderson
34 Kenwood Dr.
Hampton, VA 23666

I am 14 years old, and I frequently use my father's TRS-80 Model II computer. I was putting the program into the computer and found it didn't work. My father and I debugged the program and found an error (probably a print error).

```
1Ø5 PRINT@S(Y),CHR$(42)...
```

NOTES PREVIOUS

Peter Day
287 Pythian Ave.
Hawthorne, NY 10532

I am a secondary math teacher and did enjoy Joel Jacobson's program on primes. However, a line must be added to exclude the number 1 (one) from the list of primes as one is not a prime number.

Add Line:

```
35 IF N=1 THEN 1ØØ
```

COMPUTER CLUBS

ACSCO/AMATEUR COMPUTER SOCIETY OF CENTRAL OHIO
c/o Joel Sampson
665 Royal Forrest Blvd.
Columbus, OH 43214
1-614-262-2993
Bulletin Board: 1-614-837-CBBS

COCO-MUG/COLOR COMPUTER MILWAUKEE USER'S GROUP
c/o Joe Kleinwaechter
710 E. College Ave. #2
Waukesha, WI 53186
1-414-548-9919

COCONUTS COLOR COMPUTER USERS GROUP
c/o Robert Sullivan, Jr.
100 Kerri Lyn Road
Warwick, RI 02886
1-401-739-8743

COLOR COMPUTER CLUB OF ALBANY
Rt. 5 Box 4 Woodcliff Street
Albany, GA 31707

COMPUTER USERS' CLUB
c/o Ken Laninga
9705—88 Avenue
Grande Prairie
Alberta
Canada T8V 0B6

JACKSON AMATEUR COMPUTER SOCIETY
c/o Camp Baker Corral
2355 Camp Baker Road
Medford, OR 97501

L.A. COLOR COMPUTER CLUB
c/o Mark Randall
2227 Canyon Rd.
Arcadia, CA 91006
1-213-355-7719
Bulletin Board: 1-213-563-7727

LICA/LONG ISLAND COMPUTER ASSOCIATION
Bulletin Board: The Rainbow Connection
1-212-441-2807

LIVERMORE FALLS COMPUTER CLUB
c/o Peter Richard
F.M. Electronics
Jay Shopping Center Rt. 4
Jay, ME 04239
1-207-879-6904

MACCS/MARION COUNTY COMPUTER SOCIETY
c/o Roy G. Kahkonen
P.O. Box 248
Silver Springs, FL 32688
1-904-694-2871

MALDEN CATHOLIC HIGH SCHOOL COMPUTER CLUB
c/o Brother Fred Eid
Malden Catholic High School
Malden, MA 02148
1-617-322-3098

MODEL II/16 BUSINESS USERS GROUP
c/o Laren R. Januz and Associates, Inc.
P.O. Box 631
Lake Forest, IL 60045
1-312-362-0016

Bugs, Errors, and Fixes

NOTE TO USERS

The following changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it unless we specifically direct you to by mail. All Customer Service Bulletins (outlined in the "Software Maintenance" article in this issue) will contain the phrase "This is a Required Modification" or "This is an Optional Modification." It is up to you to decide whether or not to apply an optional modification; however, you should always implement a required modification regardless of your past experience with a package. The error it prevents may be just around the corner, and the modification could save you many hours of frustration. If you feel that changes should be made, but you do not feel qualified to make the changes yourself, please 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.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that required modification and make the PATCHes to the backup copy of the disk.
2. Apply PATCHes according to the information given in your TRSDOS manual.

Color Computer

POLARIS (26-3065)

When two players are playing, an opponent's missiles can be disabled by holding down the button on your joystick during your opponent's turn. When he/she tries to fire a missile nothing will happen.

MONSTER MAZE (26-3081)

In the second paragraph titled Aggression! on page 4 of the manual, the last two sentences should be deleted and replaced with:

"They can shoot right through walls—a trick your gun can perform once you've learned some sneaky tactics."

COLOR SCRIPSIT (26-3105)

In the fourth paragraph from the bottom of page 33 of the manual, the sentence that reads as shown below should be deleted from the manual. It is incorrect.

"(If the program has already started trying to print and something is wrong with the printer, hold down the ⟨BREAK⟩ key until the Main Menu appears. Press ⟨2⟩ and your text will reappear.)"

MUSIC PROGRAM PAK (26-3151)

We received the following information on this package which we thought might be of interest to owners of this package.

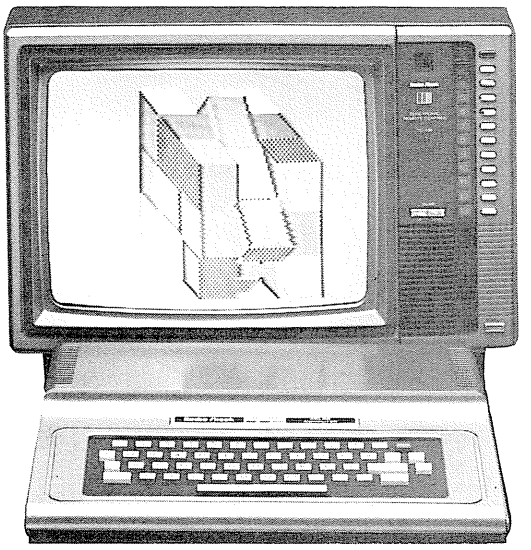
The instructions explain how to delete a note or other input but make no mention of how to insert a note into a composition. Pressing the letter I opens a space in the composition allowing a note to be inserted. One note can be inserted for each time the I is pressed. The message EMPTY prints at the bottom of the screen. Pressing the right arrow after inserting a note will close the gap and you can continue.

Pressing the letter M places a vertical "measure" line across the staff. The message MEASURE prints on the bottom of the screen. you must then press the right arrow to continue.

MADNESS AND THE MINOTAUR (26-3313)

On page 3 of the manual under the section that begins "The screen will show:", it should read as follows.

COLOR BASIC v.r.
(C) 1980 TANDY
OK



Model I/III

MODEL III COMPUTER

When writing machine language programs for the Model III, if the HL register pair happens to point to a location which contains hex 23 at the time of a user call to $CSHIN or $CSHWR, the result will be a BASIC error message such as SN ERROR. To solve the problem, always ensure that HL is loaded with zero prior to calling one of these ROM routines.

5-MEGABYTE HARD DISK (26-1130)

Never move the disk drive unit while the drive is running! Permanent damage to the drive or disk may occur if you do move the unit while it is running.

Also, do not expose a disk drive unit to a strong magnetic field (such as a bulk eraser); doing so may cause you to lose data or damage the unit. Hard disks cannot be bulk erased!

RUNCOBOL (26-2203) WITH LDOS

When OPENing a sequential file with the EXTEND phrase, it is possible to receive an ERROR 94 on a file that was previously opened, written to, and closed.

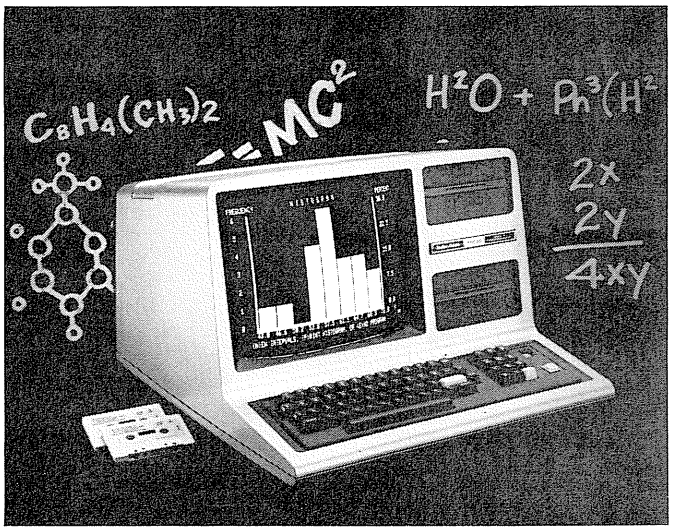
To repair the problem at LDOS Ready, type:

```
PATCH RUNCOBOL/CMD(X'522F'=CD 48 44 C8 FE 1C CØ AF
C9)
PATCH RUNCOBOL/CMD(X'9A4Ø'=2F 52)
```

All previous PATCHes to RUNCOBOL must be applied before PATCHing and running under LDOS. If you do not have the patches contact your Radio Shack Computer Center or a Radio Shack store with a Computer Marketing Representative for the previous patches.

MODEL III HARD DISK OPERATING SYSTEM (26-1130)

The addendum that comes with this package states that VisiCalc and Scripsit are moved by using CONV :X :X (VIS). The Scripsit and VisiCalc files are not visible files and will not be moved with the VIS parameter.



Pocket Computer

ENGINEERING MATH II (26-3526)

The last paragraph on page eight refers to a non-existent polar mode. Delete the last paragraph.

ENGINEERING MATH IV (26-3528 VERSION 1.0)

Function SHFT-S (solve SAS) will solve the two unknown angles incorrectly if Angle A is larger than 90 degrees.

* * THE FOLLOWING LINE CHANGES ARE REQUIRED! * *

* * * * NOTE: The problem can be fixed by making the following changes in the program. These changes must be made in precisely the order given in order to avoid problems with memory shortage.

The problem is corrected by following the steps listed below.

1. Load the program by typing CLOAD"TRIANG".
2. Make the following corrections:

CHANGES (Retype the line or refer to the Edit section of the owners manual)

The following change releases enough memory to implement the correction.

Line:

```
9 PRINT "SOLVE 3 POINTS: SHFT-F":
            PRINT "MENU: SHFT-C":
            END
```

The following changes correct the problem.

Line:

```
52 B=  (AA+CC-(2AC*COS E)):
             PRINT "SIDE B = ";B:
             D=X-(ACS ((A*SIN E)/B)*(1-2*(A*COS
             E>C))
```

Line:

```
54 PRINT "ANGLE A = ";D:
             F=Y-D-E:
             PRINT "ANGLE C = ";F:
             GOTO 2
```

4.Type CSAVE"TRIANG" to save the changes in the program.

PLOTTER CASSETTE INTERFACE OWNER'S MANUAL (26-3605)

Correction 1—There are several pages that need correction in the manual.

Page 46—In the second paragraph from the bottom of the page, the sentence "With @( ) . . . , or to fixed memories." should be deleted. It is incorrect.

Page 47—The syntax in the first program line under examples is incorrect. It should read:

```
INPUT # "PROGRAM"
```

Delete the comma that follows # in the manual.

Page 48—Program line 100 and the sentence following it are incorrect and should be deleted.

The other two program lines shown on 48 have syntax errors in them. The commas that follow the # in both lines should be deleted.

Page 52—The sentence in the last paragraph which reads "With A( ), A$( ), . . . or to fixed memories, such as @." is incorrect and should be deleted.

Page 53—Program line 10 contains a comma after # which should be deleted or a syntax error will result.

Page 54—The two program lines shown on the page have syntax errors in them. Delete the comma which occurs after the # in both lines.

Page 55—The second and third program lines numbered 30 should be changed as follows.

The second line 30 should read:

```
3Ø:PRINT"filename";T$
```

The third line 30 should read:

```
3Ø:PRINT#-1"filename";T$
```

Correction 2—The last paragraph on page 46 beginning with "Important Note: . . . " explains that arrays must be DIMensioned previous to an INPUT # statement, but fails to note that any other variables that are not fixed variables should also be created before executing INPUT #. (Variables such as: A0, Z1, XX, etc.).

Example:

Program 1

```
10 A9=10
20 PRINT#"DATA";A9
30 CLEAR
40 INPUT#"DATA";A9
```

Program 2

```
10 A9=10
20 PRINT#"DATA";A9
30 CLEAR
35 A9=0
40 INPUT#"DATA";A9
```

Program 1 would result in Error 43 in line 40 because the CLEAR in line 30 erased the variable A9 from memory. In program 2, line 35 creates the variable A9 in memory; therefore, line 40 will work.

Model II

TRSDOS-HD (26-4150 VERSION 4.1)

It is not possible to change the number of read/write retries made when accessing the hard disk.

* * THE FOLLOWING PATCH IS REQUIRED! * *

The following patch will allow the user to change the number of read/write retries made when accessing the hard disk. The controller will make a number of accesses equal to 16 times the digit in the first byte of the patch before rendering a sector unaccessible. The second byte of the patch relates to the number of attempts the controller will make to verify read after write.

At TRSDOS Ready, apply the following patch.

```
PATCH SYSRES/SYS R=20 B=16 F=0101 C=0303
```

NOTE: The system must be rebooted to effect the change.

PROFILE PLUS (26-4515 VERSION 1.0)

In the SELECT/BAS BASIC program provided on the Profile Plus master, if a comma is in the text that is being selected, everything after the comma is out of order.

* * THE FOLLOWING LINE CHANGES ARE REQUIRED! * *

The problem is corrected by following the steps listed below.

1. Backup the diskette(s) and make the changes on the Backup copy of the program.
2. In BASIC load the program by typing LOAD"SELECT/BAS".
3. Make the following corrections:

CHANGES (Retype the line or refer to the Edit section of the owners manual)

Line:

```
80 LINE INPUT #1,A$:IF LEN(A$)=0 THEN GOTO 100
```

Line:

```
140 IF EOF(1) THEN CLOSE:END ELSE LINE INPUT#1,A$:IF
    LEN(A$)>1 THEN  A$=MID$(A$,2,LEN(A$)-2)
```

4. Type SAVE"SELECT/BAS" to save the changes in the program.
5. At TRSDOS READY, make a backup copy of the corrected diskette.

SCRIPSIT (26-4531 VERSION 2.0)

In order to bring SCRIPSIT 2.0 up to a SCRIPSIT version 2.1, two patches must be made to the program STARTUP.

* * THE FOLLOWING PATCHES ARE REQUIRED! * *

The following patches will change the version number on the screen from 2.0 to 2.1 and change the date at the bottom of the screen from "*09/30/1982*" to "10/16/1982".

* * * NOTE * * * All previous Scripsit patches must be applied prior to applying the following two patches. After these two patches have been applied your SCRIPSIT will be running on version 2.1.

At TRSDOS Ready, apply the following patches.

```
PATCH STARTUP A=E300 F=30 C=31
PATCH STARTUP A=E40D F=AAB0B9AFB3B0AFB1B9B8B2AA
                     C=A0B1B0AFB1B6AFB1B9B8B2A0
```

Version 2.1 comes with the following default User Defined Print Codes:

CTRL X B is defined as 1B1F to turn Boldface on.
CTRL X L is defined as 1B20 to turn Boldface off.
CTRL X U is defined as 0F to turn Underline on.
CTRL X O is defined as 0E to turn Underline off.

There are two programs included on the release version of SCRIPSIT 2.1, DEFAULTS and PRTPCH. The program DEFAULTS allows a user to transfer diskette defaults between SCRIPSIT diskettes. PRTPCH will patch the print character width and the backspace distance required for a particular printer.

The directory has been altered to reflect a creation and update date of 10/16/1982 for all SCRIPSIT and SCRIPSIT related programs.

DOCUMENT/CTL, the file that contains all system defaults and training documents, has been reduced in size by 5 grans. This is to accommodate the two programs mentioned above and will allow installation of the SCRIPSIT Spelling and Hyphenation Dictionary plus the SCRIPSIT Utilities in any order of installation.

RSCOBOL DEVELOPMENT SYSTEM (26-4703)

On page 21 of the RSCOBOL section, the manual states that a back slash (\) causes page ejection. Using a back slash causes an error message. A slash (/) should be used for page ejection.

Peripherals

DIGITIZER (26-1195)

In the sample program for the Color Computer on page 34 of the manual, line 170 should be removed. A syntax error occurs when the program is run with this line still in.

The Digitizer and the Model II

Alfred Pacheco
349 Westmoreland
Richland, WA 99352

The sample program that came with the Radio Shack digitizer (26-1195) for the Model II was completely written in BASIC. This made the subroutines which interface the digitizer to one's programs, in my opinion, quite lengthy and difficult to use.

The following machine language routine was written to overcome this. It can be entered using the DEBUG utility program and saved using the DUMP command and recalled later by your BASIC program.

```
5E 23 56 2B D5 E5 21 39    EF 16 Ø8 3E 6Ø CF 2Ø FB
78 E6 7F FE 58 28 F2 77    23 15 2Ø EF Ø6 Ø1 21 39
EF 3E 18 CF E1 73 23 72    21 3D EF Ø6 Ø1 3E 18 CF
E1 23 23 73 23 72 3E Ø2    C9
```

Figure 1.

The only variable needed by the subroutine is a one dimensional integer array that holds two variables. This should be defined early in your program with the command

```
DIM II%(1)
```

The variable II%(0) will contain the X coordinate from 0 to 1100, and the variable II%(1) will contain the Y coordinate from 0 to 1700. Some digitizers may have a greater max value for the X and/or Y axis. In mine, the max value for the X axis was 1107, and for the Y axis it was 1766.

LOADING THE PROGRAM

The program, as shown, uses Channel A of the RS-232 port. If you wish to use Channel B for the digitizer, change the 60 at location EF0C to 62.

To enter the machine language program, follow these steps:

1) When in TRSDOS READY, enter DEBUG ON. After the response DEBUG ON, enter DEBUG.

2) When in DEBUG mode, enter M ⟨**ENTER**⟩ EF00 ⟨**ENTER**⟩.

3) Press the ⟨**F1**⟩ key

4) Enter the hex code as shown in Figure 1.

5) Press the ⟨**F2**⟩ key.

6) Press the ⟨**S**⟩ key.

7) Type DUMP DIGIT START=EFØØ, E N D = E F 3 8 ⟨**ENTER**⟩

The program that follows (Figure 2) illustrates the use of this subroutine.

```
1Ø DEFINT I-N
2Ø DIM II(1)
3Ø DEFUSRØ = &HEFØØ
4Ø SYSTEM "LOAD DIGIT"
5Ø REM
6Ø CLS
7Ø PRINT TAB(18); "DEMO PROGRAM"
8Ø PRINT
9Ø PRINT
1ØØ PRINT
11Ø PRINT"     (1) SET SCALE"
12Ø PRINT"     (2) MEASURE DISTANCE BETWEEN TWO
     POINTS"
13Ø PRINT"     (3) MEASURE THE LENGTH OF A LINE"
14Ø PRINT"     (+) EXIT PROGRAM"
15Ø PRINT
16Ø PRINT
17Ø PRINT "PLEASE ENTER YOUR CHOICE    ";
18Ø A$ = INKEY$
     : IF A$= "" THEN 18Ø
19Ø ON INSTR("123+",A$) GOTO 2ØØ, 42Ø, 73Ø, 62Ø
2ØØ REM SET SCALE
21Ø CLS
22Ø PRINT TAB(3Ø); "SET MEASUREMENT SCALE"
23Ø PRINT "PLACE CURSOR OVER FIRST REFERENCE POINT,
     PRESS SEND SWITCH."
24Ø GOSUB 67Ø
25Ø X1 = X
26Ø Y1 = Y
27Ø PRINT @(5,Ø), "PLACE CURSOR OVER SECOND REFERENCE
     POINT, PRESS SEND SWITCH."
28Ø GOSUB 67Ø
29Ø DX = X1 - X
3ØØ DY = Y1-Y
31Ø DS = SQR(DX * DX + DY * DY)
32Ø IF DS = Ø THEN PRINT  @(7,Ø),"POINTS MUST
     DIFFER."
     : PRINT
     : GOTO 27Ø
33Ø PRINT@(6,Ø),"";
34Ø INPUT "WHAT UNITS OF MEASURE ARE YOU USING";UN$
35Ø PRINT @(7,Ø),"HOW MANY "; UN$; "S LONG WAS THE
     REFERENCE DISTANCE";
36Ø INPUT DØ
37Ø SC = DØ / DS
38Ø PRINT @(9,Ø), "SCALE SET TO "; SC ;UN$ ;"S PER
     INCH."
39Ø PRINT "PRESS ANY KEY FOR MENU"
4ØØ A$ = INKEY$
     : IF A$ = "" THEN 4ØØ
41Ø GOTO 5Ø
42Ø  REM   FIND DISTANCE BETWEEN TWO POINTS
43Ø CLS
44Ø PRINT TAB(23); "MEASURE DISTANCE BETWEEN TWO
     POINTS"
45Ø PRINT
46Ø PRINT "PLACE CURSOR OVER FIRST POINT, PRESS SEND
     SWITCH."
47Ø GOSUB 67Ø
48Ø X1 = X
49Ø Y1 = Y
5ØØ PRINT @(5,Ø), "PLACE CURSOR OVER SECOND POINT,
     PRESS SEND SWITCH."
51Ø GOSUB 67Ø
52Ø DX = X1 - X
53Ø DY = Y1 - Y
54Ø DS = SQR(DX * DX + DY * DY)
55Ø DS = .Ø1 * FIX(DS * 1ØØ + .5)
56Ø PRINT @(9,Ø), USING "LINEAR DISTANCE = ##.##
     INCHES." ;DS
57Ø PRINT "SCALE DISTANCE =" ;SC * DS; UN$ ;"S."
58Ø PRINT
59Ø PRINT "PRESS ANY KEY TO RETURN TO MENU"
6ØØ A$ = INKEY$
     : IF A$ = "" THEN 6ØØ
61Ø GOTO 5Ø
62Ø REM END PROGRAM
63Ø CLS
64Ø PRINT@834, "PROGRAM ENDED"
65Ø END
66Ø REM INPUT FROM DIGITIZER
67Ø II(Ø) = USRØ(VARPTR(II(Ø)))
68Ø X = II(Ø) * .Ø1
69Ø Y = II(1) * .Ø1
7ØØ PRINT @(22,Ø),USING "LOCATION:
     (###.##,###.##)";X;Y
71Ø RETURN
72Ø REM MEASURE THE LENGTH OF A LINE
73Ø CLS
```

```
     : X1 = Ø
     : Y1 = Ø
     : DT = Ø
74Ø PRINT TAB(23) "MEASUREMENT OF A LINE SEGMENT"
75Ø PRINT "PLACE THE CURSOR OVER THE BEGINNING OF THE
     LINE SEGMENT.
76Ø PRINT "PRESS AND HOLD THE SEND SWITCH FOR 2 SECS.
     THEN TRACE OUT THE LINE"
77Ø PRINT "PRESS ANY KEY WHEN FINISHED, FOLLOWED BY
     PRESSING THE SEND SWITCH."
78Ø GOSUB 67Ø
79Ø X1 = X
     : Y1 = Y
8ØØ GOSUB 67Ø
81Ø DX = X1 - X
     : DY = Y1 - Y
     : X1 = X
     : Y1 = Y
82Ø DS = SQR(DX * DX + DY * DY)
83Ø DT = .Ø1 * FIX(DS * 1ØØ + .5) * SC + DT
84Ø PRINT @(23,Ø), "LENGTH = " ;DT ;UN$ ;"S";
85Ø A$ = INKEY$
     : IF A$ = "" THEN GOTO 8ØØ
86Ø PRINT @(12,Ø),"PRESS ANY KEY TO GOTO MENU"
87Ø A$ = INKEY$
     : IF A$ = "" THEN GOTO 87Ø ELSE GOTO 6Ø
```

Figure 2.

When you enter BASIC, make sure that you leave the memory above EF00 reserved for the machine language program. Do this by typing in the following:

```
BASIC -M:61184
```

when entering BASIC.

Next load and define the subroutine with the following:

```
SYSTEM "LOAD DIGIT"
DEFUSRØ = &HEFØØ
```

To use the subroutine, the command

```
II(Ø)=USRØ(VARPTR(II(Ø)))
```

is used. The coordinates of the axis will return in II%(0) and II%(1).

USR Functions from FORTRAN

Prof. John C. Turner
Department of Mathematics
U.S. Naval Academy
Annapolis, MD 21402

Although programs written in FORTRAN are quite fast, due to their being compiled, there are times when I have wished I had something analogous to the USR function in BASIC. However, to interface assembly language routines to FORTRAN requires the macro assembler which is rather more expensive than the Series I assembler. This is because FORTRAN and the macro assembler produce relocatable code that is linked together by the linker. The format of the files produced by both of these is much more complex than that produced by the Series I assembler. In fact, this complexity just about rules out the first option. That is to build the necessary relocatable file (/REL) file by hand. The format of /REL files is maddeningly complex because they are composed of a bit stream, instead of a byte stream. As a result, little of the "data" (object code) resides on byte boundaries and, in fact, the offset from a byte boundary changes throughout the file depending on what type of code is in the file. Some codes are 1 bit off from a byte boundary, some are 2 bits off, etc. The format of the /REL files is described in the FORTRAN manual on the last two pages. A glance at this documentation will show what you are up against if you try to hand build a /REL file. However, it occurred to me that it is not necessary to build a different /REL file for each USR routine. In fact, one very simple /REL file would do for all USR routines. First, it is necessary to know how FORTRAN passes its arguments. Before calling a subroutine, the address of the first argument is placed in the HL register pair. DE contains the address of the second argument. BC either contains the address of the third argument or the address of a table of the addresses of the rest of the arguments, if there are more than three. Note that everything involves addresses, not values. Suppose that we had an array, A, which contained code to be executed. Then calling a subroutine USR with A as the first argument would cause us to enter USR with the address of A in HL. Then suppose the USR routine merely jumped to A. If A contains a RET instruction, then the return would be to the original calling routine. The effect is that we can call machine language routines from FORTRAN. Further, all such calls can be made from a single common routine. On top of all of this, the code to be executed is data in a FORTRAN program. Thus, we can use FORTRAN subroutines to write machine language subroutines and then execute them!

But, as the Cat in the Hat would say, "That's not all, oh no, that's not all." The routine for which we must construct a /REL file consists of only one instruction, namely, JP (HL). The op code for this instruction is E9. Now, consider the simplest of FORTRAN subroutines.

```
SUBROUTINE USR
RETURN
END
```

This routine will generate only one machine language instruction, a RET. The op code for RET is C9. The difference between this routine and the one we need is only one bit! The trick is to know which bit in the /REL needs to be changed. To do this, I wrote a BASIC program which reads a /REL file and picks out the bit stream and prints it with a little description. Doing this, I was able to locate the bit that had to be changed to convert the USR program to the one we want. Listing 1 shows the BASIC program I used to write the required /REL file.

Listing 1

```
1Ø ' 7.2Ø.82 CREATE USR/REL FILE
2Ø CLEAR 1ØØØØ
3Ø DEFINT A-Z
4Ø OPEN "R",1,"USR/REL"
5Ø FIELD 1,255 AS R$
6Ø AØ=ASC("Ø")
7Ø AA=ASC("A")
8Ø W$=""
9Ø '-------- MAIN LOOP --------------
1ØØ ' READ IN BYTE IN HEX AND APPEND TO W$
11Ø ' STOP WHEN '-' IS READ IN
12Ø READ A$
13Ø IF A$="-" THEN 22Ø
14Ø A=ASC(A$)
15Ø IF A < AA THEN V1 = A - AØ ELSE V1 = A - AA + 1Ø
16Ø A$ = RIGHT$(A$,1)
17Ø A = ASC(A$)
18Ø IF A < AA THEN V2 = A - AØ ELSE V2 = A - AA + 1Ø
19Ø V = 16 * V1 + V2
2ØØ W$ = W$ + CHR$(V)
```

```
21Ø GOTO 12Ø
22Ø LSET R$ = W$
23Ø PUT 1,1
24Ø END
25Ø DATA 84, D5, 54, D4, AØ, 35, 55, 35, 28, 79, 19,
     3D, 49, 31, 25, Ø9
26Ø DATA D3, 28, Ø1, ØØ, 9A, 8Ø, 8Ø, 47, 4Ø, ØØ, 1A,
     AA, 9A, 94, EØ, ØØ, ØØ, 9E, ØØ
27Ø DATA "-"
```

In general, the usage of the USR routine is of the form CALL USR (A1,A2,A3, . . .). The first argument, A1, is the array that contains the code to be executed. I wrote a routine, HEX, to convert strings of hexadecimal digits into an array of type BYTE. This allows the program to contain a somewhat readable listing of the machine language routine that will be executed. As mentioned earlier, the DE register pair will contain the address of argument A2. If there are three arguments, BC will contain the address of argument A3. If there are more than three arguments, BC contains the address of a table of addresses of the rest of the arguments. All of this is an improvement over the USR routines in BASIC, where it is really only feasible to pass one argument to the routine.

There are some limitations to the kind of code that can be included in the USR routine. First, it must be totally relocatable, since you have no control over where it will be loaded. (After all, it resides in the FORTRAN data area.) This means that all jumps must be relative, except for jumps to known locations, like returning to TRSDOS at address 402DH. This also means that data storage within the routine must be accomplished using arguments. However, this is not much of a restriction, since the USR routine is only intended for simple routines. If you have something complicated, it is probably better written in FORTRAN.

As an example of a use of the USR routine, I wrote a machine language program to cause the BREAK key to cause a FORTRAN program to halt, regardless of what else is going on. As FORTRAN users know, the BREAK key is not special in FORTRAN as it is in BASIC. In FORTRAN, if you get into an endless loop, the only way to stop it is to press RESET.

This is a rather abrupt way of killing a program. The program I wrote for a Model III uses the real time clock interrupt to cause a scan for the BREAK key. If it is pressed, the routine jumps to TRSDOS. The real time clock causes an interrupt that ultimately jumps to 4046H. Normally this location contains a jump to somewhere else to do things like update the time, etc. My program copies this last jump address into the machine language routine. Then it overwrites the jump address with the address of the ML routine. The ML routine uses a call to 028DH, which is a quick scan for the BREAK key. It returns with a non-zero status if the BREAK key is pressed. My routine jumps to TRSDOS on this condition. Otherwise (if the BREAK key was not pressed), the routine jumps to the normal interrupt service routine whose address used to be in location 4046H. Thus, the BREAK key is scanned 30 times a second, allowing the user to abort any program that has set up this routine. There is one problem and that is that when the jump to TRSDOS occurs, the BREAK key will be down. TRSDOS interprets this to mean that it should come up as a cassette system. However, at this point, one can reboot from the disk.

In conclusion, it is possible to call machine language routines from FORTRAN without using the macro assembler. In fact, a single routine suffices for all USR routines. The /REL file for this routine can be created using the BASIC program supplied. Now all you machine language hackers can cause problems from FORTRAN, as well as BASIC.

```
      SUBROUTINE BRK
C  THIS ROUTINE ENABLES THE BREAK KEY TO INTERRUPT A
C  FORTRAN PROGRAM.
C  THERE ARE TWO MACHINE LANGUAGE PROGRAMS INVOLVED:
C     ONE IS THE INTERRUPT SERVICE ROUTINE
C     THE OTHER LOADS THE ADDRESS OF THE FIRST ROUTINE
C     INTO THE INTERRUPT VECTOR AT 4Ø46H
C
      BYTE A(12),B(6)
      N=1
      CALL HEX(A,N,'F5CD8DØ2C22D4ØF1C3-')
C  SUBROUTINE HEX COPIES A HEX STRING INTO ARRAY A,
C  STARTING AT LOCATION N.
C
C  THIS ROUTINE IS THE INTERRUPT ROUTINE
C        OP CODE   ASSEMBLY
C        F5        PUSH AF
C        CD8DØ2    CALL 8DØ2H   ;SCAN FOR BREAK KEY
C        C22D4Ø    JP NZ,4Ø2DH  ;IF SO, JUMP TO TRSDOS
C        F1        POP AF
C        C3????    JP ????      ;JP TO OLD SERVICE ROUTINE
C
      A(N)=PEEK(X'4Ø47')
C  COPY LOW BYTE OF OLD ISR ADDRESS
      A(N+1)=PEEK(X'4Ø48')
C  HIGH BYTE OF OLD ISR ADDRESS
C  NOW THE ?'S IN THE JP ARE FILLED IN
      CALL HEX(B,N,'ED534740C9-')
      CALL USR(B,A)
C
C  THIS ROUTINE ONLY HAS TWO INSTRUCTIONS
C  THE FIRST (ED534740) LOADS LOCATION 4Ø47H WITH DE
C  THE SECOND IS A RETURN (C9).
C  NOTE THAT ARRAY A IS THE SECOND ARGUMENT IN THE CALL
C  OF B.
C  THUS, ITS ADDRESS WILL BE IN REGISTER DE
C  SO THE OLD JUMP VECTOR HAS BEEN REPLACED WITH
C  THE ADDRESS OF ROUTINE A.
      RETURN
      END
      SUBROUTINE HEX(A,N,B)
C  THIS ROUTINE TAKES A HEX STRING IN B AND PUTS
C  THE CORRESPONDING VALUES INTO A, STARTING AT N.
C  ON RETURN, N IS UPDATED TO THE NEXT AVAILABLE LOCATION
C  IN A TO FACILITATE MULTIPLE CALLS TO HEX
C  THE CONVERSION STOPS WHEN A CHARACTER IN B IS FOUND
C  THAT IS NOT A VALID HEX DIGIT
      BYTE A(N),B(N)
      IØ=48
      I9=57
      IA=65
      IF=7Ø
C  DECIMAL EQUIVALENTS OF 'Ø', '9', 'A' AND 'F'
      I=1
      A(N)=Ø
5     CONTINUE
      IF((B(I).LT.IØ).OR.(B(I).GT.IF)) RETURN
      IF((B(I).GT.I9).AND.(B(I).LT.IA)) RETURN
      I1=B(I)-IØ
      IF(I1.GT.9) I1=B(I)-IA+1Ø
      I=I+1
      A(N)=16*A(N)+I1
      IF((I.AND.1).EQ.Ø) GOTO 5
      N=N+1
      A(N)=Ø
      GOTO 5
      END
```

High Res Screen Dump for the CGP-115

David Andrew Palmer
P.O. Box 814
Deep River, Ontario
K0J-1P0

Having just bought my Color Graphics Printer, I realized that it had one minor disadvantage. As yet, it did not have a high-resolution screen copy program available for it like the Line Printer VIIs. That is why I have written this program.

This program will copy any high-resolution screen in one of the Color Computer's two-color modes. It will copy in any one of the four colors available on the CGP-115. The program offers two print densities, two dots per pixel and four dots per pixel.

The average drawing time in the low-density mode is about an hour and a half, but the time varies with the number of dots to be printed (anywhere from half an hour with few set pixels to four hours with many set pixels.

To copy a screen to the CGP-115, the following procedure must be followed:

1. If the screen is already in graphics memory, just ⟨CLOAD⟩ the program and ⟨RUN⟩ it. (Be sure that the printer is powered up and operational.)
2. If you want to put a copy of the graphics which are in memory onto tape, enter CSAVEM"SCREEN", 1536, 7679, 0. To retrieve this copy, enter CLOADM"SCREEN".
3. If you have a screen already on tape (saved in the above format), then RUN this program and enter ⟨Y⟩ for the CLOADM prompt. This will take a high-resolution screen off the cassette and put it out to the printer.

Be sure to have well running pens in the printer, or the copy of the screen will appear somewhat streaked.

You should be aware that this program will neglect (ignore) the rightmost 15 pixels of the screen. This is due to the paper width in the CGP-115.

This program scans the high-resolution screen until it hits a set pixel (in COLOR 0,5) and then moves the printer to the appropriate position. The printer then places a dot at that position and the program searches for the next set pixel.

If you have a drawing that does not use COLOR 0,5, change the "5" in line 410 to the background color you are using (e.g. with COLOR 1,0 "0" would be the background color.)

Other users of this program might want to change it to give a color copy of a four-color screen, or change it to work on other plotters.

```
0 REM SCREEN COPY PROGRAM
1 REM BY DAVID ANDREW PALMER, P.O. BOX 814, DEEP
     RIVER, ONTARIO, K0J-1P0
2 REM FOR USE WITH EXTENDED COLOR BASIC AND CGP-115
10 PMODE 4, 1
15 COLOR 0, 5
90 CLS
100 INPUT "DENSITY (0-1)"; S
110 INPUT "CLOADM DRAWING (Y/N)"; A$
120 IF A$="Y" THEN PRINT "INSERT CASSETTE, PRESS
     PLAY, PRESS ENTER"
     : LINE INPUT QQ$
     : SCREEN 1, 1
     : CLOADM
130 INPUT "COLOR (0-3)"; C
180 SCREEN 1, 1
200 '
202 PRINT#-2, CHR$(17)
205 PRINT#-2, CHR$(18)
210 PRINT#-2, "I"
215 PRINT#-2, "C"; C
220 FOR Y=0 TO 192
250 FOR X=0 TO 255
300 P=PPOINT(X, Y)
350 PSET(X, Y, P+1)
     : PSET(X, Y, P)
405 A=0
410 IF P=5 THEN A=1
415 IF A=1 THEN 600
420 PRINT#-2, "M"; X*2 ","; Y*-2
500 IF S=1 THEN GOSUB 10000 ELSE GOSUB 11000
600 NEXT X, Y
9999 END
10000 PRINT#-2, "J2, 0, 0, -2, -2, 0, 0, 2"
10010 RETURN
11000 PRINT#-2, "J1, 0"
     PRINT#-2, "R-1, 0"
11010 RETURN
20000 END
```

The Sands of Egypt

Review by Bruce Elliott

The Sands of Egypt (26-3290 Suggested Retail $29.95) is the first of what I hope are many animated adventure games for disk-based TRS-80 Color Computers.

The plot is fairly simple, you are lost in the desert and your goal is to find the hidden treasure in as few moves as possible, and then escape.

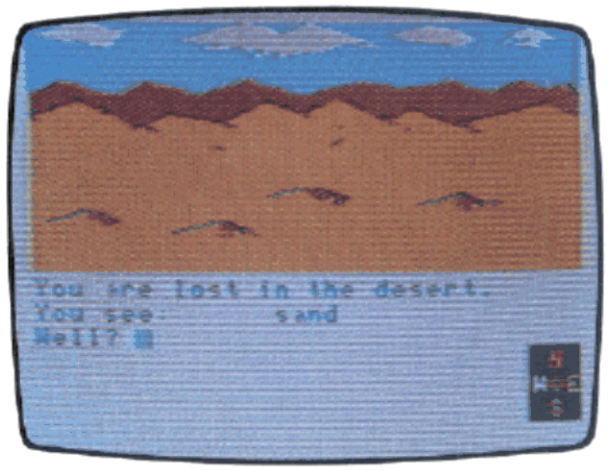
The manual which comes with the program lists required equipment as:

16K Extended Color BASIC TRS-80 Color Computer
Color Computer Disk Drive and Cable

The manual also lists as "optional" equipment:

map-making equipment (paper and pencil)

If you are new to adventures, I suggest that the map-making equipment is vital to your long term sanity and survival.



Scene 1.

THE LONG TREK

Come along with me as I set out to find the treasure. The path I will show you is the path I normally take. It may not be the quickest way, and I may not have found everything I need to complete the adventure. It is simply the path I have found to the treasure (and I am not yet sure if the treasure is the one I am supposed to find.) Honor, duty, and the spirit of the game prohibit me from revealing HOW I got to each of these from scene 1, and ignorance prevents me from showing you what may be beyond. At the time I am writing this article, I have not been able to get further than the SECRET chamber.

Scene 2.

Scene one is where it all begins. If you check your INVEntory, you find that you are carrying nothing. You should notice however that a compass is provided (right side of the screen.) The compass indicates that I may proceed North, East, South or West. After some wandering, which hopefully gets less aimless as time goes by, I arrive at scene two.

Scene two brings me to the top of a cliff. If you are making a map, you might want to re-enter the game several times to find the shortest route to the cliff. Since this is the first non-desert scene I encountered, finding the shortest distance here gives me something to judge other distances by. From the top of the cliff, I continue wandering (notice the compass) until I get to scene three. Along the way, I have picked up an object which seems like it could be useful in the desert.

A snake. Some probing, and a couple of deaths (mine) tell me that the snake will let me retreat (South or West), will not let me go East, and follows me if I try to go North (follows me? How do I show that on a map? Is it important?)

My feelings tell me that if the snake won't let me go East, then I probably want to go East. I may ALSO need to go North or West, but I'm sure I have to go East.

How do I get around the snake? Several options. In a real desert I might be able to walk around the snake (i.e., go South or West and then come back from another direction), or I might be able to kill the snake. After several more deaths, and getting lost several times, I end up at scene four.

Scene 4.

A canteen! Since I have been wandering in the desert, I am naturally thirsty. Fortunately (?) the computer knows that I am thirsty and tells me so on the screen. Is the canteen full? If it is, is the liquid poison? Will it quench my thirst? If it isn't full, where do I find water in the middle of a desert? So, more wandering, a couple of deaths from thirst, and finally scene five:

Scene 3.

Scene 5.

Water, at last. Fill the canteen, and drink—right? well, not quite, but you'll figure it out. A look around the pool shows something interesting. The text tells me that I can see a pool, camel, and pyramid but says nothing about the trees. Why not? More wandering, a quick call to the Color SIG on CompuServe, and I know how to get to the tree.

As with most things in an adventure, the path to the trees was quite simple, I had just not been creative enough. I felt all along that I had to get to the trees since the camel looked hungry. The trees were the only possible source of food that I had seen. If you are sure you have to get somewhere, and there doesn't seem to be any way, try everything. Use every verb you can think of, no matter how silly. You might also try asking yourself questions. What is it that you want to do? How can you express that idea in two words, a verb and an object.

Scene 6.

I wonder if the dates are good to eat? It seems that HELP mentioned food at least once, what was it? Also, it seems that I have come across palm fronds before, it seemed important at the time. Oh well, back to earth and scene seven:

Scene 7.

What a lovely animal. How did I get up here? Where is the accelerator pedal? How do I get this beast to move? Surely it is important, or they wouldn't have put it in the adventure.

If you haven't done so already, this might be a good time to SAVE the current status of the adventure. SAVE and LOAD can save you a lot of time and frustration as you try to get back to a known point to resume your trek.

Much wandering, many discoveries, and a few deaths later, I finally stand inside the SECRET chamber. This is a good time to mention the HELP feature that seems to be in most adventures. If you get in a bind, type HELP. You may (or may not) get some information which will guide you. In Sands of Egypt, I have found that the information provided is generally useful. Unfortunately, it is not always helpful in the immediate situation. When asking for help I was often told that this was a "DRAINING EXPERIENCE." Standing in the middle of the desert, dying of thirst, that was not exactly what I wanted to read. However, the information ultimately proved a valuable clue.

Scene 8.

The compass is worthless, and I'm getting thirsty. Darn those piranha!

Looks like it is just a matter of time until I am forced to drink, and once again be dead:

Scene 9.

Oh well, that's what adventures are all about I guess. Just one more attempt to get the treasure out and I'll go home . . .

Radio Shack Computer Center Addresses and Telephone Numbers

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

ALABAMA

BIRMINGHAM 2428 Green Springs Hwy., (205) 945-0792
HUNTSVILLE 1400 N. Memorial Pkwy., (205) 536-1581
MOBILE 405 Bel-Air Blvd., (205) 471-1617
MONTGOMERY #24 Union Square S/C, (205) 271-1500

ARIZONA

PHOENIX 10233 Metro Pkwy. E., (602) 861-1124; 4301 N. 7th Ave., (602) 277-3031
SCOTTSDALE 2525 N. Scottsdale Rd., (602) 990-2241
TEMPE 83 E. Broadway, (602) 894-2065
TUCSON 5622 E. Broadway, (602) 748-0101; Campbell Plaza, 2830 N. Campbell Ave., (602) 881-8003

ARKANSAS

LITTLE ROCK Town & Country S/C, University & Asher, (501) 568-5694

CALIFORNIA

ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
BREA Imperial Shopping Center, 391 South State College, (714) 671-1681
CANOGA PARK 8371 Topanga Canyon, (213) 347-9800
CARMICHAEL 6305 Fairoaks Blvd., (916) 484-6815
CHICO 1834 Mangrove Ave., (916) 894-0101
CHULA VISTA 1201 3rd Ave., (714) 420-3810
CITRUS HEIGHTS 7405 Green Back Ln. at San Juan, (916) 723-5183, 969-8752
DOWNEY 8031 Florence Ave., (213) 927-7882
EL TORO El Toro Rd. at San Diego Freeway, (714) 859-4411
ESCONDIDO 347 W. Mission Ave., (714) 741-6032
FREMONT Fremont Hub, 39114 Fremont, (415) 790-1112
FRESNO Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551
GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 24784 Hesperian Blvd., (415) 887-6420
HOLLYWOOD 6922 Hollywood Blvd., (213) 466-8681
IRVINE Redhill at Main St., (714) 660-0299
LAKEWOOD 5830 Lakewood Blvd., (213) 920-9671
LA MESA 5346 Jackson Dr., (714) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 629-2455; 5240 Century Blvd. (Airport area), (213) 641-4585
MODESTO 221 McHenry Ave., (209) 521-1011
MONTCLAIR 5237 Arrow Hwy., (714) 625-1541
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1933 El Camino Real W., (415) 961-0542
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
PLEASANT HILL 508 G. Contra Costa Blvd., (415) 676-9866
RIVERSIDE 3844 La Sierra Ave., (714) 689-0340
SACRAMENTO 4749 J. St., (916) 454-3287
SAN BERNARDINO 764 Inland Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairemont Dr., (714) 276-6050; 3902 El Cajon Blvd., (714) 280-0227
SAN FRANCISCO One Market Place, (415) 777-9810; 2920 Geary Blvd., (415) 221-8716
SAN JOSE 1228 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA ANA 2320 S. Fairview St. (at Warner), (714) 966-2787
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702
SANTA MONICA 511 Wilshire Blvd., (213) 458-1802
SANTA ROSA 823 4th St., (707) 542-0402
SHERMAN OAKS 14936 Ventura Blvd., (213) 788-2639
STOCKTON College Sq. S/C, 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3840 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

COLORADO

BOULDER Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
COLORADO SPRINGS 4341 N. Academy, (303) 593-7500
DENVER 8000 E. Quincy, (303) 770-1362; Green Briar Plaza, 7075 Pecos, (303) 428-6787
LAKEWOOD 2099 Wadsworth Blvd., (303) 232-6277

CONNECTICUT

EAST HAVEN 51 Frontage Rd., (203) 467-8864
FAIRFIELD 1196 Kings Hwy. & Rt. 1, (203) 255-6099
HARTFORD The Richardson Bldg., 942 Main St., (203) 549-5717
MANCHESTER 228 Spencer St., (203) 649-8210
NORWALK Rt. 7-345 Main Ave., (203) 846-3418
ORANGE Woolco S/C; 538 Boston Post Rd., (203) 795-1291
WATERBURY 105 Bank St., (203) 573-8800
WATERFORD 122 Boston Port Rd., (203) 443-0716
WEST HARTFORD 39 S. Main St., (203) 523-4283

DELAWARE

DOVER Edgehill Shop. Ctr., Rt. #13, (302) 734-5223
WILMINGTON 3847 Kirkwood Hwy., (302) 999-0193

DISTRICT OF COLUMBIA

WASHINGTON 1800 M St. NW., (202) 822-3933

FLORIDA

ALTAMONTE SPRINGS 766 B. East Altamonte Dr., (305) 339-1313
BOCA RATON 1662 N. Federal Hwy., (305) 391-3308
CLEARWATER 2460-D US 19 North, (813) 797-3223
DAYTONA BEACH Volusia Plaza, (904) 255-5909
FT. LAUDERDALE 4368 N. Federal Hwy., (305) 772-6797
FORT MYERS Edison Mall, (813) 939-3454
FORT PIERCE Center West Shopping Center, U.S. Hwy. 1, (305) 466-2200
GAINESVILLE 3315 Archer Rd., (904) 375-7063
HOLLYWOOD 429 S. State Rd. #7, (305) 966-4382
JACKSONVILLE 8252 Arlington Expnwy., (904) 725-2594; Roosevelt Mall, Roosevelt Blvd., (904) 388-6611
LAUDERDALE LAKES 4317-25 N. State Rd. 7, (305) 486-2240
MIAMI 9459 S. Dixie Hwy., (305) 667-2316; 1601 Biscayne Blvd., (305) 374-6433; 15 SE. 2nd Ave., (305) 374-7310; 20761 S. Dixie Hwy., (305) 238-2518
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St., (305) 940-6887
ORLANDO 1238 E. Colonial Dr., (305) 894-0570
ST. PETERSBURG 3451 66th St. N., (813) 381-2366
SARASOTA 5251 S. Tamiami Tr. (Hwy.41), (813) 923-4721
TALLAHASSEE 2529 S. Adams, (904) 222-4440
TAMPA 4555 W. Kennedy, (813) 879-7470; 1825 E. Fowler Ave., (813) 971-1130
W. PALM BEACH 2271-A Palm Beach Lakes Blvd., (305) 683-3100

GEORGIA

AUGUSTA 3435 Wrightsboro Rd., (404) 738-5998
ATLANTA 2108 Henderson Mill, (404) 939-9888; 49 W. Paces Ferry, (404) 231-9604; Akers Mill S/C, 2937 Cobb Parkway NW., (404) 955-5235; 113 Peachtree St., (404) 223-5904
COLLEGE PARK 5309 Old National Hwy., (404) 761-3055
COLUMBUS Columbus Square Mall, 3050 Macon Rd., (404) 568-7200
DORAVILLE 5697 Buford Hwy., (404) 458-2691
SAVANNAH Chatham Plaza, 7805 Abercorn St., (912) 355-6074

IDAHO

BOISE 691 S. Capitol Blvd., (208) 344-5450

ILLINOIS

AURORA 890 North Lake St., (312) 844-2224
CHICAGO 4355 S. Archer Ave., (312) 376-7617; CNA Plaza, 309 S. Wabash, (312) 922-0536; 72 West Adams, (312) 236-7262; 524 N. Michigan Ave., (312) 822-0533; 101 W. Washington St. at Clark, (312) 782-7207
ELMWOOD PARK 7212 W. Grand Ave., (312) 452-7500
FAIRVIEW HEIGHTS #4 Market Place, (618) 398-6410
HOMEWOOD/GLENWOOD 329 Glenwood Lansing, (312) 758-0440
JOLIET 2415 W. Jefferson St., (815) 744-7080
LaGRANGE One S. LaGrange Rd., (312) 482-3484
LIBERTYVILLE 1350 S. Milwaukee Ave., (312) 367-8230
LOMBARD 4 Yorktown Center, (312) 629-5350
NILES 8349 Golf Rd., (312) 470-0670
OAK LAWN 4815 W. 95th St., (312) 425-9130
PEORIA 4125 N. Sheridan Rd., (309) 685-7056
ROCKFORD North Town S/C, 3600 N. Main St., (815) 282-1001
SCHAUMBURG 651 Mall Dr., (312) 884-8600
SPRINGFIELD Sherwood Plaza, 2478 Wabash, (217) 787-3066

INDIANA

EVANSVILLE 431 Diamond Ave., (812) 426-1715
FT. WAYNE 747 Northcrest S/C, (219) 482-9547
GRIFFITH 208 W. Ridge Rd., (219) 838-3000
HOBART Save-More Plaza, Rtes. #6 & 51, (219) 942-0541
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., (317) 849-6896; Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221; 10013 E. Washington St., (317) 898-4887; Two W. Washington St., (317) 236-9991
SOUTH BEND 1827 South Bend Ave., (219) 277-3789
TERRE HAUTE 3460 U.S. Hwy. 41 S., (812) 234-3212

IOWA

CEDAR RAPIDS 111 First Ave., S.E. (Downtown), (319) 362-7399
DAVENPORT 616 E. Kimberly Rd., (319) 386-3457
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M, (515) 270-0193

KANSAS

OVERLAND PARK 8619 W. 95th, (913) 642-1301
TOPEKA White Lakes Plaza, West Tower, 3715 Plaza Dr., (913) 267-6420
WICHITA 2732 Blvd. Plaza S/C, (316) 681-1212

KENTUCKY

FLORENCE 7727 Mall Rd., (606) 371-2811
LEXINGTON 2909 Richmond Rd., (606) 269-7321
LOUISVILLE 2900 Taylorsville Rd., (502) 459-9901; Louisville Galleria, (502) 589-1013

LOUISIANA

BATON ROUGE 7007 Florida Blvd., (504) 928-5260
HOUMA 2348 W. Park Ave. (Hwy. 24), (504) 873-7713
LAFAYETTE University Square at Congress Blvd., (318) 235-6177
METAIRIE 3750 Veterans Hwy., (504) 454-3681
NEW ORLEANS 327 St. Charles Ave., (504) 523-6408
SHREVEPORT 1545 Line Ave., (318) 221-5125

MAINE

BANGOR Maine Square, (207) 945-6491

MARYLAND

BALTIMORE 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583; 115 N. Charles St. at Lexington, (301) 539-7251
BETHESDA 7900 Wisconsin Ave., (301) 656-0892
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike, (301) 788-3277
FREDERICK Shoppers World, Rt. 40W, (301) 695-8440
NEW CARROLLTON-LANHAM 7949 Annapolis Rd., (301) 459-8030
PASADENA 8120 Ritchie Hwy., (301) 544-2352
ROCKVILLE Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
SALISBURY Shoppers World S/C, Rt. 50, (301) 546-9223
TEMPLE HILLS 4520 St. Barnabas Rd., (301) 899-0740
TOWSON-LUTHERVILLE Yorktown S/C York Rd. at Ridgley Rd., (301) 561-2001

MASSACHUSETTS

BOSTON 730 Commonwealth Ave., (617) 739-1704; 111 Summer St., (617) 542-0361
BRAINTREE South Shore Plaza, 250 Granite St., (617) 848-9290
BROCKTON 675 Belmont, (617) 583-2270
BURLINGTON Crossroads Plaza, Rt. 3 S., (617) 229-2850
CAMBRIDGE Harvard Square, 28 Boylston St., (617) 354-7694
CHESTNUT HILL 200 Boylston St., (617) 969-2031
NATICK 1400 Worcester Rd., (617) 875-8721
SAUGUS 343 Broadway, (617) 233-4985
SPRINGFIELD 1985 Main St., Northgate Plz., (413) 732-4745
WORCESTER Lincoln Plaza, (617) 852-8844

MICHIGAN

ANN ARBOR 2515 Jackson Rd., (313) 761-6266
BIRMINGHAM 3620 W. Maple Rd., (313) 647-2151
DEARBORN Westborn Shop. Ctr., 23161 Michigan Ave., (313) 274-6161
DETROIT DWNTN 1559 Woodward Ave., (313) 961-6855
FLINT G3298 Miller Rd., Yorkshire Plaza, (313) 732-2530
GRAND RAPIDS 3142 28th St. SE., (616) 957-2040
KALAMAZOO 25 Kalamazoo Center, (616) 343-0780
LANSING 2519 S. Cedar St., (517) 372-1120
PLAINFIELD North Kent Mall, (616) 364-7558
LIVONIA 33470 W. 7 Mile Rd., (313) 476-6800
ROSEVILLE 31873 Gratiot Ave., (313) 296-6210
SOUTHFIELD 17651 West 12 Mile Rd., (313) 569-1027
TROY Oakland Plaza, 322 John R. Rd., (313) 585-3900
WARREN 29038 Van Dyke Ave., (313) 573-3870

MINNESOTA

BLOOMINGTON 10566 France Ave. S., (612) 884-1641
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy., (612) 542-8471
ST. PAUL 6th & Wabasha, (612) 291-7230

MISSISSIPPI

JACKSON 979 Ellis Ave., (601) 352-5001

MISSOURI

DES-PERES 11960 Manchester Rd., (314) 965-5911
FLORISSANT 47 Florissant Oaks S/C, (314) 921-7722
INDEPENDENCE 1325 S. Noland Rd., (816) 254-3701
KANSAS CITY 4025 N. Oak Trafficway, (816) 455-3381
ST. ANN 10472 St. Charles Rock Rd., (314) 428-1400
SPRINGFIELD 2684 S. Glenstone, (417) 883-4320

NEBRASKA

OMAHA 3006 Dodge St., (402) 346-4003

NEVADA

LAS VEGAS Commercial Center, 953 E. Sahara #31-B, (702) 731-3956
RENO 3328 Kietzke Lane, (702) 826-6327

NEW HAMPSHIRE

MANCHESTER Hampshire Plaza, 1000 Elm St., (603) 625-4040
NASHUA 429 Amherst St., Rt. 101A, (603) 881-8588

NEW JERSEY

BRIDGEWATER 1472 U.S. Highway 22 East, (201) 469-3232
E. BRUNSWICK 595 A Rt. 18, (201) 238-7142
E. HANOVER Rt. 10, Hanover Plaza, (201) 884-1200
LAWRENCEVILLE Rt. 1 & Texas Ave., (609) 771-8113
NEWARK 595 Broad, (201) 622-1339
NORTHFIELD 322-24 Tilton Rd., (609) 645-7676
PARAMUS 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rt. #22 Center Isle, (201) 467-9827
VOORHEES 35 Eagle Plaza, (609) 346-0600

NEW MEXICO

ALBUQUERQUE 2108 San Mateo NE., (505) 265-9587

NEW YORK

ALBANY Shoppers Pk., Wolf Rd., (518) 459-5527
BAYSHORE 1751 Sunrise Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-6403
BROOKLYN 531 86th St., (212) 238-3576
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
FRESH MEADOWS 187-12 Horace Harding Exp., (212) 454-1075
GARDEN CITY 960 Franklin Ave., (516) 294-3350
JOHNSON CITY Giant Shopping Center, Harry L. Drive, (607) 729-6312
KINGSTON Kings Mall, Rt. 9W, (914) 336-6262
MELVILLE TSS Mall, Rt. 110, (516) 673-4646
NEWBURGH Zayre Plaza, Rt. #17K, (914) 561-2960
NEW ROCHELLE 242 North Ave., (914) 636-0700
NEW YORK 385 Fifth Ave., (212) 889-1345; 139 E. 42nd St., (212) 953-6053; 19 W. 23rd St., (212) 691-1861; 347 Madison Ave., (212) 867-8650; 270 Park Ave. South, (212) 533-0470; 1282 Broadway, (212) 279-9260; 9 Broadway, (212) 482-8200
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd., (716) 283-2041
REGO PARK 97-77 Queens Blvd., (212) 897-5200
ROCHESTER 3000 Winton Rd., (716) 244-6400
SCARSDALE 365 Central Park Ave., (914) 472-2520
SPRING VALLEY White House Center, 88 W. Rt. 59, (914) 425-2828
STATEN ISLAND 2409 Richmond Ave., (212) 698-3100
SYRACUSE 2544 Erie Blvd., (315) 446-3017; Hotel Syracuse, 510 S. Warren St., (315) 471-6663
UTICA Riverside Mall, (315) 735-1933
VALLEY STREAM Green Acres Shop. Ctr., (516) 872-6460

NORTH CAROLINA

CHARLOTTE 3732 Independence Blvd., (704) 535-6320; Tyvola Mall, 5401 South Blvd., (704) 527-3373
DURHAM South Square Mall, (919) 493-4025
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St., (919) 483-2418
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townridge Sq., Hwy. 70 W., (919) 781-9380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0030

OHIO

AKRON Fairlawn Plaza, 2727 W. Market St., (216) 836-9303
BEDFORD HEIGHTS 5217 Northfield Rd., (216) 662-2477
CANTON 5248 Dressler Rd. NW., (216) 494-7230; Mellet Plaza, 3826 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 793-8688; 16-18 Convention Way (on Skywalk), (513) 381-4664
CLEVELAND 419 Euclid (Dwntwn), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2806; The Patio Shop. Ctr., 4661 Karl Rd., (614) 436-4666; 400 N. High St., (614) 464-2781
DAYTON Northwest Plaza, 3279 West Siebenthaler, (513) 277-6500
FAIRFIELD 7255 Dixie Hwy. (1/4 Mi. North of I-275), (513) 874-5994
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 7551 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Central Ave., (419) 531-5797
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

OKLAHOMA

OKLAHOMA CITY 4732 SE 29th St., (405) 670-4561; Springdale S/C, 4469 NW 50th, (405) 943-8712; 1101 SW 59th St., (405) 634-2406
TULSA 7218 & 7220 E. 41st St., (918) 663-2190

OREGON

EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157; 9131 SE Powell, (503) 777-2223; 3rd and Washington Sts. (Downtown), (503) 241-7024
SALEM Salem Plaza, 403 Center, (503) 588-7095

PENNSYLVANIA

ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 395-7155
BALA CYNWYD 67 E. City Line Ave., (215) 668-9950
EASTON 25th St. Shopping Center, (215) 250-0160
ELKINS PARK Elkins Park Square, 808 Old York Rd., (215) 645-5460
ERIE 5755 Peach St., (814) 868-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
LANCASTER Park City Plaza, US 30, (717) 393-5817
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 362-1200
PHILADELPHIA 7542 Castor Ave., (215) 342-2217; 1002 Chestnut St., (215) 923-3080; 1801 Market St., 10 Penn Center, (215) 568-0901
PITTSBURGH 5775 Baptist Rd., Hills Plaza, (412) 831-9694; 303 Smithfield St., (412) 391-3150; 4643 Baum Blvd., (412) 681-4000; 4768 McKnight Rd., (412) 364-4342
SCRANTON 206 Meadow Ave., (717) 348-1801
WYOMISSING Berkshire Mall West, 1101 Woodland Rd., (215) 372-8610

PUERTO RICO

HATO REY 243 Franklin D. Roosevelt Ave., (809) 759-8248

RHODE ISLAND

E. PROVIDENCE 850 Waterman Ave., (401) 438-2860
PROVIDENCE 177 Union St., (401) 831-0320

SOUTH CAROLINA

COLUMBIA Old Sears Bldg., 1001 Harden St., (803) 799-2065
GREENVILLE N. Hills S/C, (803) 292-1835
N. CHARLESTON 5900 Rivers Ave., (803) 747-5580

TENNESSEE

CHATTANOOGA 636 Northgate Mall, (615) 870-1366
JOHNSON CITY Peerless Center, (615) 282-6829
KNOXVILLE Cedar Bluff S/C, 9123 Executive Park Dr., (615) 690-0520
MEMPHIS 4665 American Way, (901) 795-4963; 1997 Union Ave., (901) 278-7935
NASHVILLE 2115 Franklin Pike, (615) 298-5484; Rivergate Plaza, (615) 859-3414

TEXAS

AMARILLO Wellington Sq. S/C, 1619 S. Kentucky, (806) 358-4567
ARLINGTON 2500 E. Randol Mill, Suite 113, (817) 274-3127
AUSTIN 8764 E. Research Blvd., (512) 459-4238; Southwood Mall, 1501 Ben White Blvd., (512) 447-0371
BROWNSVILLE 1639 Price Rd. (Hwy. 77), (512) 544-6800
BEAUMONT 5330 Eastex Frwy., (713) 898-7000
COLLEGE STATION 2414 Texas Ave., South, (713) 764-8171
CORPUS CHRISTI 1711 S. Staple St., (512) 887-8901
DALLAS 15340 Dallas Pkwy., Suite 1100, (214) 934-0275; 2930 W. Northwest Hwy., (214) 350-4144; 1517 Main St., (214) 760-8601; 2588 Royal Ln., (214) 484-9947
EL PASO 9515 Gateway West, (915) 594-8211; Kern Plaza Shopping Center, 3100 N. Mesa, (915) 544-8156
FT. WORTH 231 One Tandy Center, (817) 335-7198; 2801 Alta Mere, (817) 738-0251
HARLINGEN 1514 S. Hwy 77, Sunshine Strip, (512) 425-8890
HOUSTON 211C FM 1960, (713) 444-7006; 10543 Gulf Fwy., (713) 943-9310; 5900 North Fwy., (713) 699-1932; 6813 SW Fwy., (713) 777-7907; 809 Dallas St., (713) 651-3002, Holland Square Center, 10920 East Freeway, (713) 453-0600; Champion Forest Plaza, Champion Forest Dr. and F.M. 1960 West, (713) 580-1404; 1018 Gessner, (713) 984-9600
HURST Northeast Mall, (817) 284-1518
LAREDO 102 East Calton Rd., (512) 727-4768
LUBBOCK 3625 34th St., (806) 793-1467
ODESSA 1613 "A" East 8th Street, (915) 334-8355
RICHARDSON Fleetwood Sq. S/C, 202 W. Campbell Rd., (214) 669-1494
SAN ANTONIO 6018 West Ave., (512) 344-8792; 4249 Centergate, (512) 657-3958; Riverbend Parking Garage, 211 E. Market St. at Presa, (512) 225-0345

UTAH

OGDEN K-Mart Shopping Center, 3672 Wall Ave., (801) 394-1666
OREM Grand Central Plaza, 384 East & 1300 South, (801) 226-8651
MURRAY 6051 S. State Ave., (801) 268-8978
SALT LAKE CITY 301 South State St., (801) 322-5526

VIRGINIA

ALEXANDRIA 3425 King St. at Quaker Ln., (703) 820-2600
ARLINGTON Crystal City, 2301 So. Jefferson Davis Hwy., (703) 521-7479
FAIRFAX Westfair Center, 11027 Lee Hwy., (703) 273-6500
LYNCHBURG Hill's Plaza, Ward's Rd., (804) 237-1664
NEWPORT NEWS Newmarket South Shop. Ctr., (804) 245-8203
NORFOLK 5731 Poplar Hall Dr., (804) 461-0798; Wards Corner, 122 E. Little Creek Rd., (804) 588-7548
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453; 7728 Midlothian Turnpike, (804) 272-8803
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5955

WASHINGTON

BELLEVUE Crossroads Mall, North East 8th & 156 St., (206) 644-1804
BELLINGHAM 1111 Cornwall Ave., Suite B & C, (206) 676-0811
FEDERAL WAY 33505 Pacific Hwy. South, (206) 838-6830
OLYMPIA 106 N. Wilson, (206) 754-6111
SEATTLE 18405 Aurora Ave. N., (206) 542-6184; 1521 3rd Ave., (206) 447-1959; 5030 Roosevelt Way NE, (206) 527-1019
SPOKANE 7702 N. Division, (509) 484-7000; E. 12412 Sprague (509) 922-2800
TACOMA 7030 S. Sprague, (206) 473-7333
TUKWILA 15425 53rd Ave. S., (206) 248-3710
YAKIMA 1111 N. First St., (509) 248-9667

WEST VIRGINIA

HUNTINGTON 2701½ 5th Ave., (304) 523-3527

WISCONSIN

APPLETON 2310 West College Ave., (414) 731-6424
MADISON 57 West Towne Mall, (608) 833-6130
MILWAUKEE 6450 N. 76th St., (414) 353-6790
WEST ALLIS 2717 South 108th St., (414) 327-4240