80micro

12/82

#35

Turn Your 80 into a Sound-Actuated Typewriter

December 1982
USA $2.95 (UK£2.00)

# 80micro

A WAYNE GREEN PUBLICATION

the magazine for TRS-80* users

## From Fahrenheit to Celsius— In Six Different Languages

## Reviews: Pascal 80 And Colorforth

## A Spooler for Your Model I

## The Little Old Map-Making 80

## And a Sackful Of Holiday Treats!

0 74470 65947 12

Introducing the MTI® line of computers. We believe in offering you the best in complete, low cost computing.

**MTI MOD III PLUS 140.** A personal computer that has been designed and priced for your business, your department...yourself. MTI Computers offer you many "upgrade" features as standard. All MTI Computers include a 65-key keyboard complete with a 12-key datapad, a large 12" monitor and anti-glare screen, RS-232 printer interface, cooling unit and more. The memory is 48K and has a Z80 processor and two double density disk drives.

**LOW PRICE.** MTI computers are priced from $1999.

**MOD III PLUS 140** Computer consists of 48K bytes of memory and a double density, dual disk drive system. The system is fully compatible with the Radio Shack® DOS and peripherals. Get 178,000 bytes of memory storage per disk.

**OPTIONS:**
- LETTER QUALITY PRINTER
- WINCHESTER CP/M 2.2

**OTHER MODELS AVAILABLE:**

**MOD III PLUS 240.** Same as the MOD III 140 but has double storage capacity and dual headed 40-track drives.

**MOD III PLUS 280.** Has 1.5 megabytes storage and 2 dual headed 80-track double density disk drives.

**MOD III/WINCHESTER.** Our largest computer system. 8.2 megabytes of storage. Includes a 7.5 megabyte WINCHESTER hard disk and 80-track dual head disk head.

# CP/M 64K

The Mod III Plus C/140 system is an exciting new personal and small business computer that is both CP/M & TRS-DOS compatible.

**■ FEATURES:**
- 80 x 24 Display for CP/M
- 64K Bytes RAM Memory
- Microsoft BASIC language in ROM
- Internal Z80 ROM Diagnostic
- Z80 processor, 2MHz
- 64 x 16 screen-TRS-DOS-,DOS+
- Cooling system
- Parallel printer interface included
- Anti-glare screen
- 2 internal 40tk Disk Drives
- Compatible with TRS-80® Model III—RS-232 serial interface option.

  Compatible with CP/M applications software such as Wordstar, Datastar and Mailmerge.

## $2699

- 90 Day Warranty
- One Year Extended Warranty Available
- Local Dealer Service
- TRS-80 Compatible


# MTI: YOUR BEST VALUE IN LOW COST COMPUTERS

## PRICES FROM $1999

**OVER 100 AUTHORIZED DEALERS**
**CALL 714-979-9923 for the dealer nearest you**

CP/M is a registered trademark of Digital Research, Inc.
TRS-80® is a registered trademark of Tandy Corp.
MTI is a registered trademark of Microcomputer Technology, Inc.

247

**MICROCOMPUTER TECHNOLOGY INC.**
3304 W. MACARTHUR, SANTA ANA, CA 92704
(714) 979-9923 • TWX 910-595-1902 MTISNA

U.S. PRICES F.O.B. SANTA ANA CALIFORNIA AND MAY VARY BY AREA

# 80 Contents

## Features


**72. Voice-Controlled Typewriter**
The physically handicapped can produce printed text with this sound-controlled program.
*Mike Rigsby*

**80. The Color Computer Goes Forth**
With this new package, Color Computer owners can explore programming in a new language—Forth.
*Scott L. Norman*

**150. Tracking Lobster**
Your TRS-80 can't bait a hook, but it can help you trap a lobster using maps and graphs.
*William F. Kaczor*



**210. Pascal-80**
Pascal-80 provides a serious version of the Pascal language to TRS-80 users. You might find its structured programming rewarding.
*John B. Harrell, III*

**216. TRS-80 Tongues**
Can't decide which language to use? Here's a simple temperature conversion program in six: Basic, Fortran, Cobol, Pascal, APL, and Forth.
*Robert Nicholas and Philip Martel*



*80 Micro* (ISSN 0744-7868) is published 12 times a year by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone 603-924-9471. Second class postage paid at Peterborough NH and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada and Mexico $27.97—one year only, U.S. funds drawn on a U.S. bank. Canadian distributor Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor Graymar Data Services, Ltd. #4 258 E. 1st Ave. Vancouver BC V5T 1A6. Foreign subscriptions (surface mail) $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail) please inquire. In South Africa contact *80 Micro*, P.O. Box 782815, Sandton South Africa 2146. All U.S. and Canadian subscription correspondence should be addressed to *80 Micro* Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form 3579 to *80 Micro* Subscription Services, P.O. Box 981, Farmingdale, NY 11737.




Cover by William Giese

***TRS-80, Scripsit and TRSDOS are trademarks of Tandy Corp.**

## Articles


**88. The Calculating Genius—Part II**
Charles Babbage's Analytical Engine was the modern computer's ancestor, but it was never built.
*Rick Cook*

**94. Flaky**
A graphics program to show off the power of your Color Computer.
*Sidney C. Garrison*

**101. Bit Smitten**
This month learn how to gain access to string data for use in programs requiring random selection of words.
*Jay Chidsey*

**104. Up Your Mod I**
Double-density disk storage—a close look at all you will ever need to upgrade your Model I.
*Bruce Powel Douglass*

**118. GRAPHTRAX Tricks**
Make your MX-80 look like a magician.
*Mark Schneider*

**125. Forms Cleanup**
Eliminate drawbacks in this printer-driver by dropping the O.
*Jim Barbarello*

**130. Inside Scripsit—Part III**
How to write a spooler for Radio Shack's word-processing program.
*Craig A. Lindley*

**146. A Matter of Compatibility**
Make your program submissions to *80 Micro* compatible with other systems with this short routine. Your readers will appreciate it.
*Art Huston*

**166. $30 System Desk**
Now that you've bought a computer, build a place to put it.
*Sanford J. Asman*

**173. Hardware Hacker—Part III**
Add a disk-drive controller to your system.
*Philip M. Van Praag*

**194. The Looking for Z$ Blues**
God bless the 80 that's got its own search function.
*J. E. Hieber*

**200. CC CQ**
A Ham puts his Color Computer to dit-dah use.
*Michael Chuck*

**228. Liberated CPU**
Free your micro with this spooler.
*Tom Shield*

**236. Diskmap**
Worry no more about disk file placement.
*John B. Harrell, III*

**252. Astro-Scrambler**
Transplant your rainy day high school gym class into space and play dodge ball with asteroids.
*Randy Long*

**256. Unrolling Craps**
If you're a gambling fool, be less foolish by calculating the odds before you roll those bones.
*Dan Robinson*

**262. Search Them Strings**
This utility helps you find strings in your Basic programs.
*John R. Gunter*

**266. Certify Your Cassettes**
Eliminate bad saves and wasted time.
*Kenneth J. Bigelow*

**276. Another Sort of Sort**
After the bubble bursts, learn about position and chain sorts.
*Karl L. Townsend*

**280. Autokey**
Modify Uni-Key for your Model III.
*Don Rigg*

**298. Memory Map—Level II**
A general guide to the regions of ROM.
*Mark D. Goodwin*

# 80 Contents

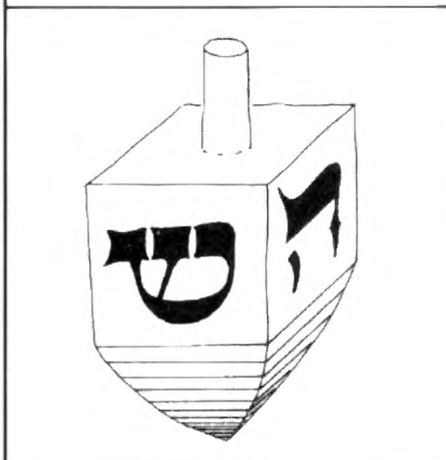


The left bracket, [, replaces the up arrow used by Radio Shack to indicate exponentiation on our printouts. When entering programs published in *80 Micro*, you should make this change.

*80* formats its program listings to run 64 characters wide, the way they look on your video screen. This accounts for the occasional wrap-around you will notice in our program listings. Don't let it throw you, particularly when entering assembly listings.

Article submissions from our readers are welcomed and encouraged. Inquiries should be addressed to: Submissions Editor, 80 Pine Street, Peterborough, NH 03458. Include an SASE for a copy of our writers' guidelines. Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased. Authors of reviews should contact the Review Editor, 80 Pine Street, Peterborough, NH 03458.

**419. Fun House**
The 12 Days of Christmas. Dreidel, a Hanukkah game. Peglegs.
*Richard Ramella*




**433. Feedback Loop**
Lazy Writer. Copying single density to double density. Printer III problems. Disk drive buyer information. Disk drive problems. EDTASM problems. Color Computer information. Buying a Model III keyboard. Model I monitor ripple. Playing Naval Wars on the PMC. Best chess game available for 80. Power surge protection.
*Terry Kepner*


**442. MONEY DOS**
How to play on the options market.
*J. M. Keynes*

**451. Index to Advertisers**

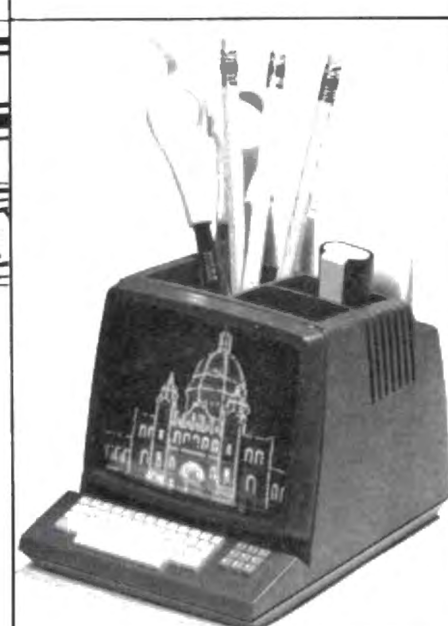
**452. Notes from Beneath the Keyboard**
A conversation with Vernon Hester, including a Hester machine-language routine to make programs compatible with all DOSes.
*Paul Wiener*

**458. Copernica Mathematica**
Analysis of variance.
*Bruce Powel Douglass*

**464. Reload 80**
How to use Color Load 80. A new DOS for the Models I and III. Anniversary Load 80.
*Art Huston*

**468. Calendar**

**470. New Products**
Vendor-One program authors' software. Mitsubishi 5.25-inch floppy drive. Color Computer macro assembler/editor. Farm Weather Center. VisiCalc consolidation system. Name That Song. Australian Winchester drives. Graphic character generator for Models I and III. Two-mode joystick. Camp management program. Reading Skills Lab. Personal Computer Owner's Directory. CompuGifts for Christmas.





Circle 31 on Reader Service card.

# 80 Contents


**312. Cybernetics—Part III**
How to interface an actuator to your computer.
*Stephen Davids*

**320. Model III Machine-Language Modifications**
When Model I software isn't Basic, you need to know some tricks before you convert the programs.
*Richard Koch*

**333. Queue Up!**
A smart businessman knows analyzing waiting lines is not as arcane as it sounds.
*David E. Clapp*

**341. Maze XIX**
Have you ever wondered how a mouse feels in a maze? Here's your chance to find out.
*Gary Teter*

**343. Open Cartridge Surgery**
Out with the bad ribbon, in with the cheap one.
*Bill Grout*

**344. NEWDOS80 Files**
An explanation of their functions.
*Randy M. Resnick*

**354. Color Diskdump**
Model I users had it, now Color Computer owners can have it, too. So examine those files.
*Ken Knecht*

**356. Forms Maker**
With USR routines and this Basic utility, you can create business forms on your Model II.
*Charles R. Perelman*

**360. Screen Strings**
Make your screen routines efficient.
*Don Davidson*

**366. Boolean Logic Operators**
How they work for Fortran and Basic.
*John Crew*

**370. Scroll Your Listings**
Use this program to list programs on the screen one line at a time.
*Mark C. Paxton*

**376. APL Primer—Part I**
What's the difference between monadic and dyadic functions? Stay tuned for a description of system commands, variable names, and more.
*Margaret M. Grothman*

**384. POKE Graphics**
Speed up your Basic displays with a little help from some dummies.
*James Schaefer*

**388. Easy Picture Editor**
Simple commands for art or game graphics on the Color Computer.
*Ron Ginger*


## Departments


**8. Remarks**
With its business health depending increasingly on its computer sales, some rough water could be awaiting the Tandy flagship.
*Wayne Green*

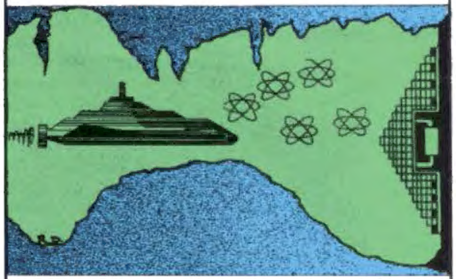
**16. Proof Notes**
A look at the year ahead.

**18. Input**
The amazing magazine mystery. Model I interface. Wayne's view of America rapped. Tips for Extended Basic. Rammer fix. More advantages of storing Assembly subroutines in Basic remark statements. PMC EPS-80 defended.

**34. Debug**
September MONEY DOS fix. Errors in Screenplay program. Model I mod for "Save Our Ship." A word on golf handicaps.

**38. Aid**
Goodwill Industries wants computers. Business students need programs. Green screen kluge. Quick Printer I for sale. More buffer for his music. Needs address of Indianapolis firm. VisiCalc templates. For Stringy Floppy owners only. Modem for 1200 downloads. TRS-232 newsletter. Color monitor question. Super Scripsit.

**40. Reviews**
Sea Dragon. TRS-80 Assembly Language. Wolfbug. Elementary Basic and Pascal with Sherlock Holmes. Lisp/UOLISP. Quest. LSI Soft-view CRT. FED File Editor. Bounceoids. The Disk Doctor. LCA-47. Lowercase Adapter for CoCo. Hypergate Centurian. Data-O-Base Calendar. Drive Control Unit.

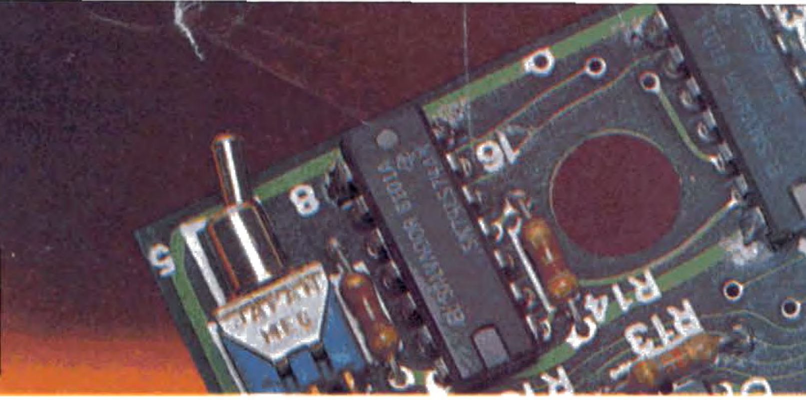

**63. Review Digest**
Understanding Computer Science by Roger Walker. Postman. Basics for Business by Douglas Hergert. Master Control. Big Things from Little Computers: A Layperson's Guide to Personal Computers by Dale Peterson. Pro-80 Extended Basic. Micropilot. The Tiny Compiler. Stellar Escort. RAM Slam.

**64. Soft Bits**
Examining the bits in binary numbers.
*Roger Fuller*

**68. Gamer's Cafe**
A call to gamers everywhere to beat the high scores of Winthrop and Mad Max.
*Rodney Gambicus*

**396. News**
Computers in the schools: boon or bane? Video games can weaken a heart. Landmark lawsuit settled. Poet Laureate of the Network Nation.


# THE SWITCH



## SWITCH TO 5"/8" DOUBLE DENSITY

# *LNDoubler 5/8*

**FEATURES**

- 5- and 8-inch* disk drives
- Single- & double-density
- Any size and density in any mix
- Read Model I, II* and III disks
- 5- or 8-inch* system disk
- Single & double sided disk drives
- DOS+ 3.3.9 included, with Disk BASIC.
- 6 month warranty
- Up to 3.75 megabytes online
- Easy installation – plug-in & run
- Analog phase lock loop data separation
- Precision write precompensation
- Regulated power supply
- Guaranteed operation at 4MHz
- All contacts gold plated
- Solder masked & silk screened
- Runs under DOS+ 3.3.9, TRSDOS 2.3, NEWDOS 2.1, NEWDOS/80 1.0, LDOS, NEWDOS/80 2.0, and ULTRADOS
- Reads 40- and 35-track disks on 80-track drives
- FD1791 controller + your FD1771
- Fits Model I expansion interfaces
- Fits LNW expansion interfaces
- Track configurations to 80-tracks
- 5 inch disk storage increased to:
  161,280 bytes – 35-track SS/DD
  322,560 bytes – 35-track DS/DD
  184,320 bytes – 40-track SS/DD
  368,640 bytes – 40-track DS/DD
  368,640 bytes – 80-track SS/DD
  737,280 bytes – 80-track DS/DD
- 8 inch disk storage increased to:
  591,360 bytes – 77-track SS/DD
  1,182,720 bytes – 77-track DS/DD

SS: single-sided DS: double-sided
SD: single-density DD: double-density

**COMPLETE** – The *LNDoubler 5/8*, switches your Model I or LNW-80 into the most versatile computer you can own. The LNDoubler's switch allows you to boot from 5- or 8-inch system disks, and it's accessible from outside the interface. The *LNDoubler 5/8* comes with a double-density disk operating system (DOS+ 3.3.9), complete with BASIC and utility programs . . . ready to run your software NOW!

**VERSATILE** – Whether you want single-sided, double-sided, single- or double-density, 5- or 8-inch operation, complete versatility is here today! Any combination of 5- and 8-inch disk storage is possible with the *LNDoubler 5/8*. Each of your present 40-track, single-sided 5-inch drives will store up to 184,320 bytes (formatted storage) – that's an 80% increase in storage capacity for only half the cost of just one disk drive. With three 8-inch double-density, double-sided drives your Model I will have 3.75 Megabytes of online storage – that's more storage than a Model II or Model III!

**ADVANCED** – The *LNDoubler 5/8* is the most technically advanced, tested and reliable double-density board you can buy. The *LNDoubler 5/8* has more features, more options and more software support than any other product of its kind.

**EASY TO INSTALL** – The *LNDoubler 5/8* is easy to install. There are no traces to cut, no wiring to do, just a screwdriver and a few minutes of your time is all that is required. The instructions are fully illustrated for all interfaces. In minutes you will be 'up-and-running', and enjoying your computer as never before.

**COMPARE** – Compare features, compare quality, compare value, and make the **SWITCH** today!

**Immediate** delivery from stock – at your dealer **NOW** for only

## $219.95

DEALERS – You too can make The Switch.

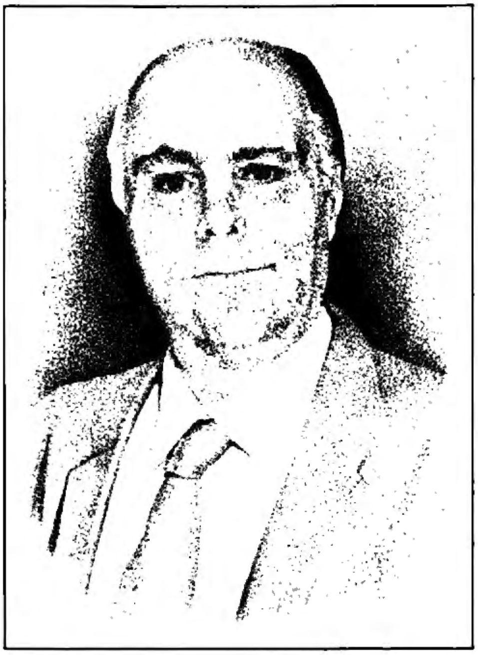
## LNW RESEARCH CORPORATION

**2620 WALNUT Tustin, CA. 92680 (714) 544-5744 (714) 641-8850**

*8" drive operation requires special cable, 8" double-density requires 3.55MHz CPU speed-up modification or LNW-80 4MHz computer. TRS-80 is a trademark of Tandy Corporation.

by Wayne Green

# Is the Shack in real trouble?



With the virtual demise of CB, a serious loss of consumer interest in hi-fi, and the overall slowdown in the buying of electronic gadgets, Radio Shack stores have found that they are more and more dependent on their computer products. It is not much of an exaggeration to put this bluntly: as go computer sales, so goes Radio Shack.

So how go the computer sales?

Starting with the Color Computer, Radio Shack's low-end system aimed at the home computer market, it's bad news. When the Shack was up against small firms selling high-priced products, their 6,400 stores gave them an unbeatable edge. Even today, with less than 5,000 bona fide computer stores, even if a rival firm managed to get 100 percent penetration of this distribution system, they would still not quite match the Shack.

But home computers are not dependent on computer stores. The low-end computer systems are being sold through toy stores, discount stores, department stores, electronic stores, and so on. The Shack has been overwhelmingly outnumbered in sales outlets—and in sales. Atari, Texas Instruments, and Commodore are all beating the hell out of the Shack in the home market.

This volume of sales has, in turn, allowed the firms to undercut the Shack in price. This further is hurting the Shack.

Shack is trying, belatedly, to develop some sales through RCA outlets for the CoCo, but the neglect of this system in many ways, as I've griped about before, may now have dealt them out of the fast-growing home-computer market as a major participant.

The Shack brought out CoCo in August 1980 and then virtually abandoned it. Either unable or reluctant to get enough of the right people on the Shack staff to handle the multiplying problems of changing from the Model I to the Model III, plus endless aggravation with the Model II, the CoCo and the pocket computers were announced, put into the catalogs, and left to fend for themselves.

So here we are two years later—a lifetime in the microcomputer industry—with a pitiful lack of serious support for the system. It seems that the Shack people are at war with their supporters and potential suppliers. On the one hand, perhaps from a desire to make every possible dollar of profit, they try to do as much as possible in-house. But even when they see that this is pushing them further and further behind, they don't change their policies. Having worked with the Shack for many years, I get the idea that once a policy is made it is put into concrete and no amount of disasters is going to change policy.

I'm sure there are at least fifty software firms out there that would love to work with the Shack to develop the needed software for CoCo. I know that our Instant Software people have at least 500 programs that could be converted to run on the system, and which they have tried to talk with the Shack about. I've talked with many hardware accessory people who have similarly been frustrated in trying to talk with the Shack.

It is getting clearer that home computers will eventually take over the video game market, allowing people to play all of the games. . . and also run home programs, educational programs, and so on. On that basis, is there anything that the Shack can do to cope with this growing market, and get back into competition?

Yes. . .and no. Yes, of course the Shack can do it—but no, not without breaking some serious concrete decisions. They've erected their own Great Wall between continued growth and gradual decline.

As the messenger I expect that I will be skinned alive, as I have in the past. They sure don't like messengers.

**The Software**

By making the CoCo incompatible with all their other Shack computers, they set themselves up with a major problem—no software. The Shack's normal approach is to hire some programmers and get the software written. But here there is a little problem. Even the massive Shack resources aren't enough to hire very many programmers. These days, programmers with any creativity and experience can sit at home anywhere they want and make all the money they need. Who needs to move to Fort Worth, where for about half the year you either have an air conditioner handy or you melt? I lived there one summer, so I know!

Most of the better programmers are writing programs and getting them distributed through software houses and are not even remotely interested in sweat-shop, regimented nine-to-five jobs. . . jobs where one can be fired as soon as a temporary problem has been surmounted.

The Shack has done the best it could to find programming talent to grind out software, and the results, as reviews of the Shack software make all too clear, are disappointing. It is getting time to face up to the failure of this system of trying to cut corners on costs, always trying to get a few pennies' more profits to drive up the quarterly stockholder reports.

If the Shack would start dealing honestly and forthrightly with some of the software suppliers to the industry, I think they could get plenty of software to support the CoCo—and that certain-

# META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

participating organization joint marketing services

## PARAGON Gold magnetics™ DISKETTES

MTC's

premium Single-Sided, Soft-Sectored DOUBLE-DENSITY, 5¼-inch diskettes with reinforcing HUB-RINGS. Individually 100% ERROR-FREE certified. Invest in GOLD!

PARAGON MAGNETICS GOLD ...... $24.95

## Scotch®

### Soft-Sectored Diskettes

| | |
|---|---|
| 5-1/4" 1S/SDen (744-0) | $28.95 |
| 5-1/4" 1S/DDen (744D-0) | $31.95 |
| 5-1/4" 2S/DDen (745-0) | $39.95 |
| 8" 1S/SDen (740-0) | $29.95 |
| 8" 1S/DDen (741-0) | $37.95 |
| 8" 2S/DDen (743-0) | $49.95 |

### Hard-Sectored Diskettes

| | |
|---|---|
| 5-1/4" 1S/DDen 10-sector (744-10) | $28.95 |
| 5-1/4" 1S/DDen 10-sector (744D-10) | $32.95 |
| 5-1/4" 2S/DDen 10-sector (745-10) | $39.95 |
| 5-1/4" 1S/SDen 16-sector (744-16) | $28.95 |
| 5-1/4" 1S/DDen 16-sector (744D-16) | $32.95 |
| 5-1/4" 2S/DDen 16-sector (745-16) | $39.95 |
| 8" 1S/SDen 32-sector (740-32) | $33.95 |
| 8" 1S/DDen 32-sector (741-32) | $39.95 |
| 8" 2S/DDen 32-sector (743-32) | $49.95 |

### Supplies

| | |
|---|---|
| 5-1/4" Head cleaning kit | $29.95 |
| 8" Head cleaning kit | $29.95 |

Authorized Distributor Information Processing Products 3M

## VERBATIM

### Soft-Sectored Diskettes

| | |
|---|---|
| 5-1/4" 1S/DDen (MD525-01) | $26.95 |
| 5-1/4" 2S/DDen (MD550-01) | $39.95 |
| 5-1/4" 2S/4Den (MD557-01) | $51.50 |
| 8" 1S/DDen (FD34-8000) | $43.95 |

### Hard-Sectored Diskettes

| | |
|---|---|
| 5-1/4" 1S/DDen 10-sector (MD525-10) | $26.95 |
| 5-1/4" 2S/DDen 10-sector (MD550-10) | $39.95 |
| 5-1/4" 2S/4Den 10-sector (MD557-10) | $51.50 |

## 'RINGS' & THINGS

| | |
|---|---|
| HUB RING KIT for 5¼" disks | $10.95 |
| HUB RING KIT for 8" disks | $12.95 |
| REFILLS (50 Hub Rings) | $ 5.95 |
| 5¼-inch diskette case | $3.50 |
| 8-inch diskette case | $3.95 |
| 5¼-inch File Box for 50 diskettes | $24.95 |
| 8-inch File Box for 50 diskettes | $29.95 |

## FILE BOX

### DISKETTE STORAGE SYSTEM

$17.95* with SCOREPAC™

Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

Reg. $24.95.....NOW $17.95

*** With the purchase of ONE**

**PARAGON GOLD**

**SCOREPAC™**

Isn't it time that you scored . . .

A Great Gift Giving Idea!!

SCOREPAC™
w/20 Paragon Gold Diskettes ...... $46.95

## MODEL I/III SOFTWARE ACCESSORIES

| | |
|---|---|
| APPARAT'S NEWDOS/80-V2 + MTC QUE CARD™ | $ 129.95 |
| PRACTICAL PERIPHERALS MICROBUFFER™ | $ 149.95 |

Michael Shrayer's

### ELECTRIC PENCIL VERSION II

SPECIFY MODEL I OR III

| | |
|---|---|
| Disk Version | $79.95 |
| Tape Version | $69.95 |

## APPLE® OWNERS

***Your first mystery is here!***

**THE CUSTOM APPLE® & OTHER MYSTERIES ....... $24.95**
by Hofacker & Floegel

Make your Apple® Computer perform tasks you never thought possible. Interface to stepping motors, fans, drive motors, etc. Remotely control and automatically monitor virtually any machine previously controlled by mechanical means. If you want to lower your heating costs or are just wondering how to hook that "new" printer to your Apple® you need this book! Volume 1 of IJG's Apple Information Series contains more than 180 pages of practical information and tested software.

## "TRS-80™ DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

TRS-80™ DISK ........... $19.95

## "OTHER MYSTERIES" VOLUME II

by James Farvour

MICROSOFT™ BASIC DECODED $26.95

## "OTHER MYSTERIES" VOLUME III

by Dennis Kitsz

THE CUSTOM TRS-80™ ........ $28.95

## "OTHER MYSTERIES" VOLUME IV

## BASIC FASTER AND BETTER

| | |
|---|---|
| BASIC FASTER & BETTER | $26.95 |
| BFBLIB subroutines diskette | $19.95 |
| BFBDEM demonstration diskette | $19.95 |



MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

24 Hr. Bulletin Board (216)289-8689

PRICES IN EFFECT THRU **December 31, 1982**
Prices, Specifications, and Offerings subject to change without notice
8212

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling.
- $2.00 EXTRA for U.S. Mail delivery.
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6.5% sales tax.

ly is an important first step. They should have done this two years ago.

Another cherished policy of the Shack that needs to be re-examined, as I mentioned a few months ago, is the one about their stores carrying only the fastest selling products. In the case for software, this may be a self-defeating policy.

In the consumer-products market, perception of the product is paramount. The computer that is backed up by a display of a barrage of software is going to be more impressive than the one with a dozen or two programs. Sure, over 90 percent of book sales are made by a couple of dozen titles, but you never see a serious book store that does not have thousands of books.

The consumer this Christmas is going to have to choose among the Atari-400, the TI 99/4A, the Commodore VIC-20, the Commodore Max, the Timex-1000, and the TRS-80 Color Computer. The Shack realizes that they are going to have to cut prices from the present $400 range down to at least $300. This still is going to be a difficult sale for them against the $189 VIC-20 or the Atari-400 at $299 plus $60 in software coupons. The TI, with their $100 rebate, is in the $200 range.

It's far too late to do anything for this Christmas season, but the Shack could get started on a plan that would, by Christmas 1983, put them into a leadership position. This would call for some serious changes of policy and for a good deal of long-range planning.

If you look through *80 Micro*, you see how much can be developed in the way of supporting industry for a system. Well, with cooperation (a concept I don't think is in the Shack dictionary), this industry would jump at the chance to support the CoCo. What would be the impact on consumer sales if the Shack had a home-computer system with a massive amount of software supporting it? What would be the result if it was also supported by the enormous accessory and peripherals industry that has sprung up around the TRS-80 system?

You can bet that Atari, TI, and Commodore would be scurrying fast to try to keep up. The Shack would have the edge in this case because a whole industry has already sprung up to support the TRS-80 by virtue of *80 Micro*, while there is relatively little support for the other home computer systems, since no media such as this magazine exist to bring on growth. The Shack would be starting with a good lead in this case.

The next step, if I were working on the project, would be to separate the toy department of the Radio Shack stores from the home computer department. I would set up a separate home computer boutique in every Tandy-owned store where I would feature the CoCo, its accessories and software, plus the Model III and pocket computers. I would relegate the Model II to the Tandy Computer Centers. Go with strength.

It is at least worth a try in a few stores to see what impact this sort of marketing might have. The Shack still has a long way to go to convince the middle-income (-class) customers that Radio Shack stores are not primarily dealers in schlock for the unwary lower-income people. By separating the home computers from their flashlights, ineffectual radar detectors, lo-fi audio systems, and so on, it might be possible to build some middle-income consumer confidence in the computer products.

I think it would work.

Once they've worked out a successful home-computer boutique formula for their own stores, they can start putting these into independent Radio Shack stores, thus adding another 2,000 or so outlets for the product.

That still isn't going to match the marketing might of Atari and the others, but with a successful home-computer boutique formula, they would be ready to set these up in *any* discount store or department store interested in a profit center. They would have the whole miniature store instead of some shelves of products, as a unit ready to franchise.

The computer and its associated products would all be packaged to simplify selling by untrained sales personnel. You want a music keyboard? Here it is. You want the unit to talk? No problem, here's the box for that. You want it to access data over the telephone? Here's a box with a modem. You want it to plot in four colors? Here's your plotter. You want it to help with surveying? Here's a software package for that. You want to

# META TECHNOLOGIES

~13

26111 Brush Avenue, Euclid Ohio 44132

**CALL TOLL FREE 1-800-321-3552 TO ORDER**

IN OHIO, call (216) 289-7500 (COLLECT)

participating organization
joint marketing services

## A Few Words About
## The NEW AIDS-III™
by SofTrends, Inc.

**1. TRUSTWORTHY.** A bad diskette or an undependable machine can ruin your whole day. The NEW AIDS-III checks itself. If something's wrong, it tells you, instead of turning on you like a mad dog.

**2. GENEROUS.** The NEW AIDS-III doesn't use BASIC. But it does use the memory BASIC uses. And to make the most out of that extra memory, it selectively compresses bytes into tiny bits. It all adds up to more usable data records. Up to 3 times as many.*

**3. THOUGHTFUL.** The NEW AIDS-III never treats you harshly. If one keystroke will do, it won't ask for two. The NEW AIDS-III remembers things like report formats, search strategies and file names. So you don't have to. It even reminds you, gently, to save your important data.

**4. INTELLIGENT.** The NEW AIDS-III is smart. It doesn't waste your time with questions about record sizes, field counts and other technical mumble-jumble. A new system can be created, or an old one modified, in a couple of minutes. Even if your name isn't Albert Einstein.

**5. KIND.** The NEW AIDS-III is always ready to help. It says so, on every screen display. HINTS™ (Help INdexed To Screen) tells you on which page in the NEW AIDS-III manual to look for more information. The manual is easy to understand and easy on the eyes.

**6. QUIET.** If you hit the wrong key, you won't hear any annoying buzzes, clicks or chirps. Instead, FLAWS™ (FLash-Annunciated Warning System) will create a striking visual effect. But only for an instant. And without affecting any of the text on the screen. Guaranteed to catch the eye of the fastest touch-typist.

**7. ALERT.** If the NEW AIDS-III is left alone, it lets you know it missed you. It worries about your important data. After several minutes of no activity, the NEW AIDS-III creates a striking visual display to get your attention. Touch any key to let it know you're still there, and it stops. For a little while, anyway.

**8. LEAN.** There's no fat in the NEW AIDS-III. That's because it uses SofTrends' proprietary PMX™ system architecture. Small, lightening-fast, reliable. Lean? Yes. Mean? Definitely not.

**9. FAST.** Searches and sorts hundreds of records in seconds. Screens are displayed in the blink of an eye. Disk access rates approach one-thousand characters per second. No waiting for "garbage collection". The NEW AIDS-III lives fast. Up to 10 times as fast.*

**10. REASONABLE.** At only $79.95, the NEW AIDS-III is very reasonable. Downright inexpensive, if you value your time. Join the thousands of AIDS owners around the world. Order yours today and put the NEW AIDS-III data management system to work for you.

* As compared to MTC AIDS-III, Version 1.0

**Specify Model I or Model III ............................ $79.95**

**** CALL REGARDING UPGRADE POLICY ****

**CALCS-IV (ADVANCED CALCULATION SUBSYSTEM) has the following features:**

- All the capabilities of CALCS-III.
- SAVE REPORT FORMATS on disk for EDITING, as required.
- More than TWICE the COMPUTATIONAL POWER of CALCS-III.
- Non-interpretive report execution for FASTER EXECUTION.
- Reports can be run by a NOVICE.

**ONLY $20 when purchased with AIDS-III/Version 2.0!**

**Specify Model I or Model III ............................ $39.95**

## NEW AIDS SUBSYSTEMS
by SofTrends, Inc.
## VISAPLEX™

Interfaces AIDS-III and VisiCalc*. Use AIDS-III for data entry, sorting and selection. Then load the data into VisiCalc*, perform computations, summations, etc. Like what you see? Change the data back into AIDS-III format for future processing. Remarkably easy to use. Comprehensive documentation complete with examples.

**Specify Model I or Model III ........... $39.95**

## ADEPT™

This module is ideal for entering large batches of data into AIDS-III. Features include pre-defined field values, definition and expansion of abbreviations, transposition of entries, range checking, entry of data from previous record, expanded validation and more! Type as fast as you can . . . no problem! Use with VISAPLEX™(above) to provide a comprehensive data entry facility for VisiCalc*. Complete documentation with examples.

**Specify Model I or Model III ........... $29.95**

If you own Apparat's NEWDOS/80-Version 2.0 you need

## BREVI-T™
by SofTrends, Inc.

If you have trouble remembering command formats, want to simplify use of DOS, do a lot of program development or just want to be more effective with your TRS-80™, then BREVI-T is for you. Abbreviations can be defined for both DOS and BASIC. These are automatically expanded as part of the command line processor. Optionally, parameters may be defined as part of an abbreviation. For example, "F 1" might be used to FORMAT drive 1. Change the 1 to a 2 and FORMAT drive 2. It's that simple. Complete with easy-to-follow instructions, examples and a sample abbreviation file.

**Specify Model I or Model III ........... $19.95**

Let your TRS-80™ Test Itself With

## THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC
by David Stambaugh

A complete checkup for your MODEL I or MODEL III. THE FLOPPY DOCTOR-Version 3 completely checks every sector of single or double density 35-, 40-, 77-, or 80-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

SYSTEM DIAGNOSTICS ..... $24.95
For MODEL III ..... $29.95

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. VisiCalc is a trademark of Visi Corp. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS IV, MERGE III are trademarks of MTC.
1982 by Meta Technologies Corporation

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

24 Hr. Bulletin Board
(216) 289-8689

PRICES IN EFFECT THRU
**December 31, 1982**
Prices, Specifications, and Offerings subject to change without notice
8212

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling.
- $2.00 EXTRA for U.S. Mail delivery.
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6.5% sales tax.

do boat navigation? No problem. All of the accessories and software would be integrated and packaged.

But what about training salesmen? That's one of the weakest aspects of the Shack stores right now. In fact, I suspect that this problem alone has kept most of the independent Shack stores from really participating in the bonanza so far. You didn't think I was going to ignore such a key problem, did you? The answer is simple, once you think of it.

Video disks. Yep! With disk players dropping in price, I would run, not walk, to my nearest video production organization and put together powerful video programs that show what the CoCo, Model III, and pocket computers can do. They would be indexed to show off all the accessories and software, and show the use of the systems for playing games, for business applications, education, scientific, control, and so on. With the video disk and a control CoCo in the salesroom, the customer would be able to delve into the possibilities of the systems with no trained sales help needed.

In this way the best demonstrator of the Shack system in the world could do the selling to every prospect. Further, the buying process in itself would be fun. Further, there would be no end to the sales to every customer. He might buy the basic computer and a couple of programs the first time around, but he'd be back regularly to check out more and more applications...and fun...for his system. It is *fun* to buy things...and fun to use them and show them off to friends. Who has ever bought a car and not started looking for gadgets to buy for it? Who has bought a camera and not looked for more lenses, special films, and so on?

If the Shack can show that a 200-square-foot home-computer boutique can make more profit than other uses of that space, you can bet that Sears, Lechmere, K-Mart, and other stores will be putting 'em in as fast as they can. And if they can't make big profits in a fast growing field—a go-go field—on home computers, then the Shack needs better management.

I'm reminded of the basic problem that Heath faced when they brought out their computer in 1977. They had about 50 stores around the country and they had decided to make do with this distribution. I tried hard to convince them that in the microcomputer field it was marketing and distribution, not the magnificence of the computer, that would determine the winners and the losers. I wanted them to open up their sales to computer stores, electronics stores, and so on, so they would have a better chance at making their system available in all of the marketing areas of the country.

The Shack had a considerable advantage over everyone else in the micro business by virtue of their thousands of stores. This, plus their advertising budget, helped them get into first place in microcomputer sales...shooting them ahead of Heath. But today, particularly on the low end, the Shack is falling way behind in sales outlets and thus in sales. There is a good chance that if the Shack does not respond fast to this deteriorating situation, they will just have to abandon the low end and the CoCo. This could play hell with the Tandy stock prices...and shake up management.

Well, at least the Shack is going strong with their main entry, the Model III, right? Yeah? If that is so strong, why do the *80 Micro* reader surveys show that 50 percent of the readers are thinking of buying something other than a TRS-80 for their next computer? This indicates some very deep-rooted and widespread disenchantment with the Shack, something that the Shack management seems to be trying its best to ignore.

Indeed, we've seen the Apple come along and, with fewer outlets, pass the TRS-80 by in sales. And while TRS-80 owners are generally happy with their systems, the Apple owners are almost evangelical about theirs. In this I suspect that the core, if you'll pardon the expression, lies in the way the firms support their systems. Where Apple has been very supportive of the industry that has sprung up around their computer, we find the Shack seeming to begrudge any sale not made by them and them alone.

When the Shack puts on a computer show, there are no supporting hardware firms invited...no software firms exhibiting...no magazines and books. When Apple organizes an Applefest, supporting firms are not only invited to display, they are encouraged in many ways.

There are many good books on the TRS-80 that are not permitted in the Shack stores. There are thousands of good software programs...a wealth of support hardware items. In each case, the Shack seems to be doing its best to keep news of these products from its customers, apparently wanting to do everything it can to be sure to get every nickel spent for anything related to the TRS-80 for themselves.

No matter the intent of management, this is the view of many users (and Radio Shack managers), so if there is a misinterpretation of intent, there is a serious lack of communication. Indeed, many TRS-80 owners have a sour view of Radio Shack, seeing the firm as the epitome of the big company that is contemptuous of its customers and the general public. It is difficult to interpret many of their actions in any other way.

Whether Apple pulls further ahead and IBM passes the Shack by depends on decisions made in the Tandy Towers this year. It is already late in the game, but not too late. The Shack could pull it together and get powered up to regain a strong market share of the home-computer market by the end of 1983. They could stop the attrition of their sales in the small business computer market. But these things are going to take some soul-searching, perhaps painful, decisions. They are going to call for some major changes of approach and the breaking of hardened Shack policies.

With such a high percentage of the Tandy sales being involved with the TRS-80, and with many of the other products being so weak in sales, there is a serious question about the survival of Tandy should the TRS-80 sales start to weaken.

In early 1977 Mits was thoroughly in command of the microcomputer industry. They were as far or further ahead of the rest of the industry as Radio Shack was in 1980. Then, but two years later, Mits had been swallowed up by Pertec and virtually disappeared from the scene. Pertec could easily have remained the top firm in the industry if management had not made one disastrous decision after another.

Next, it was Imsai that was on top of the heap...then Processor Tech. Each pulled some market miscalculations and major blunders that sank them. The big question, then, is whether the Shack can cure its problems and get back to its number one position again, or whether it is going to continue to lose sales and market share on all fronts, a process that can only lead to disaster. ■

# Our "almost wholesale" prices just got 2% lower.

**Take an additional 2% off our listed prices, until December 24.**

## Have a merry computer and a happy software.

### PRINTERS

| | |
|---|---|
| COMREX CR-1 PARALLEL | 839 00 |
| COMREX CR-1 SERIAL | 859 00 |
| C-ITOH F-10 40 CPS PARALLEL | 1390 00 |
| C-ITOH F-10 40 CPS SERIAL | 1390 00 |
| C-ITOH 8510 PROWRITER PARALLEL | 480 00 |
| C-ITOH 8510 PROWRITER SERIAL | 590 00 |
| EPSON MX-80 III W/GRAFTRAX + | $CALL |
| EPSON MX-80 III F/T W/GRAFTRAX + | $CALL |
| EPSON MX-100 III W/GRAFTRAX + | $CALL |
| IDS PRISM 80 W/COLOR | 1599 00 |
| NEC 8023A | 485 00 |
| NEC SPINWRITER 3530 P RO | 1995 00 |
| NEC SPINWRITER 7710 S RO | 2595 00 |
| NEC SPINWRITER 7730 P RO | 2595 00 |

### VERBATIM

| | |
|---|---|
| MD 525-01, 10, 16 | 26 50 |

### DISKETTES STORAGE

| | |
|---|---|
| 5¼" PLASTIC LIBRARY CASE | 2 50 |
| 8" PLASTIC LIBRARY CASE | 3 50 |
| PLASTIC STORAGE BINDER W/ INSERTS | 9 95 |
| PROTECTOR 5¼" | 21 95 |
| PROTECTOR 8" | 24 95 |

### TRS-80 HARDWARE

| | |
|---|---|
| PERCOM DATA SEPARATOR | 27 00 |
| PERCOM DOUBLER II W/ DOS 3.4 | 159 00 |

### MODEL III DISK DRIVE KITS

**CONTROLLER KIT** INCLUDES ALL BOARDS ASSEMBLED AND TESTED, INTERNAL CONTROLLER, MOUNTING BRACKETS, SWITCHING POWER SUPPLY, AND INSTALLATION INSTRUCTIONS. HANDLES 4 DRIVES 360 00

**KIT #1:** CONTROLLER KIT AND 2 TM100-3 SINGLED-SIDED, 80 TRACK DRIVES 875 00

**KIT #2:** CONTROLLER KIT AND 2 TM100-4 DOUBLE-SIDED, 80 TRACK DRIVES 1060 00

**KIT #3:** CONTROLLER KIT AND 2 TM100-1 SINGLED-SIDED, 40 TRACK DRIVES 720 00

**KIT #4:** CONTROLLER KIT AND 2 TM100-2 DOUBLE-SIDED, 40 TRACK DRIVES 850 00

| | |
|---|---|
| OKIDATA MICROLINE 80 | 389 00 |
| OKIDATA MICROLINE 82A | 460 00 |
| OKIDATA MICROLINE 83A | 700 00 |
| OKIDATA MICROLINE 84 | 1170 00 |
| OKIGRAPH 82 | 49 95 |
| OKIGRAPH 83 | 49 95 |

### EPSON ACCESSORIES

| | |
|---|---|
| MICROBUFFER‡ MBS-8K | 159 00 |
| MICROBUFFER‡ MBP-16K | 159 00 |
| MICROBUFFER IN-LINE 32K | 299 00 |
| MICROBUFFER IN-LINE 64K | 349 00 |
| GRAFTRAX PLUS | 60 00 |
| MX-80 RIBBONS | 8 95 |
| MX-80 PRINTHEAD | 40 00 |
| MX-100 RIBBONS | 24 00 |
| MX-100 PRINTHEAD | 49 00 |
| EPSON SERVICE MANUAL | 20 00 |

### 16K RAM SPECIAL 13.95

Set of 8 NEC 4116 200ns Guaranteed one year

### MODEL I, III SOFTWARE

| | |
|---|---|
| PROSOFT NEWSCRIPT/LABELS MOD I, III | 119 00 |
| OMNITERM SMART TERMINAL MOD I, III | 89 95 |
| MAXI-MANAGER MOD I, III | 89 95 |
| DOS PLUS 3.4 MOD I, III | 89 00 |
| LDOS 5.1 MOD I, III | 119 00 |
| MICROSOFT EDITOR ASSEMBLER + MOD I (DISK) | 44 95 |
| MICROSOFT FORTRAN MOD I | 80 00 |

### DISKETTES

ALPHA DISKS ............ 21 95

Single sided, certified Double Density 40 Tracks, with Hub-Ring Box of 10. Guaranteed one year

| | |
|---|---|
| TANDON 80 TRK DISK DRIVE W/ P.S. | 345 00 |
| TANDON 40 TRK DISK DRIVE W/ P.S. | 289 00 |
| LNW DOUBLER W/ DOSPLUS 3.3 | 138 00 |
| LNW DOUBLER 5/8 W/ DOSPLUS 3.4 | 171 00 |

### ISOLATORS

| | |
|---|---|
| ISO-1 3-SOCKET | 53 95 |
| ISO-2 6-SOCKET | 53 95 |

### MODEMS

| | |
|---|---|
| NOVATION CAT ACOUSTIC | 135 00 |
| NOVATION D-CAT DIRECT CONNECT | 156 00 |
| NOVATION AUTO-CAT AUTO ANSWER | 219 00 |
| HAYES SMART MODEM (300 BAUD) | 227 00 |
| HAYES SMART MODEM (1200 BAUD) | 540 00 |
| LEXICON LEX-11 | 109 00 |
| SIGNALMAN MODEM W/ RS-232C | 85 00 |

### SUPPLIES

AVERY TABULABLES

| | |
|---|---|
| 1,000 3½ x 15/16 | 8 49 |
| 3,000 3½ x 15/16 | 14 95 |
| 5,000 3½ x 15/16 | 19 95 |

### FAN FOLD PAPER

**(Prices F.O.B. S.P.)**

| | |
|---|---|
| 9½ x 11 18lb WHITE 3,000 ct | 29 00 |
| 14 7/8 x 11 18lb WHITE 3,000 ct | 39 00 |

### SPECIALS

SPECIAL NO.1

TRS-80 DISK AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 44 50

SPECIAL NO.3

NEWDOS/80 2.0, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 149 00

SPECIAL NO.4

MICROSOFT BASIC COMPILER, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 179 00

SPECIAL NO.5

MICROSOFT BASIC DECODED AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 49 95

DOUBLE DENSITY SPECIAL

PERCOM DOUBLER II, NEWDOS/80, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 279 00

COMMUNICATION SPECIAL MOD I & II

NOVATION MODEM, OMNITERM TERMINAL SOFTWARE

| | |
|---|---|
| ACOUSTIC SPECIAL (CAT) | 219 00 |
| DIRECT CONNECT SPECIAL (D-CAT) | 239 00 |
| SERIAL CABLE | 25 00 |

DISK LIBRARY SPECIAL

RACET DISKCAT CATALOGING PROGRAM, FLIP SORT (50 DISK CAPACITY), 50 STICK-ON DISK LABELS 59 95

### BOOKS

| | |
|---|---|
| THE CUSTOM TRS-80 | 24 95 |
| MICROSOFT BASIC FASTER & BETTER | 24 95 |
| CUSTOM I/O MACHINE LANGUAGE | 24 95 |

### TRS-80 GAMES

All games are disk versions. Cassette versions may not be available.

| | |
|---|---|
| TEMPLE OF APSHAI | 31 35 |
| HELLFIRE WARRIOR | 31 35 |
| STAR WARRIOR | 31 35 |
| RESCUE AT RIGEL | 23 36 |
| CRUSH, CRUMBLE AND CHOMP | 23 36 |
| INVADERS FROM SPACE | 17 95 |
| PINBALL | 17 95 |
| MISSILE ATTACK | 18 95 |
| STAR FIGHTER | 24 95 |
| Z-CHESS III | 24 95 |
| ADVENTURE NO. 1, 2, & 3 | 34 95 |
| ADVENTURE NO. 4, 5, & 6 | 34 95 |
| ADVENTURE NO. 7, 8, & 9 | 34 95 |
| DUEL-N-DROIDS | 17 95 |
| STARFLEET ORION | 21 95 |
| INVASION ORION | 21 95 |
| OLYMPIC DECATHLON | 24 95 |
| MONTY PLAYS MONOPOLY | 31 95 |
| SARGON II | 31 95 |
| BLACKJACK MASTER | 27 95 |
| ROBOT ATTACK | 17 95 |
| GALAXY INVASION | 17 95 |
| SUPER NOVA | 17 95 |
| TUESDAY MORNING QUARTERBACK | 26 95 |
| LUNER LANDER | 18 95 |
| THE MEAN CHECKER MACHINE | 21 95 |
| SPACE ROCKS | 18 95 |
| PIGSKIN | 17 95 |
| ZOSSED IN SPACE | 18 95 |
| ARCADE-80 | 21 95 |
| COMBAT | 18 95 |
| SPACE INTRUDERS | 17 95 |
| SILVER FLASH | 17 95 |
| MORTON'S FORK | 26 95 |
| PROJECT OMEGA | 21 95 |
| SCARFMAN | 17 95 |

**Call our Modem line for weekly specials.**

**Alpha Byte COMPUTER PRODUCTS**

## To order or for information call (213) 706-0333

## Modem order line: (213) 883-8976

We guarantee everything we sell for 30 days — no returns after 30 days. Defective software will be replaced free, but all other software returns are subject to 15% restocking fee and must be accompanied by RMA slip. No returns on game software, unless defective.

We accept VISA and MasterCard on all orders; COD orders, up to $300.

Shipping charges: $3 for all prepaid orders, actual shipping charges for non-prepaids. $3 for COD orders under 25lbs. ($6 for over) plus a $4 surcharge; add 15% for foreign, FPO and APO orders. Calif. add 6% sales tax, L.A. County add 6½%.

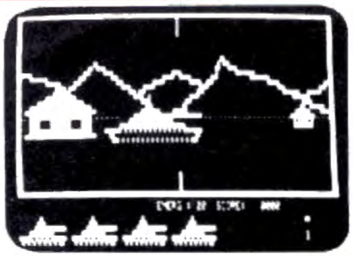
Prices quoted are for stock on hand and are subject to change without notice.

**31245 LA BAYA DRIVE, WESTLAKE VILLAGE, CALIFORNIA 91362**

✓473

## SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run Scarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound. Price A

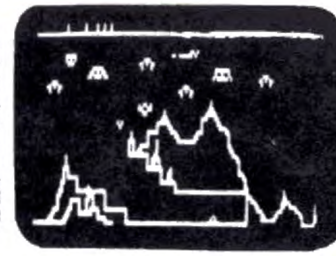


## ARMORED PATROL

A realistic tank battle simulation. Your view is a 3-D perspective of an alien landscape. Maneuver your T-36 tank to locate and destroy enemy tanks and robots that lay hidden, ready to assault you. Clever graphics create the illusion of movement and dimension. From Adventure International. With sound. Price B

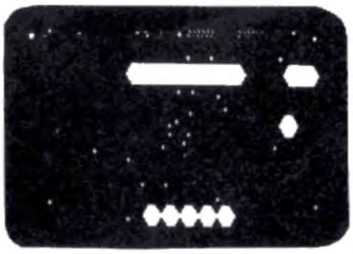
## REAR GUARD

Deadly waves of enemy Cyborg craft attack your fleet from the rear. You are the Mothership's sole defender. You have unlimited firepower but the Cyborgs are swift, nimble attackers. Your abilities are tested hard in this game of lightening fast action and lively sound from Adventure International. Price: B



## STRIKE FORCE

As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest, rapid fire missiles, long range radar, and incendiary "star shells." Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House. Price: A

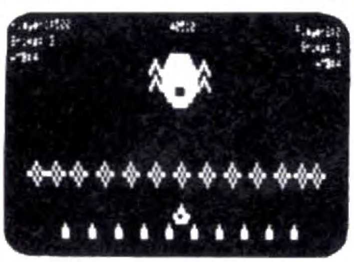


## BOUNCEOIDS

Huge boulders careen off the walls. You're in the middle, in danger of being flattened. Keep your wits about you as you blast these "bounceoids" from the screen. Large ones break into many small ones. Clear a screen, and enter a fast-paced challenge stage with a chance for big bonus points. From the Cornsoft Group. Price: A

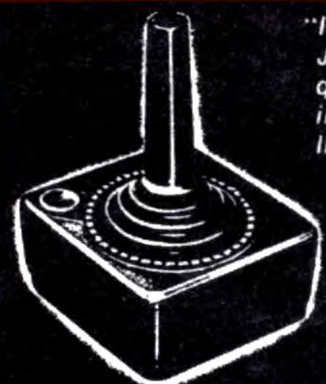
## CATERPILLAR

An arcade favorite! Stop these multi-sectioned crawlers before they creep down through the mushrooms. Zap one and it splits into two smaller bugs, each with its own sense of direction. There are moths and tumble bugs too. It all adds up to lots of fun for kids and adults alike. From Soft Sector Marketing. With sound. Price code A



## DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice. Price: A



"If you purchase Alpha's Joystick you get the exquisite pleasure of enjoying (action games) to the limit of arcade-style realism."

—80 Microcomputing 80 Reviews, Jan '82

## FEEL THE POWER...

## THE ALPHA JOYSTICK

ONLY $39.95

- + Features the famous Atari Joystick
- + Works with all Model I or III systems
- + Compatible with any other accessories
- + Saves your keyboard from abuse
- + Experiment in BASIC. Use A = INP(0)
- + Complete, ready to plug in and use
- + Model I, plugs into KB or E/I
- + Model III: plugs into 50 pin I/O bus

Price includes Joystick + Alpha Interface + Instructions + Demo Program listing. Please specify Model I or III.

© 1982 ALPHA Products

14 DAY MONEY BACK GUARANTEE

## CRAZY PAINTER

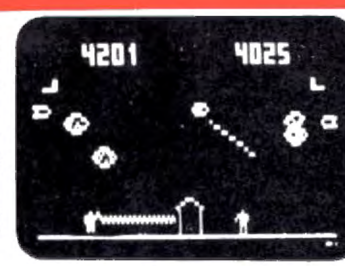
You have to paint the floor white. We give you the paint and brush. Sounds easy? Hah! You'll be confounded by stray dogs, snakes, sloshing buckets of turpentine, even a ravenous "paint eater." A crazy, imaginative new game with ten selectable levels of skill for new or seasoned game players. Lot's of laughs. Price: A

## SUPER NOVA

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine. Price: A

## METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking. Includes sound effects! Price: A




## OUTHOUSE

You are the mighty protector of this small (but important) wooden structure. For reasons unknown, a bizarre gang of miscreants wish to vandalize, loot and otherwise destroy the little "half moon house." Your patrol craft has lasers and smart bombs to deal with this terror. From SSM with sound. Price A

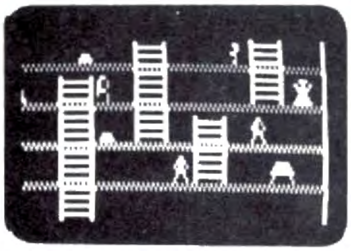
**Toll Free Order Line**
**800-221-0916**

Orders Only, NY & Info call (212) 296-5916. Hours: 9-5 E.S.T.

*TAPE: For Model I + III, 16K Level II*
*DISK: For Model I + III, 32K, 1 Disk*
*All games are joystick compatable or may be played using the arrow keys.*

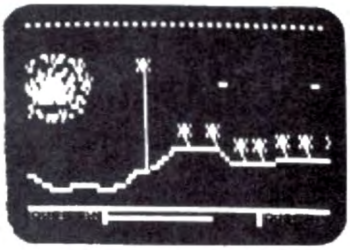
### GAME PRICES

A: TAPE: $15.95 • DISK: $19.95
B: TAPE: $19.95 • DISK: $24.95
C: TAPE: $24.95 • DISK: $24.95



## PANIK

Trapped at an enemy building site, your fate seems certain. Your laser is empty and evil Mzors are closing in. You'll have to climb ladders and think one step ahead of the various monsters. A challenging game for agile minds. From Fantastic Software with voice (Disk has larger vocabulary). Price: B



## SEA DRAGON

Your submarine, the U.S.S. Sea Dragon, penetrates a mined enemy channel. Armed with missiles and torpedos, you engage the enemy while navigating unknown waters. Succeed or come to a salty end in this game. 29 screens of horizontally scrolling seascrape and sound from Adventure International. Price: B

# SAVE

# 10, 15, 20%

## THE BEST FOR LESS

**As you can see, all the best games from the top producers are joystick compatible. These games are fun without the joystick but we hope that you are one of the many thousands who enjoy the advantage of real joystick action.**

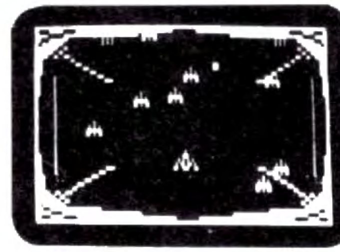
**Now you can deduct up to 20% on the price of games: buy any 2 games deduct 10%, buy any 3 games deduct 15%, buy any 4 games deduct 20% from game prices.**

## TOP TEN

1. **SCARFMAN** - *All time favorite*
2. **ARMORED PATROL** - *Super 3D graphics*
3. **PENETRATOR** - *Rave reviews*
4. **STELLAR ESCORT** - *Fast and Challenging*
5. **CRAZY PAINTER** - *Unique game concept*
6. **PANIK** - *Remarkable Voices*
7. **DEFENSE COMMAND** - *Tough struggle*
8. **CATERPILLAR** - *Good rendition*
9. **ROBOT ATTACK** - *With voice*
10. **SEA DRAGON** - *Amazing "Seascape"*

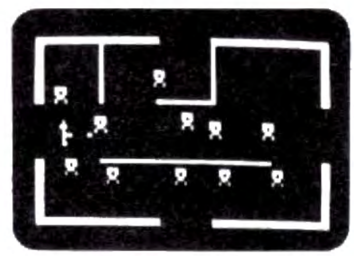
## STELLAR ESCORT

The latest super action game from Big Five. As the Federation's top space fighter you've been chosen to escort what is possibly the most important shipment in Federation history. The enemy will send many squadrons of their best fighters to intercept. With sound. Disk version has voices. Price: A



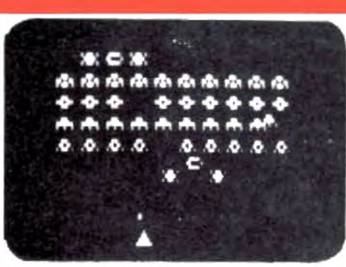
## ROBOT ATTACK

Talks without a voice synthesizer, through the cassette port. With just a hand laser in a remote space station, you encounter armed robots. Some march towards you, more wait around corners. Careful, the walls are electrified. Zap as many robots as you dare before escaping to a new section. More robots await you. Price: A



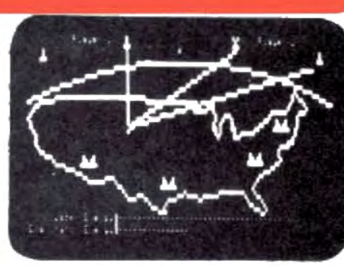
## LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound. Price: A



## GALAXY INVASION

The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects! Price: A



## LASER DEFENSE

In this game of ICBM's, high-energy lasers and particle beams, you control the U.S. strategic defense satellite system. From your viewpoint high above the globe, you intercept Soviet nuclear missiles in flight and attempt to destroy their scattered missile silos. With sound from MED Systems. Price: B

## CHICKEN

Will the chicken cross the road? That's up to you. Can you guide these helpless little chicks across the perilous 10 lane super highway to safety? Or will you bumble, littering the blacktop with a storm of chicken feathers? A humourous yet challenging game of nerves from SSM with sound. Price: A

## PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program. Price: C

**ALPHA Products**

*79-04 Jamaica Ave., Woodhaven, NY 11421*

ADD $2.00 PER ORDER FOR SHIPPING AND HANDLING.
WE ACCEPT VISA, MASTERCARD, CHECKS, M.O.
C.O.D. ADD $3.00 EXTRA
NY RESIDENTS ADD SALES TAX.
OVERSEAS, FPO, APO: ADD 10%
DEALER DISCOUNTS AVAILABLE.

✓17

# PROOF NOTES
## The editors look at the issues



Subscriptions:
*Problems with Subscriptions:* Send a description of the problem and your current and/or most recent address to: *80 Micro*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737.
*Change of Address:* Send old label or copy of old address and new address to: *80 Micro*, P.O. Box 981, Farmingdale, NY 11737. Please give eight weeks advance notice.
*Microfilm:* This publication is available in microform from University Microfilms International. United States address: 300 North Zeeb Road, Dept. P.R., Ann Arbor, MI 48106. Foreign address: 18 Bedford Row, Dept. P.R., London, WC1R4EJ, England.
*Dealers:* Contact Ginnie Boudrieau, Bulk Sales Manager, *80 Micro*, Pine St., Peterborough, NH 03458. (800) 343-0728.

### Ready to go for another big year

1982 hasn't even run its course, and we're already well into production for the first half of 1983. So far, it looks like we're going to have another interesting year.

January will feature one of the most exciting programs we've ever run—Bruce Tonkin's The Creator. This program generator originally sold for $295, but mayhem and misfortune led Tonkin to decide to virtually give his brain child away. This could very well be the best software bargain you'll ever get, commercial or otherwise.

February's highlight will be the results of our Young Programmer's Contest. We're getting buried under entries, and we'll be publishing the best ones. We've been agog at what kids are doing with computers, and are looking forward to displaying the fruits of their labors. February will also concentrate on home education. We've lined up a Color Computer College Bowl, a preschool math program, a spelling challenger, and a states-and-capitols drill.

March is our hardware construction issue. We've had lots of requests for some meaty hardware projects, so we went out and found some gems. For starters, we've got a $5 CP/M kit for the Models I and III. We'll also show you how to build a credit-card reader that will act as a security lock for your micro. We're also working on a primer that will tell you how to build hardware projects even if you don't know which end of a soldering iron to hold.

April will focus on micros in sports, and will also feature articles on computer languages other than Basic. May's cover topic will be graphics—simple techniques for fancy graphics, and incorporating graphics into applications programs. June will provide programs and articles aimed at the professional. July's theme will be the Color Computer, and August will, of course, be our annual games issue.

Also, you Model II owners will be happy to know that we're working on a scheme to convert some of our Model I/III programs to your machine. And, of course, we'll continue our increased coverage of the Color Computer.

We've got several new series and columns on tap. One will be an on-going tutorial on cryptography, while another will discuss integrating machine-language routines into your Basic programs. In addition, many of our regular columns will be back for another year, including J. M. Keynes' "MONEY DOS," Richard Ramella's "Fun House," Philip Mills' "Medical Opinion," and Dennis Kitsz's "80 Applications."

The magazine isn't all we're working on, either. We're putting together a new series of low-budget handbooks on a variety of themes. The first will be an anthology of Scripsit mods. Other subjects include home management, astronomy, and education. We haven't set a publication date yet, but look for these handbooks soon.

All in all, we think it's going to be an exciting year. Stick around and join the fun.

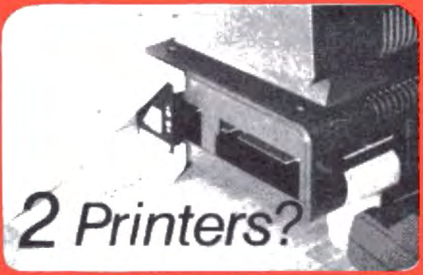
* * * * *

You might have noticed some recent format changes in the magazine. We're going to an English Times typeface, which we think is more attractive and easier to read than Helios. For a while, though, you're going to see the two typefaces mixed, as we use up articles that were finished before the change. Bear with us if we look a bit motley. ■

*—E.M.*

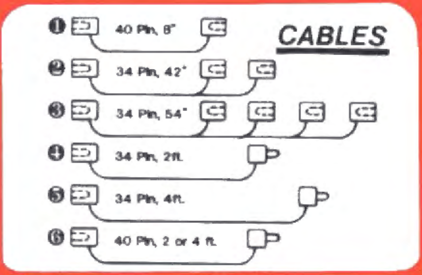
# NOW MODEL I AND MODEL III!

Now Model III users can take advantage of the ALPHA I/O system too. Our new MOD III/I BUS CONVERTER allows most port based Model I accessories (such as our ANALOG-80, INTERFACER 2 and INTERFACER-80) to connect to the Model III bus. MOD III/I BUS CONVERTER, complete with all connectors, only $39.95.




**PRINTSWITCH**

Have 2 printers on line at all times and select printer 1 or 2 by means of a conveniently located switch. End the problem of constantly plugging and unplugging printer cables. PRINTSWITCH is a compact module that plugs onto the parellel printer port of your TRS-80 and provides an edge connector for each of your two printers. It works with any two types of printers: dot matrix, daisy wheel, plotters, TRS-80 converted selectrics etc. Assembled, tested, ready to use with connector and instructions. For Model I or III (please specify). ONLY $59.00

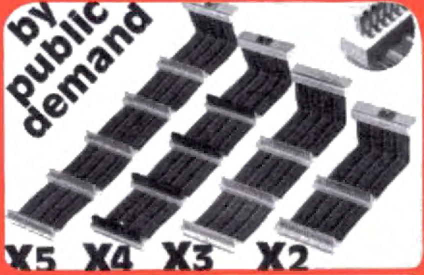



**SUPERIOR QUALITY REPLACEMENT & EXTENSION CABLES**

Highest quality cable and high force, gold plated contacts ensure the utmost in connection reliability.

| | | |
|---|---|---|
| ❶ KEYBOARD TO EXPANSION INTERFACE | | $21 |
| ❷ DISK DRIVE CABLE FOR 1 OR 2 DRIVES | | $32 |
| ❸ DISK DRIVE CABLE FOR 3 OR 4 DRIVES | | $45 |
| ❹ DISK DRIVE CABLE EXTENDER | | $22 |
| ❺ PRINTER CABLE EXTENDER | | $24 |
| ❻ 40 PIN BUS EXTENDER — 2 ft. | $22 | 4 ft. $24 |

Custom cable configurations are also available. Call us.




**YOU ASKED FOR IT "EXPANDABUS" X1, X2, X3 AND X4**

CONNECT ALL YOUR TRS-80 DEVICES SIMULTANEOUSLY on the 40 pin TRS-80 bus. Any device that normally plugs into the keyboard edge connector will also plug into the "EXPANDABUS". (The X4 is shown with protective covers included). The TRS-80 keyboard contains the bus drivers (74LS367) for up to 20 devices, more than you will ever need. Using the E/I, it plugs either between KB and E/I or in the Screen Printer port. Professional quality gold plated contacts. Computer grade 40 conductor ribbon cable.

X2 $29 — X3 $44 — X4 $59 — X5 $74

Custom configurations are also available, call us.


**ANALOG-80 A WORLD OF NEW APPLICATIONS POSSIBLE**

8 DIGITAL MULTIMETERS PLUGGED INTO YOUR TRS-80!!! Measure Temperature, Voltage, Current, Light, Pressure etc. Very easy to use, for example: let's read input channel #4: 10 OUT 0,4 Selects input #4 and also starts the conversion 20 A = INP(0) Puts the result in variable A. Voila! Specifications: Input range 0-5V to 0-500V. Each channel can be set to a different scale. Resolution: 20mV (on 5V range). Accuracy: 6 bits (1.5%). Port Address: jumper selectable. Plugs into keyboard bus or E/I (screen printer port). Assembled and tested. 90 day warranty. Complete with power supply, connector, manual. $139


**TIMEDATE 80 REAL-TIME CLOCK/CALENDAR MODULE**

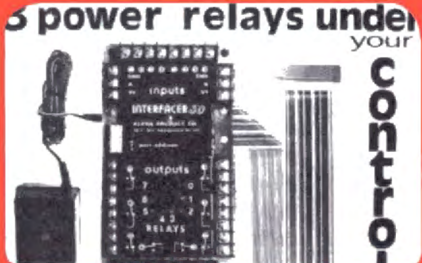
Keeps quartz accurate time for 3 years on 2 replaceable AAA batteries (not included). Gives MO, DATE, YR, DAY of WEEK, HR, MIN, SEC and AM/PM. Features "INTELLIGENT CALENDAR" and even provides for Leap Year. This compact module simply plugs into rear of Keyboard or side of Expansion Interface (may be tapped inside E/I). Includes cassette software for setting clock and patching to any DOS (including NEWDOS 80 1.0). Optional "Y" connector allows for further expansion. For Model I. Fully assembled and tested. Complete with instructions and cassette. ONLY $95.00. "Y" option add $12.00


**DISK DRIVE EXTENDER CABLE, FREE YOUR MINI-DRIVES.**

End the daisy-chain mess once and for all. Fits all mini drives: Percom, Aerocomp, Shugart, Micropolis, MTI, Vista, Pertec, Siemens, BASF. East to install, just remove the drive cover, plug in the EXTENDER CABLE and replace the cover.

Now you can change and move your drives without disassembly. Keep the cover on and the dust out. High reliability gold plated contacts, computer grade 34 conductor cable. Tested and guaranteed.

Get one for each drive — ONLY — $8.95




**INTERFACER-80** The most powerful Sense/Control module

- 8 industrial grade relays (single pole, double throw isolated contacts, 2 Amp @ 125 volts). TTL latched outputs are also accessible to drive external solid state relays.
- 8 convenient LEDs constantly display the relay states. Simple "OUT" commands in basic control the 8 relays.
- 8 optically isolated inputs for easy direct interfacing to external switches, photocells, keypads, sensors, etc. Simple "INP" commands read the status of the 8 inputs.

Selectable port address. Clean, compact enclosed design. Assembled, tested. 90 days warranty. Price includes power supply, cable, connector, superb user's manual. $159

## GREEN SCREEN WARNING

IBM and all the "biggies" are using green screen monitors. Its advantages are now widely advertised. We feel that every TRS-80 user should enjoy the benefits it provides. But **WARNING**: all Green Screens are not created equal. Here is what we found:

- Several are just a flat piece of standard colored Lucite. The green tint was not made for this purpose and is judged by many to be too dark. Increasing the brightness control will result in a fuzzy display.
- Some are simply a piece of thin plastic film taped onto a cardboard frame. The color is satisfactory but the wobbly film gives it a poor appearance.
- One "optical filter" is in fact plain acrylic sheeting.
- False claim: A few pretend to "reduce glare". In fact, their flat and shiny surfaces (both film and Lucite type) ADD their own reflections to the screen.
- A few laughs: One ad claims to "reduce screen contrast". Sorry gentleman but it's just the opposite. One of the Green Screen's major benefits is to increase the contrast between the text and the background.
- Drawbacks: Most are using adhesive strips to fasten their screen to the monitor. This method makes it awkward to remove for necessary periodical cleaning. All (except ours) are flat. Light pens will not work reliably because of the big gap between the screen and the tube.

Many companies have been manufacturing video filters for years. We are not the first (some think they are) but we have done our homework and we think we manufacture the best Green Screen. Here is why:

- It fits right onto the picture tube like a skin because it is the only **CURVED** screen **MOLDED** exactly to the picture tube curvature. It is Cut precisely to cover the exposed area of the picture tube. The fit is such that the static electricity is sufficient to keep it in place! We also include some invisible reusable tape for a more secure fastening.
- The filter material that we use is just right: not too dark nor too light. The result is a really eye pleasing display.

We are so sure that you will never take your Green screen off that we offer an unconditional money back guaranty. Try our Green Screen for 14 days. If for any reason you are not delighted with it, return it for a prompt refund.

A last word: We think that companies (like ours) who are selling mainly by mail should •list their street address •have a phone number (for questions and orders) •accept CODs (not every one likes to send checks to a PO box) •offer the convenience of charging their purchase to major credit cards. How come we are the only green screen people doing it?

Order your **ALPHA GREEN SCREEN** today. $12.50

## ALPHA Products

✓17

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD: ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

79-04 Jamaca Ave., Woodhaven, N.Y. 11421 ☎ Info and order: (212) 296-5916

# INPUT

## Magazine Mystery

When *80 Micro* was advertised, I decided to become a charter subscriber. On November 20, 1979, I sent off my check for $36 and eagerly waited for the first issue and the book of 80 programs. They never came!

Then, the February 1980 issue came. I was elated. Then the March issue arrived. I wrote for and received the book of 80 programs but still no January issue. I wrote to Wayne Green and soon the January issue arrived, but no letter of acknowledgement. I got my magazine so I didn't care: my problem was solved.

**Wait!**

No April issue. I wrote the subscription service. No reply, no magazine. I moved to my current address. I sent in a change of address. I wrote to the subscription service. No reply, no magazine. No big deal: I bought the magazine at the local hobby shop. No June issue, so I wrote the subscription service again. No reply, no magazine. I received the April issue as a forwarding from my prior address by the post office. Now I understood—it just took a little time to get things unraveled.

**Wait!**

No July issue, so I wrote the subscription service. No reply, no magazine. In August, I decided that this 50-mile trip to the hobby shop each month was becoming a pain, so I decided to send money for a new subscription and mark off the other money to experience. Anyway, the postage for writing letters was soon going to exceed my original subscription payment.

I subscribed again on August 4, 1980, at a cost of $18. I received the September issue in late September, and shortly thereafter, I received my October issue. Two weeks later, I received the August issue. Of course, I had already purchased a copy from the hobby shop. At last, my problems were solved. It was obviously my fault because I had moved.

**Wait!**

No November issue. I wrote the subscription service. No reply, no magazine. Since I had to go to the hobby shop to buy some things, I bought a magazine and wrote it off to experience. Needless to say, I became irritated.

No December issue. I find that my local Radio Shack dealer is now carrying *80 Micro* so I buy my issue there. Now I only have to pay the retail price for the magazine and travel three miles. I gave my problem some deep thought and decided that I would just continue to purchase from the local store and forget subscription service. The retail price is high but it is still cheaper than the subscription service!

Alas, my good fortune was limited. Due to the small demand for the magazine as an off-the-shelf item, the store was going to discontinue it. It seems that other TRS-80 users locally were having no problems with the subscription service.

What do I do now? I can go back to driving 50 miles for my magazine or I can try the subscription route again.

I decided to try a subscription one more time. Of course, I'm becoming better educated on how to do business with mail order outfits, as I had also been stung in buying software by mail. Do *not* send money until after you receive the product!

I tore out one of the handy-dandy subscription postcards in the last issue of my store-bought magazine, filled it out, and marked it *bill me*.

Success at last!

I received the November 1981 issue. I received the December 1981 issue. Then I received another December issue. Then I got another December issue. This computer stuff was beginning to get to me.

Not to worry!

I figured it out! This was the third time I had subscribed, so now I was going to get three copies each month. Wow! I gave away the extra copies and relaxed. I should have known better.

The January 1982 issue came. Then the second and third issues came, just as I figured.

**Wait!**

I then got copy #4, and then #5. This is a madhouse situation!

I received my bill for the subscription in the mail. I had them now. After spending 30 years as a detective in law enforcement, I knew I could solve any mystery. I would include a letter with my check. That way, if they cashed my check, I would know that they received my letter.

I'm becoming quite crafty.

Success! In February, the multiple issues stopped. In fact, everything stopped. No nothing! I should have known!

Off to the computer store 50 miles away to buy the February issue.

No March issue. Of course, I didn't expect one. Off to the computer store again. At least, I get my merchandise before I pay my money.

What do I do now?

Write another letter. Waste of time. Obviously, there is no actual person at the subscription service address—only a computer. That thought begins to bother me. Is it possible that there is no one at *80 Micro*?

My fellow employees are beginning to stare at me and ask about my health. They can't understand why I'm taking so long to complete my magazine fraud case, on which I have been working for over a year. I've been known to solve megabuck fraud cases in less time.

Six-figure crimes take only three to six months. The old man is slipping, probably be going out on disability pension soon; mumbles to himself about magazines, computers, no humans existing in New Hampshire.

Off to the computer store for the April 1982 issue when I learn everyone in town already has received theirs. Soon I must find a reasonable story to tell my wife, as she is also looking at me strangely. She cannot understand why a grown man would drive to a store to buy a magazine when everyone knows that you can subscribe by mail. I just hope she doesn't find out that the store is 50 miles away and that I also subscribe to the same magazine.

Before I've found an appropriate story to tell my wife, my April issue arrives in the mail. My wife asks why I received a copy in the mail when I had already bought that month's issue.

Today, I got the February issue. I can't believe this. I must be going insane. My wife is laughing hysterical-

*From Computer Plus to YOU . . .*

# PLUS *after* PLUS *after* PLUS

Model 16 128K
1 Drive $4199
2 Drive $4799

Color Computer 16K $249
w/16K Ext. Basic $335
w/32K Ext. Basic $449

Model III 16K $799
Model III 48K
2 Disk & RS232 c $1899

Okidata 80 $325
Okidata 82A $425
Okidata 84 $1029

Color Computer Disk Drive
Drive 0 $470 Drive 1 $315

Smith Corona TPI
Daisy Wheel $599

## BUY DIRECT *Here are just a few of our fine offers . . . call TOLL FREE for full information.*

### COMPUTERS

| | |
|---|---|
| Model II 64K | $2999 |
| Model III 4K LEV I | 599 |
| MODEL III 16K | 799 |
| MODEL III 32K | 856.50 |
| *MODEL III 32K | 831.50 |
| MODEL III 48K | 914 |
| *MODEL III 48K | 864 |
| Model III 48K 2 Disk & RS232 c | 1899 |
| Color Computer 16K | 249 |
| Color Computer 16K w/extended basic | 335 |
| Color Computer 32K w/extended basic | 449 |
| ‡Color Computer 32K-64K w/extended basic | 510 |
| Pocket Computer 2 | 230 |
| Model 16 1DR 128K | 4199 |
| Model 16 2DR 128K | 4799 |
| DT-1 Data Terminal | 599 |
| PT-210 Portable Terminal | 779 |

### MODEMS

| | |
|---|---|
| Lynx Direct Connect MI/MIII | 235 |
| Hayes Smart Modem II | 235 |
| R.S. Acoustic Coupler AC-3 | 134 |
| R.S. Modem I D.C. | 130 |
| R.S. Modem II D.C. | 210 |

### PRINTERS

| | |
|---|---|
| Daisy Wheel II | 1715 |
| DWP-410 | 1335 |
| Smith Corona TPI Daisy Wheel | 599 |
| Epson MX80 | 599 |
| Epson MX80 FT | 549 |
| Epson MX100 | 735 |
| CGP-115 | 199 |
| DMP-100 | 315 |
| DMP-200 | 599 |
| DMP-400 | 1029 |
| DMP-500 | 1569 |
| Microline 80 | 325 |
| Microline 82A | 425 |
| Microline 83A | 679 |
| Microline 84 Parallel | 1029 |
| P. C. Plotter Printer | 199 |

### DISK DRIVES

| | |
|---|---|
| R.S. Model III 1ST-Drive | 679 |
| Tandon 40 Track MI | 289 |
| Color Computer Drive 1 | 315 |
| Color Computer Drive 0 | 470 |
| Primary Hard Disk MII | 3999 |
| Primary Hard Disk MIII | 1999 |

### ETC.

| | |
|---|---|
| CCR-81 recorder | 52 |
| C. C Joysticks | 22 |
| 16K RAM N.E.C. 200 N.S. chips | 25 |
| 64K Ram Chips | 75 |
| Color Computer Flex D.O.S. | 99 |
| Brand Name Software • Send for listing. | |
| R.S. Software 10% off list | |

*Computer Plus New Equipment, with NEC RAM installed. 180 Day Computer Plus Warranty.

‡Color Computer 64K requires Disk 0 and Flex D.O.S.

**TOLL FREE 1-800-343-8124**

*We have the lowest possible Fully Warranteed Prices AND a full complement of Radio Shack Software.*

**computer plus**

P.O. Box 926
480 King Street
Littleton, MA 01460
617-486-3193

Write for your free catalog

✓18

Prices subject to change without notice.
Not responsible for typographical errors.

TRS-80 is a registered trademark of Tandy Corp.

AMERICAN EXPRESS VISA MasterCard

ly. She thinks this is goofier than an Abbott and Costello movie. She asks a dumb question: Why do you have to buy *that* magazine when you already subscribe to five others with no trouble?

I ignore her, because I have a solution.

I will write a mystery novel. With this plot, I will become world-famous. I have found the perfect mystery that no one can solve.

With all the money I will earn, I can move to Peterborough, NH, and retire. If I look carefully, I can probably find an abode on Pine Street, and then I can walk down to *80 Micro* each month and pick up my copy.

*Henry A. Hoover*
*P.O. Box 479*
*Elk Grove, CA 95624*

*Just kidding around, Henry. Heh heh.—Eds.*

## Model I Interface Worries Over

I am writing in response to a letter in the June/July issue of *80 Micro* entitled "Expansion Interface Worries." Tandy won't help, but Personal Micro Computers (475 Ellis St., Mountain View, CA 94043) can. The expansion interface created for the PMC, called the "Expander," can be used with the Model I. Tandy's expansion interface costs $497, but PMC's is only $415 and the case is neater.

*David Heyman*
*Conway, PA*

## Crusade Commentary

I would like to respond to Steven Gross' letter in 80 Input (September 1982, p. 26). Had Mr. Gross bought a Model I and been misled into buying a Line Printer VIII, he would understand our frustrations with Radio Shack. The LP VIII was designed for use with the Color Computer and its use with the Model I is severely limited—and it's not even "foreign" hardware.

*Martin W. Whitehouse*
*339 N. Sellars Mills Road*
*Burlington, NC 27215*

## Auto-Writer Replaced

Dan Robinson's broad review of word-processing products (September 1982) was a valuable and much-needed contribution.

A new enhanced system (Data-Writer) has replaced the Auto-Writer program reviewed by Mr. Robinson. Mr. Robinson mentions Auto-Writer's selection, form letter, and report-generating capabilities. Now with its own data-entry program, text editor, file manager, double-precision math procedures, two-level sort, mailing-label program, and 4500-record capacity, Data-Writer has become a stand-alone data base manager, though it can also read files written with a word processor.

Written for the Models I and III, Data-Writer is priced at $125. Further information is available from Software Options at 800-221-1624.

*J. R. White, President*
*Software Options Inc.*
*19 Rector St.*
*New York, NY 10006*

## No Losers

I disagree with Wayne's statement that "...about 100 years ago, Europe found that America was a great place to dump all of its losers..." ("80 Remarks," September 1982). Not so; while many Europeans wished they could come to America, only the most resourceful and brightest made it. Winners, not losers, arrived.

America's strength today is because of the industry of these people.

*Harry J. Cozen*
*1285 Westwind Circle*
*Westlake Village, CA 91361*

## Wayne's Wrong

Wayne Green's "80 Remarks" column (*80 Micro*, September 1982) attempts to undermine American technology and intelligence. The Japanese have only one edge over the Americans: money. Labor is cheaper in Japan and companies get grants from the Japanese government for research. Japan doesn't have to support a large military, so funds can be diverted to business.

There is no solid proof that the Japanese have an overall higher intelligence than Americans. Only one survey has been done in that field yet, which hardly makes it a fact. The Japanese who were tested were given Japanese versions of the American IQ test. Because of the language and culture difference, the Japanese may have had an unfair advantage.

Wayne also makes the observation that the Japanese are encouraged to take technical jobs through programs in high school while "most of our kids are spending their time on drugs, disco, and driving." That's a false statement (the Japanese people have a drug problem, too)—there are still a large number of competent and intelligent people who attend high school, college, and graduate school in this country.

Wayne also thinks that CP/M is the "foot in the door" for the Japanese industry. No matter how extensively CP/M is used as a system, an industry such as computers is marked by rapid change. CP/M will never be the final operating system, and neither will anything else. The Japanese were ready to enter the microcomputer market with or without CP/M by 1980 anyway.

The success of the Japanese microcomputers will have to be seen. While they do produce silicon chips successfully, this is no indication of their success with computers. They have little or no software support, their user-friendliness is in doubt, and they enter the American market at a time when many people are saying, "Maybe we should buy American."

If the Japanese have such an edge on the Americans, why do they feel it necessary to steal company secrets from American companies? No, Wayne, American technology is alive and well. America invented transistors; America invented integrated circuits, and America invented video tubes. The Japanese appear very skilled at reconstruction, but have not invented many new products.

Wayne, what you wrote about the America of the past is utterly wrong. Americans are not hard workers any more than anyone else. If they are, it is not because of the minority Protestant work ethic or the "hard working spirit of the pioneers." Neither of these are significant enough to account for what you claim they do. The "losers" that came from Europe 100 years ago were not losers. They came to this country and found jobs, were productive, and contributed greatly. (Remember, Einstein was an immigrant.)

The bosh about success in America

Special: SAVE $15
NEWSCRIPT + Mailing Labels
$139.95

# "IF YOUR PRINTER


# IS ON THIS LIST, NEWSCRIPT™ SHOULD BE ON YOUR COMPUTER."

### NEWSCRIPT'S exclusive print processor gives you total printer control.

NEWSCRIPT's exclusive print processor takes over where formatting leaves off with over 200 print processing and support features. NEWSCRIPT takes you beyond text formatting — beyond just printing text on paper. With NEWSCRIPT's Print processor you take full advantage of your printer's hardware and software features, creating an unsurpassed printed image. Text editing (the part you see), is only part of the job, printing is the other part — the part others will see.

### NEWSCRIPT controls over 80 popular printers.

You don't even have to know how your printer works — only what it can do. NEWSCRIPT will do the rest with such major features as underlining, right justified proportional printing, top and bottom titles, top/bottom left-right-center page numbering, chaining and embedded printing of any length file with disk spanning, fully formatted merging of form letters with selective editing, boldface, sub/super scripts, character substitution/translation, table of contents, indexing, hanging indents, paragraph numbering, line numbering, double width characters, italics, hard and soft hyphens, in-memory spooling, and many other features.*

### NEWSCRIPT'S text editor has sophistication to match its print processing.

Buffered key entry rates to 450 characters per second — you never drop characters. Windowing to 240 characters, block move and copy within and between files, definable auto save, "HELP" and "WHOOPS" commands, repeat and query last command, search and replace within column and line limits as well as globally, and an automatic interface to the ELECTRIC WEBSTER spelling checker (sold separately).

NEWSCRIPT'S 277 page manual contains an introductory tutorial with explanations of the beginner's most common needs, a "How to Section" to help when you're stuck, a fully alphabetized description of the commands with literally hundreds of examples, a topical index with over 1,300 entries, and a handy quick-reference card (naturally!).

Start getting the printed results only NEWSCRIPT can give you for $124.95. Requires TRS-80 Model I or III with 48K and 1 disk (minimum — 2 recommended)

**NewScript** T.M.

is available at computer stores, selected B. Dalton Book Sellers, and selected independent book dealers. If your dealer is out of stock order direct. Include $3.00 (domestic), $6.00 (Canada) for shipping and handling. Foreign residents add $15.00 plus purchase price, in U.S. funds.

TO ORDER, CALL NOW, TOLL-FREE:
(800) 824-7888, Operator 422
Calif: (800) 852-7777, Oper. 422
Alaska/Hawaii: (800) 824-7919
For technical information call:
(213) 764-3131, or write us. ⌐91

**PROSOFT**®

**Dept. C, Box 560, No. Hollywood, CA 91603**

---

NEWSCRIPT companion programs (sold separately): MAILING LABELS $29.95, DAISY WHEEL PROPORTIONAL $49.95 (not required for Daisy Wheel II), PENCIL & SCRIPSIT FILE CONVERSION $24.95, ELECTRIC WEBSTER (spelling checker and automatic correction) $149.50, GEAP (TRS-80 graphics — requires Epson MX-80) $49.95, DOTWRITER (Hi-res graphics — requires Epson MX-80/100 with Graftrax) $69.95, GEAP/DOTWRITER combination (requires Epson MX-80/100 with Graftrax) $99.95

---

***Dealers: NEWSCRIPT is distributed by IJG, Inc. (714) 946-5805***
****Some features work only if your printer has the mechanical capability.***

NEWSCRIPT trademark TTS Corporation   PROSOFT registered U.S. Pat. Office   TRS-80 registered trademark TANDY Corp.

not being related to intelligence is just that—bosh.

Wayne also states (in reference to America) that "big business, with the blessing of big government, takes over every successful entrepreneur." That system is precisely what gives the Japanese their edge. The Japanese government funds the large Japanese companies in a situation where government, business, and people benefit.

Remember, Wayne, American society and the computer field are much alike. Both are diversified, and standards never succeed. The CP/M operating system might be valuable for a few more years. The computer field changes so fast, I plan to save my old Model I and sell it to the Smithsonian in 10 more years.

*Howard Ullman*
*3533 Amberacres Drive*
*Apt. #108 E*
*Cincinnati, OH 45237*

## They Did It Again

I just received my September issue of *80 Micro* and was reading the editorial comments and reader letters. In your "80 Remarks" column you chide Tandy for their lies and deception.

Well, they did it to you again. On page 24 of the September issue is a section entitled "A Pat on the Back For Radio Shack." The first letter describes the superior performance of Radio Shack and Tandy Corp. What the letter does *not* state is that the author, John Von Tobel, was, during the time in question, an employee of Radio Shack. As an employee, he of course would have received the royal red-carpet treatment. Considering that his most recent assignment with Radio Shack placed him in a regular Radio Shack store right next door to our local Computer Center and repair facility couldn't have hurt.

Radio Shack has turned out some fine products (I am the proud owner of a heavily modified early 1978 Model I, that I would not trade for anything). However, when Radio Shack runs ads like they did when they turned over almost their entire Computer Center staff here, the light at the end of the tunnel may well be an oncoming train. Those ads were seeking "Bright, career-oriented sales people—no computer experience required." My question is, "If I were a businessman, would I trust the implementation of a computer system to such people?"

Until Radio Shack lives up to their word, I fully expect other makers to nibble away at the twin Tandy towers until they topple.

*Donald Stephens*
*1962 Pommel Ave.*
*Las Vegas, NV 89119*

## Extended Basic Tips

Here are some tips for Extended Basic Color Computer users. If you renumber an extremely large program accidentally, never press reset to regain the original state. That act leaves you with a demolished program, and no way to revive it. Having wrong GOTOs or GOSUBs is a much better fate than a program full of havoc.

If you make a mistake in the edit mode, and want to change it back, you could edit it out, or you could press reset and find that the line is in its original state no matter what you did to it. An easier way is to press 'a' while still in the edit mode. This works the same as the other two, but is easier to perform. Remember, once you enter an edited line, you can't do anything about it.

*Carl Huben*
*6929 Lee Crest Drive*
*W. Bloomfield, MI 48033*

## Rammer Fix

I made a few modifications to Mike Keller's "Rammer" (p. 296, August 1982).

The program now allows you to fire missiles with the up arrow key and you're provided with a shield for your ship that protects it from debris or collision three times. The shield appears and falls away when the ship is struck.

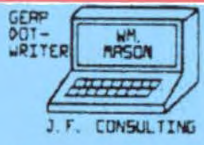
Line 20 checks to see if the Model III special characters are in effect and turns off the clock. Line 40 defines the missile trail (BT$) and the shield (CS$). In line 50 B$ is the missile and S is the number of shields available. Lines 270 and 280 are the actual missile fire routines.

Because I compressed the program (see Listing 1) and renumbered the

```
10 '          Rammer         Aug82 pg.296    80Micro
20 CLS:PRINT@0,CHR$(192):IFPEEK(15360)=32PRINTCHR$(21):GOTO20ELSEC
LS:CMD"T":GOSUB200
30 CLEAR200:DEFINTA-Z:R=195:P=0:KB=14400:LT=32:RT=64:HM=0:TR=226
40 BT$=CHR$(2):CS$=CHR$(176)+CHR$(32)+CHR$(176)+CHR$(26)+CHR$(24)+
CHR$(24)+CHR$(24)+CHR$(149)+CHR$(195)+CHR$(170)+CHR$(26)+CHR$(24)+
CHR$(24)+CHR$(24)+CHR$(131)+CHR$(131)+CHR$(131)
50 M=0:H=0:X=0:RP=15968:N=63:LE=15938:RE=15996:CL$=CHR$(30):S=3:B$
=CHR$(253)
60 ES$=CHR$(34)+CHR$(222)+CHR$(34)
70 CD$=":* *:"
80 DEFUSR=32755
90 PRINT@960,CHR$(R);" = 3     Enemy Ships Hit:          Enemy Ships
 Missed:";
100 PRINT@HM,CL$;:PRINT@1014,M;
110 P=RND(N)-1:IFP>60THEN120  ELSEPRINT@P,ES$;:M=M+1
120 POKERP,32
130 X=USR(X)
140 IFPEEK(KB)=RTANDRP<RETHENRP=RP+2:POKERP+63,TR
150 IFPEEK(KB)=8THENMP=RP:GOSUB270
160 IFPEEK(KB)=LTANDRP>LETHENRP=RP-2:POKERP+65,TR
170 IFPEEK(RP)=32ORPEEK(RP)=2THENPOKERP,R:GOTO100
180 S=S-1:H=H+1:M=M-1:N=N+5:PRINT@RP-15425,CS$;:PRINT@985,H;:PRINT
@964,S;
190 IFS<0GOTO230  ELSE100
200 FORX=32755TO32766
210 READY:POKEX,Y:NEXTX:RETURN
220 DATA17,191,63,33,127,63,1,128,3,237,184,201
230 CLS:FORE=1TO30:PRINTSTRING$(64,(160+E));:NEXT:CLS:PRINT@395,"Y
ou scored ";H;" hits out of a posible";M
240 PRINT@530,"Your Kill Ratio is";(H/M)*100;"%"
250 PRINT:INPUT"Care to play again ";A1$
260 IFA1$="Y"ORA1$="y"THENRUNELSECLS:END
270 IFPEEK(MP)=32ORPEEK(MP)=2PRINT@MP-15360,B$;:POKEMP+64,2:MP=MP-
64:IFMP<15360RETURNELSE270  GOTO270
280 POKEMP+64,2:PRINT@MP-15362,CD$;:H=H+1:M=M-1:PRINT@985,H;:RETUR
N
```

*Program Listing 1. Rammer Fix*

# NOW - TRUE PROPORTIONAL PRINT ON YOUR EPSON




Another **GEAP** expansion module for Epson Graftrax Owners
**AND NOW — TRUE PROPORTIONAL PRINT ON YOUR EPSON!**

# GEAP DOT WRITER

EPSON GEAP RCM COMPUTERS DOT WRITER TRS80

**FEATURES: GEAP CHARACTER FONTS • CREATE HI-RES DRAWINGS WITHOUT TRS-80 MODIFICATIONS OR PROGRAMMING KNOWLEDGE • CREATE OR MODIFY FONTS • PRINT TIME OPTIONS SUCH AS MAGNIFY, DOT SPACING CONTROL, REVERSE CHARACTERS, UNDERSCORING, TRUE PROPORTIONAL PRINT • SUBSCRIPT, INTERMIX FONTS AND HI-RES GRAPHICS ON THE SAME LINE • EXPANDED PRINT AND MORE!**

ACTUAL CHARACTER FONTS/SIZE

ABCDEF
ABCD abcd
ABCD abcd
ABCDEabcde
ABCD abcd
MINICUBES
SMALL ENHANCED
FLAT FACED PRINT
Plain Print
MICROPRINT ABCD12345

THE EXAMPLES ABOVE ARE JUST SOME OF THE SUPPLIED FONTS!

ABCDEF
ABCDE
ABCDEFGH
ABCDEFG
ABCD1234
ABCDabcd123
ABCDabcd123
ΔΣ ¹⁄₂ ¹⁄₄ ¹⁄₃ ¹⁄₆
ABXΔEΘΓHIKΛMZ

THE EXAMPLES ABOVE ARE JUST SOME OF THE MANY EXTRA FONTS!

## True Proportional Print

They said it couldn't be done! But you're looking at the True Proportional print from an Epson MX80 (printed in plain Epson at 1e font, but reduced to fit the ad). You can have type styles like those displayed (and many more) and you get user defined underlining, multiple underlining, expanded print and Hi-Res drawings. All of this and more in an easy to use program that will print the ASCII files created by NewScript or your processor. Simply follow NewScript dot command format and include our special commands. Then let Dot Writer take over. You haven't used your Epson until you've used GEAP and Dot Writer. Dot Writer comes with 11 fonts and additional disks of 7-10 fonts each are available at additional cost.

Feature packed, Dot Writer allows underlining each word or underlining words and blanks. You can even specify how many underlines you want. IT IS ALSO POSSIBLE to mix fonts with the ALTERNATE FONT command, as I have done here. Proportional printing even works with most of our special fonts. Finally, you can take full advantage of the power of your Epson printer.

**SAMPLES - High Res and Modified Characters**


NOTICE

This is what we consider the most powerful word processing station available for the TRS-80 Model I/III.

**GEAP 2.1** used to create low resolution graphics for the screen or for printout. GEAP 2.1 is also used to create the high resolution graphics, (together with Dot Writer), which you can integrate directly into your text.

**Dot Writer 1.5** used to create high resolution graphics for your Epson printer (with the aid of GEAP 2.1). Dot Writer also includes the **DOTPRINT** program which will print ASCII files created by your word processor but will give you the added power of our commands. This added power includes **TRUE** (not simulated) **PROPORTIONAL** print and you can use this feature with normal Epson style print, most of our special fonts, or fonts that you may create. You can intermix fonts on a line, or intermix HI-RES drawings with text, underscore words, or words and spaces and choose multiple underlining. You can expand characters or words and intermix them as you desire. The list of powerful commands goes on and on!

**NewScript 7.0** is the word processor we recommend. Its EDIT feature makes the creation of ASCII files, ready for DOTPRINT, a snap. NewScript is the best word processor available and it is fully integrated with GEAP and DOT WRITER. Both GEAP and DOT WRITER can be entered from the NewScript Menu. DOTPRINT allows reentry into NewScript after it completes a print operation.

**Electric Webster** is the finishing touch, also fully integrated with NewScript, (or it can be easily integrated with your word processor). Electric Webster eliminates the spelling errors that plague us all!

The perfect and complete word processing station. GEAP 2.1, Dot Writer 1.5, NewScript 7.0 and Electric Webster. More power than you could have dreamed!

### Additional Products

In addition to our own products, we also carry the following fine software.

**NewScript 7.0** - NewScript is the best and most powerful word processor available for the TRS-80. **$124.95**

**Electric Webster** - Cornucopia Software's top-notch spelling checker. A 50,000 word dictionary that you can customize to your needs. Error free hyphenation and unique grammar checking are extras that make Electric Webster a must for your work station! With correction feature only **$149.50**

**Faster** - Also a fine product from ProSoft. Faster optimizes BASIC code. **$25.95**

**Trashman** - A utility that reduces string compression time by as much as 95%. **$39.95**

**Quick Compress** - A real space saver from ProSoft software. **$19.99**

**RPM** - A real time disk motor speed analysis program. Keep your disks running perfectly! **$24.95**

**GEAP** - The **ULTIMATE** in TRS-80 Graphics. **GEAP 2.1** has instant graphic letters. Create your own screen graphics easily by: Magnifying, Reversing, Multiplying, Rotating, Merging and much more. Then let **GEAP** CREATE a BASIC program to recreate your graphic masterpiece! Numerous EXPANSION modules give **GEAP** limitless power. EPSON /OKIDATA/ Radio Shack LPV/IV printer block graphics; QUICKCURSOR with 2 speed controllable cursors, instant line, rectangle and circle. **NEWSCRIPT** interactive expansion module. Much more! There is **NO** other graphic utility on the market that is as POWERFUL, VERSATILE and EASY TO USE as **GEAP 2.1** **$49.95**

**DOT WRITER 1.5** - The **UNDISPUTED** leader in HIGH RESOLUTION graphics. Numerous Special Hi-Res Type Fonts for your Epson MX 80/100. **CREATE YOUR OWN** type fonts or HI-RES graphics. No hardware or modifications needed! **AND NOW, TRUE PROPORTIONAL PRINT, UNDERLINING, EXPANDED PRINT, AND MUCH MORE** using your Epson Printer. AND proportional print works with almost all of our **SPECIAL** type fonts! **DOT WRITER 1.5 + GEAP 2.1**, turns your TRS-80 and EPSON into a Hi-Res Graphic **TYPESETTER!** Requires 48K Disk and Epson MX 80/100 with Graftrax 80/PLUS. GRAFTRAX available for easy home installation.

| | |
|---|---|
| **GEAP 2.1 + DOT WRITER 1.5** | **$99.95** |
| **DOT WRITER 1.5** | **$69.95** |
| **Additional Disks with 7-10 fonts each.** | **$30.00** |

MasterCard VISA

**PRICES END JAN 83**

**JF Consulting**

West Coast
74355 Buttonwood
Palm Desert, Ca. 92260
MOD I/III (714) 340-5471

East Coast
221 Hirschfield Dr.
Williamsville, NY 14221
(716) 634-3026

✓ 273

lines, it would probably be easier to type it in than to change the original.

*Dan Clemens, Proprietor*
*Dan's Computer Works*
*Route 1 End Road*
*Summerville, OR 97876*

*How popular is "MONEY DOS"? These four letters came in on the same day.—Eds.*

Just a note to let you know how much I enjoy "MONEY DOS" by J.M. Keynes. His column is fascinating and profitable.

*Philip Raczka*
*10076 Belmeadow Drive*
*Twinsburg, OH 44087*

Since the first installment of "MONEY DOS" by J.M. Keynes I have been hooked by the clear, concise, readable text. I urge you to continue to carry "MONEY DOS" in the future.

*Frank Westphal*
*310 E. Philadelphia St. SP. 162*
*Ontario, CA 91761*

I enjoy your magazine but if it got down to the real nitty-gritty, I'd have to say "Keep the magazine and send me MONEY DOS."

*Ace Nace*
*7105 Turner Terrace*
*Fort Worth, TX 76118*

I am interested in investment programs and feel that "MONEY DOS" by J.M. Keynes has had several very interesting and informative topics during the last several months. Please add more business investment articles and programs in future issues.

*James H. Ferris*
*3328 High Vista*
*Dallas, TX 75234*

## Bittersweet Coverage

The editorial coverage on our Electric Pencil System (EPS-80) was bittersweet. The subhead that stated the trouble with a ROM-based word processor was that it could not be changed, did not have any qualifiers. Three points should be made in this regard.

● The Electric Pencil program is a user-proven program that has withstood the test of time. There are newer, better programs, but none as easy to learn.

● Any ROM-based program can be replaced by another ROM-based program.

● Since the EPS-80 is also a microcomputer (Z80-based) with Basic built in, there are other word processor programs available to operate in this environment.

Our EPS-80 is an economical, easily learned, and highly flexible system for the person who needs word processing primarily and computing power secondarily.

The EPS-80 can be upgraded to a disk-based system when the user is ready for more compatibility. Also, commercially available (and user installable) kits provide for CP/M software compatibility.

*R. R. Troxell*
*Director, Product Management*
*Personal Micro Computers Inc.*
*475 Ellis St.*
*Mt. View, CA 94043*

## Medical Software Support

Philip Mills' ("Medical Opinion," August 1982) article on the Medical Office System by Windham Software is a bit too glowing. I purchased the MOS, and while I agree that it is a step in the right direction I disagree that all is well.

It's slow, and the procedure fee column would not accept a charge over $999.99. When I called Windham to discuss this glitch, I was told that Dr. Hayes was not available. I called back and was told that he was gone for the day. I called again and was told that he was gone for the week. I then wrote a letter and now, two months later, am still waiting for a reply.

This contradicts the statement that "support is excellent."

*Leon M. Morrison, M.D.*
*#9 Medical Park*
*Morehead City, NC 28557*

## Dr. Mills Replies

*The Medical Office System was written by an internist and is more suitable for an internal medicine clinic. Dr. Morrison's clinic is surgically oriented and would find it less satisfactory.*

*It should also be pointed out that the program is brand-new. If Windham Software is responsive to physician feedback, in a few years they will be marketing a very satisfactory program. I suspect that they are experiencing the realities of software marketing and its growth pains, and have had to adjust to the time demands of selling software.*

*Philip R. Mills, M.D.*
*3000 Halls Ferry Road*
*Vicksburg, MS 39180*

## Windham Software Replies

*We disagree that the system is slow; for a micro system, response time is extremely fast. It takes an average of two seconds to put any one of 3,750 patient records on the screen. We do not know of any other micro system that can beat this timing.*

*The problem with fees higher than $999.99 has been corrected. The system now accepts fees up to $9999.99. We sincerely hope that inflation will not force us to increase the maximum fee amount for the next few years.*

*John R. Hayes, M.D.*
*Windham Software Inc.*
*29/31 Ivanhill St.*
*Willimantic, CT 06226*

## Hidden Routines

Here is another footnote to Hugo Jackson's fine article "Memory Size?" (*80 Micro*, April 1982). Burt Fenner noted that space for Z80 Assembly-language programs can be reserved in Basic remark statements as an alternative to storing them in string variables (80 Input, August 1982, p. 16). Mr. Fenner pointed out that remarks are not visited by the Basic "garbage collector." Mr. Fenner also noted that once the Z80 codes have been POKEd into the remark statement, the POKE routine may be deleted along with any data statements.

There is a third advantage that Mr. Fenner may have overlooked: The Basic program, shortened by removal of data and POKE statements, with the Z80 program in its remark, can be CSAVEd and CLOADed, even edited and listed! The listing of the remark will appear strange, since the Z80 codes will often refer to control and special characters.

There is a slight penalty if any zero codes occur in the Z80 routine: CLOAD will end the remark at the first zero code in the Z80 program. The next-line pointer that uses the first two bytes of the remark in memory must be corrected. If the Z80 routine uses less than 100 bytes, then the correction value for the Model III is the length of the Z80 routine. Program Listing 2 illustrates this method.

Statement 2 reserves 30 bytes for Z80 codes (even though the example routine uses only 12 bytes). Statement 1 restores the next-line pointer for statement 2 as if the Z80 routine included a zero code. Statement 35 copies the data values into the remark, while statements 40–80 display a byte-by-byte map of statements 1 and 2. Statement 90 executes the example routine, which simply draws a white bar near the bottom of the screen.

After the program has been run at least once, it can be reduced as shown in Program Listing 3 and CSAVEd.

A tip of my hat to Burt Fenner for this sly way to conceal a Z80 routine in a Basic program!

*David B. Dillon*
*Baederwood Lane*
*Derwood, MD 20855*

```
1 POKE 17400,30
2 REM 123456789012345678901234567890*
3 REM Above REMARK is 30-byte Z80 program buffer
6 REM Address 17400 in statement 1 is beginning of statement 2
        in a Model III. Use 17144 for Model I.
7 REM Value 30 POKEd in statement 1 is, coincidentally, the
        length of the Z80 program area reserved in the REMARK of
        statement 2.
8 REM If the Z80 program contains no zero-codes, statement 1 is
        not needed. Reduce the FOR arguments in statement 35 by
        15, to 17391 and 17390+length, respectively. Also reduce
        the POKE value in statement 90 by 15, from 254 to 239.
10 DATA  33,128, 63, 62,191,  6, 64,119, 35, 16,252,201
20 REM Above is a 12-byte Z80 program that draws a white line
        near the bottom of the screen.
35 FOR J%=17406 TO 17417: READ I%: POKE J%,I%: NEXT J%
37 REM Statement 35 POKEs the DATA values into the REMARK.
40 J%=17385
50 I%=PEEK(J%)+256*PEEK(J%+1): K%=PEEK(J%+2)+256*PEEK(J%+3)
60 PRINT J%,K%,I%
70 FOR L%=J%+4 TO I%-1: PRINT L%;PEEK(L%),: NEXT L%
80 PRINT: PRINT: J%=I%: IF K%<2 GOTO 50
85 REM Statements 40-80 display a byte-by-byte map of state-
        ments 1 and 2. Statement 90 executes the Z80 program
        placed in the REMARK.
90 POKE 16526,254: POKE 16527,67: PRINT USR(0)
95 REM Type the program and RUN it once to load the REMARK.
        Then delete every line except 1, 2 and 90. Add other
        lines if you wish, then CSAVE. The Z80 program will be
        recorded as statement 2, and will CLOAD.
96 REM If the Z80 program contains zero-codes, it will not
        LIST properly after a CLOAD until statement 1 has been
        executed, either from the keyboard or in a RUN.
97 REM Long Z80 programs can use a relative jump (JR) from the
        end of one REMark to the start of another.
```

*Program Listing 2*

```
1 POKE 17400,30
2 REM !END?>USING@w#
90 POKE 16526,254: POKE 16527,67: PRINT USR(0)
```

*Program Listing 3*

## Eds. Speak Out

*POKEing machine language into remark statements is a great idea, but in our opinion, if it has a zero, POKE it into a string, or high memory.*

*Listing 2 works for a Model III under cassette Basic. The author could*

# Convert your TRS-80*
# into a World Class Computer

ALSO... UNIQUE RED & BLUE CRT'S AVAILABLE NOW!

NOW ALSO AVAILABLE FOR MODEL I AND OTHER 12" MONITORS

## that
## REDUCES EYE FATIGUE AND DOESN'T FLICKER

## ...with LSI's new *Soft·View*™ Replacement CRT...

The black & white "TV Screen" CRT (picture tube) which came with your TRS-80*model II or III is an inexpensive rapid "P4" Phosphor CRT intended for TV use. The display is actually strobing 60 times a second. No amount of "green plastic" will stop this strobing or eliminate the eye fatigue it causes. But a new *Soft·View*™ CRT display tube with a slower decaying, colored Phosphor will.

- Available in slow-decay green (similar to new IBM* and APPLE III*monitors) or medium decay "European Orange" (easy on the eyes, elegantly beautiful, and the standard for CRT displays in Europe)
- Leaded glass stops X-ray emission
- Optional Anti-Glare Frosted Glass available to reduce eye strain from glare
- Easy installation — tube comes with pre-mounted hardware
- 30-Day Money-Back Guarantee
- Ideal for Word-Processing & Programming, fast enough for Games & Graphics
- Finest quality double-dark glass and phosphor fields make the letters seem to be coming out of black space

## Try This Test:

LSI's new *Soft·View*™ CRT

Turn the brightness control on your TRS-80*all the way up. Wave your hand up and down in front of the screen. See how jerky it seems? Just like in front of a strobe light! That's because the screen actually is strobing at you. A slower-phosphor CRT will reduce that troublesome strobe effect. That's why most of the newer monitors, from IBM* to Apple III* are using the new slow-phosphor CRT's.

LSI SYSTEMS *Soft·View*™ CRT's: 80M-1182

- ☐ #GN42 Green Phosphor $79.95
- ☐ #GN42G Green Phosphor with anti-glare $89.95
- ☐ #OR34 Orange Phosphor $89.95
- ☐ #OR34G Orange Phosphor with anti-glare $99.95

**ADD $7 FOR PACKAGING AND UPS SHIPPING**

Langley-St.Clair Instrumentation Systems, Inc.

✓462

To Order Call: 1-800-221-7070

Dealer Inquiries Invited

132 West 24th Street, New York, N.Y. 10011 212-989-6876

*IBM*, APPLE* and TRS-80* are trademarks of IBM, APPLE Computer & TANDY Corp.


The Official
FROGGER™
by SEGA®
The popular arcade game for TRS 80® Model I & III
TRS 80® Model I & III
$19.95 Tape
$22.95 Disk
Joystick compatible. Frogger is available from your local software dealer or order direct.
All orders by VISA, Mastercard or check add $1.50 shipping. Indiana residents please include 4% sales tax.
The Cornsoft Group
6008 N. Keystone Ave., Indianapolis, IN 46220 (317) 257-3227
and ™ designate trademarks of Sega Enterprises, Inc.
© 1981 Sega Enterprises, Inc.
TRS 80® is a registered trademark of the Tandy Corporation.


**Includes exclusive challenge mode.**

Bounceoids come crashing down from space — and rebound across the screen. Blast them or they will smash you. You have a shield but its lifespan is limited so use it sparingly.

The commotion will attract alien natives with poison darts, off-world snakes and shaking bugs. Be on your guard. The longer you survive the harder it is to simply exist. Bounceoids start dropping from the sky in tiny clusters or with constantly changing outer dimensions.

Reach the exclusive challenge level and face the flying space flock in a tense challenge of strategy, coordination and targeting skills. Joystick compatible.

$15.95 tape, $19.95 disc plus $1.50 shipping and handling charges. MasterCard & Visa orders accepted. Available at your software dealer.

**The Cornsoft Group** 308

**6008 N. Keystone Ave., Indianapolis, IN 46220 • (317) 257-3227**


## Become a "Crazy Painter" and create a masterpiece ...if you can.

A mischievous puppy, snakes and poisonous turpentine buckets force changes in your painting. In higher skill levels, "paint eaters" start chewing up your work. Suddenly, everyone's a critic. Can you overcome them?

And on top of it all, in the ***Exclusive Challenge Mode,*** you have to catch an army of runaway puppies. Crazy Painter is ***Joystick compatible.***

$15.95 tape, $19.95 disc plus $1.50 shipping and handling charges. Indiana residents add 4% sales tax, Master Card or Visa orders accepted. Also available at your software dealer.

**The Cornsoft Group** ~319

**6008 N. Keystone Ave., Indianapolis, IN 46220 • (317) 257-3227**

*have used memory locations 16548 and 16549 to find the beginning of line 2. The following line will do it.*

```
1  MA = PEEK(16548) + 256*PEEK(16549):
MA = PEEK(MA) + 256*PEEK(MA + 1)
NA = PEEK(MA) + 256*PEEK(MA + 1):STOP
```

*From the command mode, type:*

```
?MA,NA,NA – MA
```

*MA is the address of the first byte of line 2. NA is the address of the first byte of line 3. NA–MA is the number of bytes in line 2.*

*Now, for use at run time, change the stop statement in line 1 to:*

```
NA = MA + nnnnn:POKEMA,NA – INT
(NA/256)*256:POKEMA + 1,INT(NA/256)
```

*Change the nnnnn to the value you got for NA – MA. Using this change ensures that the code you write will work for all systems.*

*Change line 35 to read:*

```
FOR J% = MA + 6 TO MA + 17: READ
I%: POKE J%,I%: NEXT J%
```

*Change line 40 to:*

```
J% = PEEK(16548) + 256*PEEK(16549)
```

*For cassette Basic change line 90 to:*

```
MA = MA + 6:POKE 16526,MA –
INT(MA/256)*256:POKE16527,INT(MA/256)
```

*For Disk Basic change line 90 to:*

```
DEFUSR = MA + 6
```

—80 Micro *technical staff*

## Straight, Not Flip-py Answers

I believe that all floppy disks are coated on both sides with the same magnetic medium, to reduce or eliminate warpage. You can punch two new read holes and a write-protect slot and make both sides of the disk usable. This modification makes the disk a flippy and it can be read and used on both sides.

The advertised price for flippy disks is about 25–33 percent above the cost for single-sided disks—why?

If the disk jacket contains one set of read holes and one write-protect notch, or it contains two of each of the above, it's all done at one punching at zero additional cost.

Also, your magazine and most others fail to deal with the magnetic medium. There are variations in manufacture, quality control, and so on, and there are standards (ANSI or others) so let's have some articles on the basics and straight answers on the disk prices.

*R.M. Sanford*
*12787–139th St. N.*
*Largo, FL 33540*

## My Savior

Thanks *80 Micro* and "80 Remarks" for saving me just in the nick of time. I was all set to buy a TRS-80 Model II when I read in your fine magazine that the Model II is obsolete. I did go out and buy an Apple. It's a great machine.

*Harry J. Standish*
*5216 NE Sandy Blvd.*
*Portland, OR 97213*

*Do we detect a note of sarcasm?*
*—Eds.*

## Truth in Advertising

Firms that advertise in *80 Micro* and other publications should live up to their ads. In the July issue of *80 Micro* a retailer advertises "we will not be undersold" and yet their prices are considerably higher for some hardware than other firms advertising the same equipment in the same issue of the magazine. When asked why they made such a statement they replied that it was an advertising come-on to get customers. I advise potential customers of any firm to price shop for equipment and services before buying.

*Walter Steesy*
*Interlaken, NY 14847*

## Error Trap

Line 60060 of Kurt Leafstand's "QuickCalc" program (October 1982, p. 120) is the end of the listing; it does not continue.

Randy Hawkins was incorrectly given credit for the program "Enter The Dragon" in Proof Notes (August 1982). The credit should have gone to Brice Hadlock. We stand corrected.



## UTERM Fix

I really enjoy using Bob Hart's UTERM program (*80 Micro*, June/July 1982, p. 128). However, if you try to use this program and have an early keyboard, you may run into some problems.

My keyboard has the old ROMs and the famous "quick and dirty" audio fix from Radio Shack. My board contains the small XR2 mod to help the volume sensitivity problem using cassette tapes. This improvement will not allow anything above or below 500 baud to enter the computer.

Since UTERM uses the cassette port and is set for 300 baud, the program will transmit but not receive using Bob's software, the cassette port, and the Shack's mod. To overcome this deficiency, a small DPDT switch must be added to switch out the XR2 mod when using the UTERM program. It is surprising to note that Tandy sells Model I modem software that uses the cassette port, but their own equipment is not compatible when you are stuck with an old keyboard!

Anyone wishing to learn more about the switch installation may get a copy of the instructions by sending an SASE to me. It takes about 10 minutes total time to have UTERM running on your system.

*Richard C. Walter, Jr.*
*824 East Chestnut St.*
*Lancaster, PA 17602*

## Two Finger Typer

I would like to point out Lazy Writer's ease of use for a handicapped person limited to the use of one hand. Just about all of Lazy Writer's commands, editing as well as print and directory commands, call for the entry of a single keystroke. Rarely are you required to use a control key at the same time, and when you do, the keys are well within range of two fingers.

*Sidney Bloom*
*114-A West Church St.*
*Frederick, MD 21701*

## Easier Directory

For Model III TRSDOS users, Basic's CMD"D:d" produces a directory listing that is often more convenient than DIR. The program below, DIREC/CMD, displays the directory just like Basic. This program uses the top fifteen bytes of memory and is written for a 48K Model III with TRSDOS. For a 32K machine (see Program Listing 4) change the addresses in lines 160 and 240.

*Carl Anderson*
*Wilmington, DE*

```
                00100 ;
                00110 ;         DIREC/CMD
                00120 ;         FROM 'TRSDOS READY' TYPE:
                00130 ;         DIREC :D
                00140 ;         WHERE D=DRIVE
                00150 ;
FFF0            00160           ORG     0FFF0H  ;FOR 32K USE 0BFF0H
4419            00170 DSPDIR    DEFL    4419H   ;DISPLAY DIRECTORY CALL
402D            00180 JP2DOS    DEFL    402DH   ;TRSDOS READY ADDRESS
FFF0 3A2C42     00190           LD      A,(4225H+7H)    ;GET 7TH BYTE
FFF3 DD217142   00200           LD      IX,4271H        ;THESE TWO LINES PUT A
FFF7 DD7700     00210           LD      (IX),A          ;INTO 4271H FOR DSPDIR
FFFA CD1944     00220           CALL    DSPDIR          ;DISPLAY DIRECTORY
FFFD C32D40     00230           JP      JP2DOS          ;BACK TO TRSDOS
FFF0            00240           END     0FFF0H          ;FOR 32K USE 0BFF0H
00000 TOTAL ERRORS
```

*Program Listing 4*

## MONEY DOS Fix

The programs in the September 1982 MONEY DOS column (p. 364) contained several small errors. Listing 1 was renumbered, correctly updating all GOTOs and GOSUBs but leaving references in remark statements unchanged. Listing 2 did not write an end-of-file marker on cassette files. Program Listings 1 and 2 correct these errors.

In addition, lines 470–530 of Program Listing 2 on page 374 were accidentally printed at the end of Program Listing 1 on page 372.

*80 Micro* regrets the errors.

```
170 REM DELETE 180 IF NO DISK
550 CMD"T":REM CHANGE THIS LINE TO 550 REM IF NO DISK.
590 REM 600-660 ARE FOR DISK USERS .  TAPE ONLY USERS
          MAY DELETE.
1140 IFD$="SUGAR"THEN E1=4: E2=10: E3=23: RETURN
```

*Program Listing 1*

```
50 EN$="EOF"
280 IF U$="" THEN IF SS=2 THEN PRINT#-1,EN$
285 IF U$="" THEN CLS:PRINT"I'M FINISHED": END
```

*Program Listing 2*

## Screenplay Errors

My article (Screenplay) in the September issue contains a couple of errors. On page 208, the third paragraph of How to Use Screenplay states, "If you want to write as though using a typewriter, press shift down arrow..."—it should read "...press shift, down arrow, left arrow..." as Table 2 correctly indicates.

The Loop function that flashes all 10 pages of text on the screen in a continuous loop for animation effect contains an error. Line 7000 should read:

```
FORP=1TO10:GOSUB9100:GOSUB5800:
GOSUB9800:IFS=1THEN1200ELSENEXT:
GOTO7000
```

The end of line 1820 was omitted. It should include:

```
ELSEIFI>126THENI=I+J
```

Jim Stauble of Des Moines, IA has shown me another way to code lines 9900 and 9910.

```
9900 F=0:FORR=1TO1:F=NOTF:IFF=-1
POKEV,IELSEPOKEV,PEEK(V+T)
9910 FORW=1TOJ:I$=INKEY$:
R=-(I$>""):IFRAND1RETURNELSE
NEXT:NEXT
```

This coding is shorter than the original but has no effect on the typing speed. Placing a subroutine immediately following the calling section can help the interpreter find the subroutine more quickly. Unfortunately, this subroutine is called by several sections of the program, so many references would have to be changed.

For those interested in this approach to editing and printing memory contents, a machine-language version of Screenplay is available. Called Coretext, it runs on disk or stringy, is tailored to the MX-80/Graftrax, has many more features, and is faster than the fastest typist. The cost is $10 prepaid or $5 if you provide disk or wafer and SASE. The same terms apply if you want me to send you a copy of Screenplay.

If you want to know what line 7000 does (the incorrect line 7000, that is), it copies all 10 pages of text memory into higher memory, starting at address 32768. It's used for transferring Screenplay text to the text memory area of disk Coretext.

*Warren Merkey*
*114 SE 1st St.*
*Gainesville, FL 32601*

## Bug Report

To correct "Save Our Ship" (*80 Micro*, August 1982, p. 174) Model I users should change the value 65535 in line 29 of Listing 1 and line 20 of Listing 2 to 65536. Also add a new line to Listing 1:

```
31 FX = FX + 65535 * (FX>32767)
```

Model III users should change the value 65535 in line 29 of both listings to 65536. Also add the same new line to the program as above.

*Randy Hawkins*
*6214 Hidden Cove*
*Corpus Christi, TX 78412*

## Pronouns vs. Prepositions

The September 1982 issue of *80 Micro* was great. Dan Robinson's comprehensive report was good except for a slip that he or your copy editor should have caught.

On page 123 of Mr. Robinson's review of Aspen Software's Grammatik program, he writes: "...and too many prepositions (especially I) may make the writer look like an egomaniac." A preposterous preposition if ever I've heard one! Don't you mean pronouns? How can a diddly little preposition (at, after, by, down, for, from, in, into, etc.) make a writer appear egomaniacal? Pronouns, especially the first-person singular pronoun (I), can indeed make a writer appear a little self-centered. However, the sample of Grammatik's text summary you cited in the article only lists prepositions; nowhere do we see a mention of pronouns. You computer guys do understand the distinction between a pronoun and a preposition, don't you?

*Jeff Shear*
*46 Wiggins St.*
*Princeton, NJ 08540*

*Oops.—Eds.*


NEW!
ESF III
NOW AVAILABLE!
SIMPLY AMAZING!
NEW LOW PRICE—$99.50
How else can you describe the *Exatron Stringy Floppy* system? You could say that it's an under $100, compact, reliable, robust, high-speed, computer-controlled, easy-to-use, well-supported alternative to disk drives, for a Model I or III TRS-80—*simply amazing!*
**Amazing Technology**
Based on a special endless-loop tape cartridge, called a Wafer, the ESF system was designed specifically for computer data storage. The direct-drive transport mechanism has only one moving part, and data is transferred to and from the tape at a rate of 7200 baud.
**Amazing System**
Thousands of ESF buyers have been amazed by 16K programs loading in less than 20 seconds; automatic verification of saved programs; up to 70K bytes, and 99 files, on a single Wafer; a ROM operating system (RAM based in Model III); no need for an expansion interface; and 1-year parts and labor warranty.
**Amazing Support**
With an ESF system you don't just get a piece of hardware, you get total support with hundreds of user workshops; dozens of high-quality, reasonably priced programs (such as Electric Pencil 2.0, Electric Spreadsheet, File Management System and Technical Word Processor); access to hundreds of FREE public-domain programs; an @NEWS user column in 80-US; @LOAD program magazine; and a toll-free information line.
**Amaze Yourself**
To see for yourself how amazing the ESF system is, or for more detailed information, call us toll-free at **800-538-8559** (inside California **408-737-7111**) and take advantage of our 30-day money-back return policy. Copies of the 80-page manual are available for $4.95 (which you can credit towards an ESF), and while you're on the line ask about our equally amazing 64K RAM/ROM board for the Model I.
exatron
excellence in electronics
exatron
181 Commercial Street,
Sunnyvale, CA 94086
TRS-80 is a trademark of Tandy Corporati
Stringy Floppy is a trademark of Exatron Corp

## Handicapped Handicap

If a golfer claims a handicap of eight it might be assumed that he is pretty good—unless it turns out that this handicap was computed using Michael Avery's system ("Two Strokes a Side," *80 Micro*, May 1982) instead of the official USGA system.

Any golf league, association, or club is free to use any system for assigning handicaps, just as it can adopt any set of rules for the play of the game. Scores and handicaps obtained under modified rules cannot legitimately be compared with those acquired with the standard rules. Articles such as Mr. Avery's reinforce the mistaken idea that handicaps are simply numbers assigned arbitrarily, with poor golfers getting big handicaps and good golfers low handicaps. USGA rules regarding handicaps are explicit, and differ drastically from Mr. Avery's rules and from those he describes as the USGA system.

The use of course par in place of course rating is simply not justified. Under the USGA system, a par 72 course can have a rating anywhere from 62.8 to 80.6. These are extremes —course ratings are generally two to five strokes less than par.

The look-up table that Mr. Avery uses (he gets the endpoints on the wrong side, does not reproduce the entire table, and overlooks the fact that SM might be equal to 358.8) is one that has not been used by the USGA since 1975. It is not true that this table includes the increased difficulty factors in lowering one's handicap as the handicap approaches zero; it was published for those who find the problem of taking 85 percent of a number and rounding to the nearest integer too mathematically demanding.

Avery's system differs from the USGA's in using the best five of the last 10 scores rather than the best 10 of the last 20, and in using nine hole scores instead of complete rounds.

It is not true that the USGA handicap system cannot be applied to those with fewer than five scores; while these handicaps are not official, they are computed and designated as temporary by adding the suffix T.

Mr. Avery asserts that his handicapping system "will do a credible job for a small golf league." If this is all that is desired, a much simpler system, not requiring a computer, could be devised.

*"I developed this program as an aid to golf leagues. . . ".*

A golfer's USGA handicap is defined as 96 percent of the average of the lowest 10 of his last 20 differentials, provided at least 20 scores have been reported. If M (the number of reported scores) is less than 20, then N, the number of differentials to be counted, is determined as follows: if $M \leq 5$, then $N = 1$, if $M = 6$, then $N = 2$, and if $M > 6$, then $N = M/2$, rounded to the next lowest integer if M is odd. Details, together with the current look-up table, can be found in the USGA magazine *Golf Journal* (Nov./Dec. 1980, pp. 19–22).

While the USGA system is far from perfect, all clubs assigning handicaps to their members should use a uniform system. Inability to adapt the standard system to a small computer is a poor reason for changing the system.

*B.J. Ball*
*Rte. 3, Box 372*
*Watkinsville, GA 30677*

## Michael Avery Replies

*I indicated in the article that a modified form of the USGA system was used. The method I developed does not differ drastically, as Mr. Ball states. I indicated that course par could be used in place of a course rating because many small courses don't have a course rating. Are these courses to be denied handicapping?*

*If Mr. Ball had checked the program he would see that the possibility of SM being greater than 358 is taken care of in line 1530.*

*When the program was written (about four years ago), it had the correct handicap look-up table. The new table can be found in the latest edition of the golf rules, section 6. In the same section, the statement "Fewer than 5 scores: no handicap" would seem to contradict Mr. Ball's statement.*

*The data storage routines I used were not very efficient. Using more efficient numeric string packing for cassette storage, as I now know, would greatly improve the storage performance.*

*The rules I used for determining the number of differentials used are identical to the USGA method, but scaled down for a maximum of 10 scores of course.*

*I developed this program as an aid to golf leagues who wished a local handicap. No attempt was made to make this an official handicap, nor do I think it necessary. All that counts is that the calculation be fair and consistent. I believe that the program accomplishes that goal.*

*—Michael Avery*

## Golf Fix

Helpful callers have identified a few errors in the coding of my "Callaway Golf Scoring Program" (*80 Micro*, May 1982, p. 300). The changes are as follows:

```
1220 L=0:L1=0:P1=0:IF SH<>1GOTO 1270
2380 PRINT X+Y,T$(X+Y),T1(X+Y)
2400 X=X+10:IF NP>X GOTO 2370
```

and delete line 2360.

To use the program with a disk system do not set the memory size, and delete lines 10–30 and 160–200. The screen print routine will not work, but if your DOS has such a feature you can use that instead.

*Rodger Wells*
*1008 Kehoe Drive*
*St. Charles, IL 60174*

## Donations Needed

Goodwill Industries is interested in acquiring computer equipment. We are particularly interested in Model I hardware and software. Our organization is a non-profitable, charitable corporation, and contributions are tax deductible.

If any reader is interested in making a donation to Goodwill, contact me for further information.

*Sidney L. Bloom*
*President, Board of Directors*
*Goodwill Industries*
*114-A West Church St.*
*Frederick, MD 21701*

## Need Programs

I am putting together a book of business programs for students in business education courses at the local technical institute. The accounting and business education students at this point only need to run programs.

The technical institute has Model IIs and Model IIIs. We hope to set up a micro lab next year. I would appreciate any and all programs that your readers would send for student use.

*David P. Senkpiel*
*Instructor, Business Education*
*Guilford Technical Institute*
*420 Manor Drive*
*High Point, NC 27260*

## Cheap Green Screen

There are several green screens on the market, but being a penny-pincher, I came up with an idea that works very well.

I went to the local office supply store and bought a dark green and a light yellow punchless report cover. I put the yellow inside the green and attached them to my screen with scotch tape. This combination gives a nice green color, and costs only 50 cents.

*Richard A. Little*
*407-15th Ave.*
*Sterling, IL 61081*

## For Sale

I have a new Quick Printer I for sale. It comes with the instruction manual and two rolls of paper. I am asking $300 or best offer.

*Zacharias Beckman*
*Pegasus Computer Systems*
*44040-6th Ave.*
*Paw Paw, MI 49079*

## More Music

Has anyone written a program similar to Radio Shack's Micro Music? I'm looking for a larger text buffer. I would like the program to include: starting at a specified point, notes shorter than eighths, an additional two octaves (one higher, one lower), and a softer "N"atural command that returns specified options (speed, tone, octave) to normal.

I don't want to integrate with Basic programs; this is for personal enjoyment.

*D. Griffith*
*2932½ W. Knights Ave.*
*Tampa, FL 33611*

*Jake Commander recommends Orchestra 80 from Software Affair.—Eds.*

## Need Wiring Diagram

Does anyone have the address of McClain and Associates Inc., Indianapolis, IN? They made and sold an interface called Selcon V and converted IBM 1980s to use on a TRS-80 Model I. They used to advertise in *80 Micro*.

I bought a unit and need a wiring diagram for the 40-pin connector that connects the IBM 1980 with the interface. Does anyone have a wiring diagram that tells what color wire goes to what pin on the connector?

I have written to them but the mail is returned and they don't have a phone listing in Indianapolis.

*Robert Hober*
*RR #3, Box 125*
*Holiday Island, AR 72632*

| Parameters For ESF Wafers | START.LENGTH.ENTRY |
|---|---|
| Galaxy Invasion MEM (23200) | @SAVE1,23296, 9472,32724 |
| Robot Attack MEM (18432) | @SAVE1,18432,14336,32686 |
| Eliminator MEM (18688) | @SAVE1,18688,14080,20692 |
| Meteor Mission | @SAVE1,17664,11777,17664 |
| Cosmic Fighter | @SAVE1,21760,10753,26073 |
| Super Nova | @SAVE1,24064, 8449,25491 |
| Attack Force | @SAVE1,17152,13570,30453 |
| Lunar Lander | @SAVE1,20352,12290,20352 |
| Defense Command | @SAVE1,18569,14136,18750 |
| The Count or Adventure Sampler | @SAVE1,17152,14592,17232 |
| Penetrator (Main or Editor) | @SAVE1,17408,15100,17408 |
| Starfighter (Main or Trainer) | @SAVE1,16918,15100,17116 |
| Pinball (Acorn) | @SAVE1,20480,12289,29696 |
| Swamp War | @SAVE1,17152, 5500,17520 |
| EDTASM Plus (Microsoft) | @SAVE1,17280,12035,17280 |
| Z Bug (Microsoft) | @SAVE1,17280, 6070,17280 |
| TLDIS (Disassembler) | @SAVE1,28672, 3351,28672 |
| COPSYS | @SAVE1,16969, 1012,17129 |
| Interlude (Over 16K) | @SAVE1,16640,16130,23040 |
| Gobbleman | @SAVE1,17408,13312,17408 |
| Australian Programs | |
| Ghost Hunter | @SAVE1,20992, 7925,21200 |
| Killer Beetle | @SAVE1,20480, 6588,26868 |
| Alien Cresta | @SAVE1,20992, 8501,23449 |
| Centipede Attack | @SAVE1,20992, 4131,21476 |

*Table 1*

## VisiCalc Templates?

Are there any VisiCalc user-developed programs for Investment Stock Analysis and Bond Interest Earnings, and so on? No doubt some experts have developed better programs than the average user and may not object to making them available.

I would appreciate any suggestions.

*Robert B. Foster*
*4909 Nasa Road 1*
*Seabrook, TX 77586*

*See* Desktop Computing *for an explanation of VisiCalc and its uses.—Eds.*

## ESF Copies

This information will be beneficial to anyone with an Exatron Stringy Floppy. I've enclosed a list of parameters to copy many popular machine-language programs onto Stringy Floppy. This information is tested for use on the Model I, Level II TRS-80.

From the Ready prompt type system. From the *? prompt type the file name (or first letter of the file name) of the machine-language program you're loading. After the program has loaded, the *? prompt reappears. Type /12345 to initialize the Stringy Floppy and return you to Basic.

Now type @SAVE1,start,length,entry. Start is the starting address, length is the number of bytes to move, and entry is the entry point. (See Table 1 for the parameters.)

Please don't use this information to pirate other people's programs.

Gobbleman will crash as soon as you answer the 1 or 2 player question. After the crash the computer reboots. From the Ready prompt type system. From the *? prompt type /17408. Gobbleman will run correctly.

*A. Tito*
*103 Lauren St.*
*Urangan, Queensland 4658*
*Australia*

## Missing Copy

I am writing to see if you can help me obtain a copy of the November 1981 issue of *80 Micro*. When I wrote in January to ask the subscription department for my missing issue, I was told it was no longer available and my subscription was extended one month. I then thought perhaps you kept a certain number for back issues, so I ordered a back issue, but again the reply was it was no longer available. Anything that you can do or suggest in obtaining the November 1981 issue would be greatly appreciated. I value each one of my *80 Micro* issues highly, and I don't want to be one short of a complete collection.

*Murray Jones*
*P.O. Box 698*
*Campbell River, B.C.*
*Canada V9W 6J3*

*The November 1981 issue is as rare as proverbial hen's teeth. As a matter of fact, even some of our editors don't have a copy. Is there a reader out there who can help Mr. Jones out?—Eds.*

## Modem Search

I am looking for a modem that can be set at 300 or 1200 baud, and software to enable me to download or upload to disk from CompuServe or The Source. I was told that it's cheaper to download at 1200 to disk than to download at 300 baud to my printer.

I own a Model III Level II, 48K, 2 disk drives and an Epson 80 FT printer.

*Terry Rubin*
*2928 Rockingham Circle*
*Orlando, FL 32808*

*See New Products, November 1982.*
*—Eds.*

## TRS-232 Newsletter?

I would like to contact users of the Small Systems Software TRS-232 interface for the Model I. Many users of this clever interface could benefit from a newsletter; in particular, I am interested in exchanging information on inserting the TRS-232's printing code into machine-language programs which print via the parallel or possibly RS-232 ports only.

Please contact me if interested, and let me know what programs you have converted to TRS-232 use.

*Sidney L. Bloom*
*114-A West Church St.*
*Frederick, MD 21701*

## Monitor or TV Screen?

I would like to purchase a color monitor that I can use with my Color Computer and with my next computer (whatever that is).

Does Amdek, NEC, Sanyo or one of the other color monitor manufacturers have a model that doesn't use RGB that would be compatible with my Color Computer, or is further modification needed?

*Samuel Baker*
*2823 Baker St.*
*San Francisco, CA 94123*

## More Memory?

I recently installed 32K worth of memory in my Model III (from 16K to 48K) with chips purchased from one of your advertisers (Computer Plus). Everything seems to function normally except that now the PEEK and POKE commands won't function at addresses greater than 32767, which is where my 16K ended! I can load and run programs into the new memory, the ?MEM command gives a 48K response, but PEEKing and POKEing into the new section gives me an OV error. I can't burn in the new chips and my warranty is quickly running out on them! Can anyone help?

Also, does anyone know a simple way to send special characters such as control codes to a printer using cassette Scripsit?

*J. William McDonald*
*4206 Trio Lane, Apt. #3*
*Louisville, KY 40219*

## Mission Aid

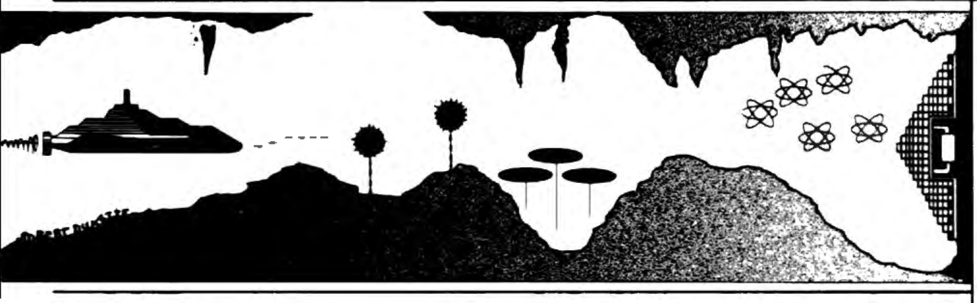
Our missionary organization (associated with the US Center for World Mission, Pasadena) uses a donated Model I, 48K with tape drive. We bought the TC-8 operating system and an ET12 interface, linking the system to an IBM Electronic 50 typewriter. Our main use is word processing, but we use a general ledger program adapted from *80 Micro*.

We want to manage our mailing list with the Radio Shack Cassette mail List, but there are problems. Anyone who has used the program knows it sorts very slowly. Has anyone successfully modified this program to speed up the sorting process for use with the TC-8 system? Finally, how can I get lowercase with the mail-list program?

If you can help or would like to contribute equipment or software, please write or call (213) 797-1937.

*J. Gene Adkins, President*
*Omega World Missions Inc.*
*1605 Elizabeth St.*
*Pasadena, CA 91104*

edited by Janet Fiderio



## Table of Contents


Sea Dragon . . . . . . . . . . . . . p.40
TRS-80 Assembly-Language Programming . . . p.41
Wolfbug . . . . . . . . . . . . . . p.41
Elementary Basic
Elementary Pascal . . . . . . . . p.44
Lisp/UOLISP . . . . . . . . . . . p.46
Quest . . . . . . . . . . . . . . . . . p.48
LSI Soft-view CRT . . . . . . . p.51
FED File Editor . . . . . . . . . . p.53
Bounceoids . . . . . . . . . . . . . p.54
The Disk Doctor . . . . . . . . . p.54
LCA-47 Lowercase Adapter . . . . . . . . . . . . . . . . p.56
Hypergate
Centurian & Patrol . . . . . . . p.60
Date-O-Base Calendar . . . . p.61
Drive Control Unit . . . . . . . p.62
Review Digest . . . . . . . . . . . p.63


*Our reviewers use a five-star rating system. One star represents the low end of this spectrum, while five stars represent the spectacular and high end of the spectrum.*

★★★★

**Sea Dragon**
**Adventure International**
**Box 3435**
**Longwood, FL 32750**
**Model I or III**
**$29.95 disk**

**by Eric Maloney**
***80 Micro* staff**

When Sea Dragon made its debut here at *80 Micro*, our Eliminator fans didn't like it, while everyone else did. This was curious, because at first blush the two games are similar. Both feature a horizontal scroll, and both involve destroying (or avoiding) objects rising from the bottom of the display. In fact, Sea Dragon looked like an Eliminator clone.

But it didn't take too many hours beneath the phosphor waves to realize that Sea Dragon has an aquatic rhythm that gives it an integrity of its own. It's as cool and casual as Eliminator is hot and furious, with enough of a kick to keep it challenging.

The object of Sea Dragon is to guide a submarine (presumably called the Sea Dragon, though the documentation isn't clear on this) through a series of underwater passages and destroy a nuclear reactor. During the trip (which covers 24 screens), you must blast or evade a variety of mines, stalactites, battleships, lasers, and enemy attack stations. You also must maintain your oxygen supply by surfacing occasionally, or risk total obliteration.

Each direct hit is worth from 50 to 500 points. The big banana, of course, is the reactor, which is worth 5,000 in the novice mode or 10,000 in the expert mode. Once you blow it away, you go back to the beginning, at a faster and more difficult pace.

Like most arcade games, Sea Dragon requires digital dexterity. You must learn how to move your ship over, under, and around the rising mines, when to speed up or slow down, when to surface for air, when to run or shoot, and how to avoid getting snuffed by the enemy attack stations.

When you figure all that out, you get to tackle the tough part—doing it while maneuvering through the tunnels. Touching the walls means complete annihilation (what else?), and the tunnels sometimes seem to go on forever. Finding a place to surface is cause for raucous jubilation, enemy mines or no. (This is definitely not a game for claustrophobics.)

At first, your scores will be pitifully low, which is just as well, because you won't want anything but to get through the tunnels. When you're finally good enough where points count, AI offers you their familiar scoreboard, which lets you (if you use disks) save the top 10 scores. This feature is terrific in places (such as the *80 Micro* office) where a number of people are using the same disk, and will extend its playtime considerably. The permanent scoreboard should be standard on any quality game, and other manufacturers would do well to follow AI's lead.

Despite Sea Dragon's many qualities, what sets it a notch above the pack is the organic nature in which it was designed. The authors conceived a scenario, and then gave it the unique contours and rhythms its contents suggested. Thus, while the pace would be excruciatingly slow in outer space, it is well-suited to Sea Dragon's underwater environment. (The expert mode, on the other hand, is frenetic enough to give your heart fibrillations.)

Sea Dragon is another example of an emerging aesthetic in the computer game world. Software authors are beginning to realize that a game needs to be internally consistent to stand the test of repeated plays. It doesn't work to simply state in the documentation that a blip on the screen is a space ship or a tank or a submarine. The game must invite the player to participate, to feel the action, to sense the flow. The world it creates must be complete and self-contained. This is possible only if the shape of the game supports and enhances its premise. Authors Wayne Westmoreland and Terry Gilman (Armored Patrol, Eliminator, and now Sea Dragon) are quickly developing a track record that shows they have the knack.

Sea Dragon's major drawback is the combination of keys it requires you to use. The back arrow decreases speed and the enter key fires missiles, and it's almost impossible to do both at the same time. Also, you use the up and down arrows to move vertically, and you won't find a position that lets you get to both comfortably. The result is that you'll often hit the wrong keys when you can least afford to.

All in all, Sea Dragon is worthy of the AI stamp. It's not as distinctive or original as AI's classic Armored Patrol, but it offers more than enough to be a welcome addition to most game libraries. It might even catch on with Eliminator fans. ■

★★★

***TRS-80 Assembly Language Subroutines***
**William Barden, Jr.**
**Prentice-Hall**
**Englewood Cliffs, NJ 07632**
**$18.95**

**by G. Michael Vose**
***80 Micro* staff**

Bill Barden has become the guru of TRS-80 Assembly-language programming. The many thousands of words he has written on this tricky subject have endeared him to a vast audience of programmers hungry for tips on technique. He has even mastered Assembly language for the TRS-80 Color Computer. And it appears that he's not done yet!

Evidence attesting to Barden's leadership pops up every day, most recently in a book called *TRS-80 Assembly Language Subroutines*. Inside, you'll find 65 subroutines, all written in Z80 Assembly language. They'll allow you to perform a multitude of difficult-to-program functions with speed and ease. The book is a veritable fountain of useful routines.

But wait a minute. When was the last time you needed to implement a 16-by-16 multiply operation in Assembly language? Or an ASCII-to-binary conversion? How about a multiple-precision BCD add?

This is pretty heady stuff! It seems that most of these subroutines, because they are very specialized or complicated, are useful only if you're a fairly sophisticated programmer. This is not evident, though, when you casually leaf through the first several pages.

Barden's attempt to make the book appeal to as large an audience as possible is confusing. I felt as though I had taken the wrong road in an adventure. It is obvious that a great majority of the routines in the book would be useful only to the kind of person who needs to implement a random number generator in Assembly language, for example, not a beginner's task. Yet the first two chapters of the book are a halfhearted attempt at introducing newcomers to Assembly language, a most difficult subject.

The writing in these first two chapters also belies the fact that Barden has been over this ground so many times that he has lost interest in this aspect of the subject. Nevertheless, he has thrown them in, apparently to justify selling this book to an audience of beginners as well as seasoned Assembly-language mavens.

In spite of this shortcoming, this is a useful book. Many of the subroutines are useful to the Basic programmer who wants to speed up graphics, dump graphics to a printer, perform some elementary animation (moving dots), or implement a time delay. Other routines will aid the Assembly-language programmer who hasn't yet figured out how to write a random number generator, how to read/write disk sectors or cassette records, or how to read RS-232 switches.

The best feature of the book is its format. Each subroutine is written in relocatable code, meaning it can be used anywhere in memory because it contains only relative memory addresses. Each routine is printed as an assembled source-code listing so that the routine can be entered into a machine with either an editor/assembler or a monitor like Debug. Plus, each routine is also listed as a Basic data statement so that the routine can be POKEd into memory from Basic. Barden and his editors have even computed and listed the checksum for each routine so that the programmer can verify his data entry.

The book is spiral bound so that it lies flat on your lap or desktop while you're typing in the code for a routine. There is an appendix summarizing the Z80 instruction set so that making changes to specific routines will be easy for the experienced programmer.

Barden offers all the routines in the book on disk—the cost will be disclosed to you if you express an interest by writing to him.

There is no question that books like this one have a place in the library of information that is now available for the TRS-80. Anything that makes programming the computer easier is appreciated. It seems that Bill Barden will evermore be ready to fill that need. ■

★★★

**Wolfbug 64K**
**Color Computer Upgrade**
**Atomic City Electronics**
**3195 Arizona Ave.**
**Los Alamos, NM 87544**
**Wolfbug with adapter card $55**
**64K RAM card without memory $44.95**
**64K RAM card with memory $144.95**

**by David L. Wasler**

Upgrading your 4K, 16K, or 32K Color Computer to 64K is a relatively easy modification when using Atomic City Electronics' 64K upgrade. The upgrade consists of two major components: a monitor called Wolfbug, which has 12 single-key commands, and a 64K RAM adapter card that reconfigures the old RAM socket

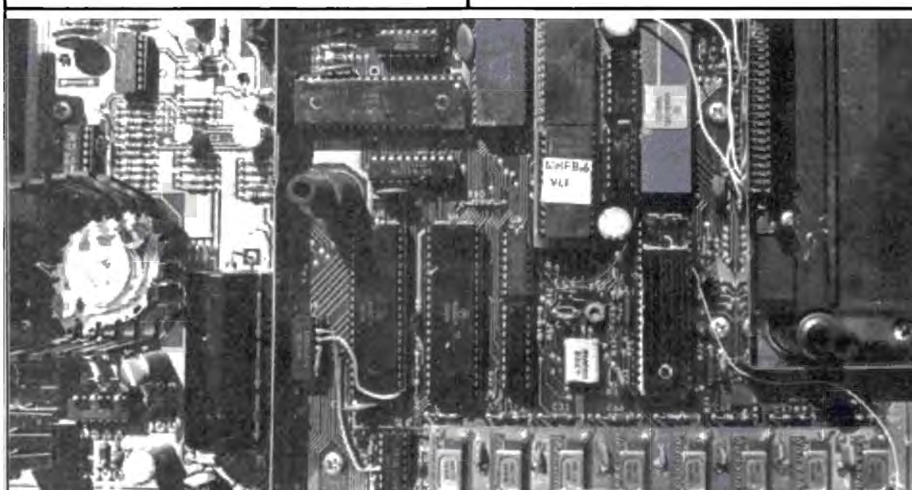
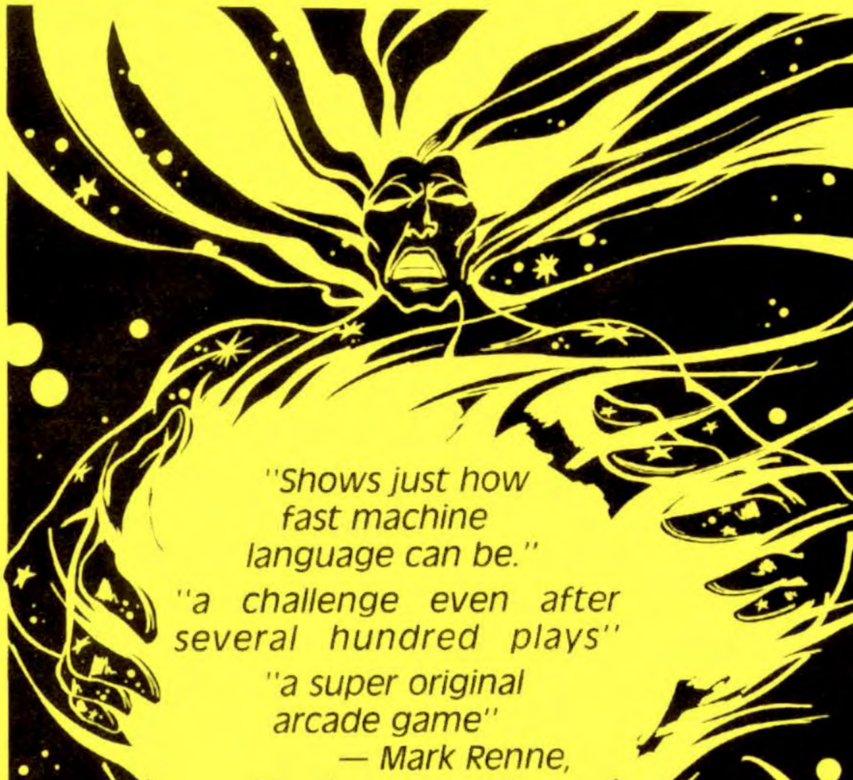


*Photo 1. Top view of the Color Computer. To the left is U29 and U11 with the jumper. Bottom center is the 64K RAM card with 4164 RAM. Top center is the Wolfbug adapter card.*

# ARCADE EXCELLENCE




Med Systems' staff of programmers is known for producing some of the finest software available today. Now we announce two arcade games worthy of the Med Systems name: Laser Defense and Star Trap. Laser Defense: a totally original action game, not just an arcade ripoff! And Star Trap: the first home arcade game to allow two players to compete **simultaneously**! Both feature state of the art graphics, sound, one or two player options, and high score saving.

But don't take our word for it. See the review in the August, 1982 issue of *'80 Microcomputing*. And remember, Med Systems offers to refund your money if you aren't satisfied. Just return the game within 14 days of receipt.

**Laser Defense**
Your laser satellites must defend helpless cities against nuclear attack while attacking and destroying the enemy's launch silos. Particle beam weapons, nuclear power stations, the eradicator, and a two-map graphics system combine to produce one of the finest arcade creations available.

**Star Trap**
The first **two player** arcade game!

Bouncing between two gigantic 3-D hypergrids are the stars of a thousand worlds. Capture as many as you can, or use your nova to annihilate them! **Simultaneous** two player option allows players to compete head-to-head in real time!

Laser Defense and Star Trap are available on cassette or diskette and require 16K of RAM.

| | | |
|---|---|---|
| **Laser Defense** | Cassette | $15.95 |
| | Diskette | $18.95 |
| **Star Trap** | Cassette | $15.95 |
| | Diskette | $18.95 |

*Please add $2.00 for first class postage, $4.00 for overseas air mail.*

## MED SYSTEMS SOFTWARE

P.O. BOX 3558 CHAPEL HILL, NC 27514
TO ORDER, CALL 1-800-334-5470

✓ 83

## OR SEE YOUR DEALER

to accept the new 4164 RAM. The uniqueness of this upgrade is its simplicity. Traces do not need to be cut; just solder in a couple of wires for power, address lines, and some jumpers, and you have 64K of usable RAM.

The instructions, which are very thorough, consist of two parts. One is for the EPROM adapter card, which holds Wolfbug in a 2716 EPROM, and the other is for the 64K RAM adapter card.

**Wolfbug Adapter Card**

The instructions for the Wolfbug card tell you to remove the Basic ROM and install the new card. Be careful; Radio Shack sockets are the type that prefer flat pins, like the ones on an integrated circuit, and Atomic City's are the round-type pin. I filed the sides of the pins slightly flat to solve the problem.

Now, carefully push the Wolfbug adapter card. Care must be taken here to avoid damaging the adapter card or breaking the pins. If there is any resistance, check for a mismatch. Four wires must be soldered; one is to a ground wire while the other three are for decoding Wolfbug.

**64K RAM Adapter Card**

Installing the 64K RAM adapter card is as simple as the Wolfbug card. Remove the old RAM, U20 through U27, from their sockets and install the 64K RAM card in their place. I experienced the same problem with the pins here as on the Wolfbug, so once again I filed the sides of the pins.

Another problem I encountered was that the RAM sockets, which Radio Shack installed, were misaligned, so care was needed when pushing the RAM card into place.

Two wires on the card require soldering: the wire on the left side of the RAM card to TP12 (+5 volt) and the right wire to V10, pin 35 (RASI). Two RAM size jumpers must be reconfigured, one by U4 to the 4K position, the other by U10 to the 16K position.

The Color Computer will run with these modifications now, but you cannot use the MAP type #1 mode yet. A jumper must be installed to get the MAP #1 (that is free memory from $0000 to $F800). For this part of the upgrade I recommend new U11 (74LS138) and U29 (74LS02) chips, because you can take out the MAP #1 mode, install the U11 and U29 chips and take your computer in for repair—after, of course, you take out the RAM card and Wolfbug.

On the new U11, I bent pin 5 straight out; on U29, pins 11, 12, and 13, also. Next, I inserted the U11 and U29 into their proper sockets and soldered a jumper from U29 pin 13 to U11 pin 5, U29 pin 8 to U29 pin 11, and from U29 pin 12 to U8 pin 21. This completed my 64K upgrade. After double checking my work, I turned the power on and tested to see if I had the correct voltage on pins 1, 8, and 16.

The big step came next. After turning off the power, I carefully installed the new 4164 RAM, making sure all the pins were in their sockets. After turning the power on I requested the memory size. I got back 24871, which is correct for the Extended Basic computer. For owners of Standard Basic, it will be 31015.

Next I tried Wolfbug, which has a memory-test routine, by holding the break key, pushing and releasing the reset button, and answering R to the Wolfbug prompt. This switches the system to all-RAM mode (MAP type



✓See List of Advertisers on Page 451

#1), 64K only. After being prompted a second time, I entered T for memory test. I next answered $0550 to the start-address query and $F7FF to the end-address query (which is the end of free memory). Immediately, characters appeared on the screen and began to move, indicating that the memory was being tested. The memory test takes about 15 minutes.

**What is Wolfbug?**

Wolfbug is a machine-language monitor that resides at $F800 to $FFEA. Its 12 single-key commands are as follows:

- A is an ASCII dump of any memory location. When A is entered, you are prompted with a start address. After entering the address, a memory page appears ($BF or 192 decimal memory address equals a page). An ASCII character will appear on the screen if a value at the required address equals an ASCII value. Hitting enter moves you to the next page.
- B enters Basic or returns you back to Basic.
- D routes the output to the display.
- F is the floppy disk boot.
- G is a go-to-a-machine-language program. After entering G, you are prompted with a Go-To query; enter your hex address and off you go.
- H is a hex dump of memory similar to A, except it gives you a hex value (a page here equals $5F or 96 decimal). Hitting enter advances you to the next page.
- L transfers Basic to RAM in the 64K mode and runs it.
- M examines or changes memory. When entering M, you are prompted with an examine-address query. Enter a hex address. Hit the space bar to advance and the back-space key to move back.
- P sends everything out to the printer that appears on the screen.
- T tests memory. When entering you are prompted with a start-address query and then an end-address query. The memory-test routine checks RAM by writing a pattern and then reading it. This pattern is altered 256 times.
- R switches you to MAP #1 mode, which is the full 64K mode.
- X allows typing text to the screen, like a typewriter. If the P command is entered before the X command, hit the enter key and it will send the screen information to the printer.

Wolfbug is an excellent upgrade from its installation to fine documentation. I have been using it for six months and have had no problems.■

---

★ ★ ½

*Elementary Basic: Learning to Program Your Computer in Basic with Sherlock Holmes*
*Elementary Pascal: Learning to Program Your Computer in Pascal with Sherlock Holmes*
**Chronicled by John H. Watson**
**Edited with Commentaries by Henry Ledgard and Andrew Singer**
**Random House Inc.**
**Vintage Books Division**
**Softcover, 265 pp. each**
**$12.95 each**

**by John P. Mello Jr.**
***80 Micro* staff**

As the preface to these books goes, Ledgard and Singer were taking a nip of tea one day when word came a trunk had arrived from England for Ledgard's wife Edwina. Neither man thought anything of the announcement until the next day when Singer noticed his friend was distraught.

"Henry, is something wrong? You don't seem yourself this morning," Singer said.

"It's the trunk," Ledgard replied.

"Trunk?"

"The trunk that came for Edwina from England yesterday."

"Oh, *that* trunk. What about it?"

"It's a very strange trunk; that is, I mean its contents are strange. The trunk is full of manuscripts."

"What's so strange about that?"

"They all seem to have been written by a certain Dr. John Watson about a certain Mr. Sherlock Holmes."

*"Watson opens each chapter describing an encounter with Holmes and his Analytical Engine."*

"Now really, Henry, be serious."

"I am completely serious, but I haven't told you the best part of it. What do you suppose the manuscripts are about?"

"I refuse to have my leg pulled in this fashion, Henry!"

"Andrew, Holmes was an ace programmer. According to Watson, he used a successor to Babbage's Analytical Engine in his work. The manuscripts are Watson's chronicles of their experiences. Holmes's insights are brilliant. His deductive powers and methodical nature made him a natural programmer."

Even if apocryphal, Ledgard's and Singer's premise is intriguing. It is also entertaining because the authors take pains to capture the flavor and proper Victorian prose of the Holmes stories.

Watson opens each chapter describing an encounter with Holmes and his "Analytical Engine." Holmes uses the magical machine to solve a murder at the Metropolitan Club, unlock the mystery of a socialite's disappearance, identify cigars by their ashes, find a rare Bible, break up a burglary ring, create a cipher for the Foreign Office, add molecular weights, write coroner's reports, uncover the handiwork of his arch foe Moriarty, and write a treatise on his methods. Each problem is analyzed by the master of Baker Street, displayed first as an "algorithm"—a shorthand form of the program—then as an actual program.

At the end of each chronicle, there is a didactic commentary by the authors detailing the concepts used by Holmes to solve a problem. As you might expect, the commentaries lack the stylistic charm of the chronicles, but are invaluable in giving a detailed explanation of Holmes's methods. They also serve as counterweights making the books easier to read. When Watson showed signs of information overload ("Enough, Holmes! I can absorb no more."), Holmes usually suggests a diversion, a night at the opera or a cold partridge and a bottle of Montrachet. The commentaries act like such a diversion for the reader.

One disappointing feature of the books is the chronicles in them are almost identical, so if you read one of the books, the second one's ruined. After all, without the chronicles, these are just books on programming. However, reading the books together can give you some insights into how Pascal stacks up against Basic.

Another problem a novice might encounter is getting the Basic programs to

*"... the commentaries lack the stylistic charm of the chronicles, but are invaluable in giving a detailed explanation of Holmes' methods."*

work. The authors spend a lot of ink explaining the idea behind "pretty-writing"—properly spacing the elements of a program to make it easy to read and understand—but ignore "user friendliness." For example, in the program for enciphering messages, you are told "enter message characters." I put in the entire message only to have the program bomb—only one letter could be enciphered at a time.

Can you start with zero knowledge about these programming languages and become proficient in them from reading these books? No. I had some background in Basic and none in Pascal when I started reading the books. I found my Basic knowledge necessary to handle the later chapters in the Basic book. After reading the Pascal book, I still did not feel proficient in the language.

So without prior knowledge of one of these languages, you could find these books—despite their clever approach to the subject—heavy sledding. Before you pick them up, you should have been puttering in the languages for awhile or taken a short course in the fundamentals, preferably the latter. There is no substitute for thinking when it comes to programming, Ledgard and Singer noted in their books. The same may be true of classroom teaching. ■



✓ See List of Advertisers on Page 451

★ ★ ★ ★

**Lisp**
**Supersoft**
**P.O. Box 1628**
**Champaign, IL 61820**
**$100 disk**
**$75 cassette**

★ ★ ★ ★

**UOLISP**
**Far West Systems and Software**
**P.O. Box 3301**
**Eugene, OR 97403**
**$109**

by **Bruce Powel Douglass**

Lisp stands for list processor. Lisp is the language of choice in artificial-intelligence programming, because of its flexibility, power, and ability to work with symbolic expressions. The two Lisps reviewed here are good implementations of the language; they do, however, take different approaches.

### Lisp

Supersoft's Lisp is not a "tiny" Lisp, but a complete version allowing full recursion. It has a PROG construct to allow using sequential types of programs written in Lisp. Lisp runs in 16K with cassette systems and supports single-precision, floating-point arithmetic and functions. It uses the Basic ROM routines to do this, which accounts for considerable reduction in the size of the Lisp interpreter.

The manual is fairly good. It is not a tutorial on Lisp, but does describe the Supersoft implementation well. It lacks an index, but does have a table of contents and a list of relevant references. It assumes little programming expertise on the part of the user other than a little familiarity with Lisp.

The language itself is easy to use. A Trace debugging facility is provided so you can trace a function's execution. Also, a symbolic-differentiation program is supplied, as well as a Taylor-series expansion program. These allow easy entry of mathematical expressions and generate the derivatives or Taylor series, respectively.

The package provides a few abilities beyond normal Lisp packages. The Basic transcendental functions (such as SIN, COS, ATN, and LOG) are supported. Explode and Implode, functions not supported by UOLISP, are supported. PEEK and POKE are also available in Supersoft Lisp, as is LPRINT. TRS-80 graphics commands are not supported.

Disk I/O is provided so that you can load or save functions and programs to or from disk. The files are saved in ASCII format, so you can edit them with a text editor. Generally, for long programs, it is easier to create and edit them with a text editor rather than using the Lisp environment. You can save property lists of atoms as well as functions and atoms. Tape users use CLOAD and CSAVE, and disk users use Load and Save; disk users can also kill disk files from within Lisp.

Edit, a Lisp editor, lets you modify existing Lisp functions. To use it, load it from disk (with the command (LOAD '(EDIT/LSP)) ) or from tape ((CLOAD) ). You enter the editor with the command (EDIT FN), where FN is some existing Lisp function. You can add to the editor by using it to edit itself. Some of the abilities of the editor include moving across or down (into) a list, going to the top of the list, inserting text, replacing text, setting breakpoints, printing the entire function, and deleting an item. For simple editing, Edit works fine, but for extensive editing you should use a full-screen editor. Incidentally, you can use The Alternative Source's KBE for this purpose and have a full-screen editor within the Lisp environment.

Supersoft Lisp is an excellent Lisp for beginners and advanced programmers. It is easy to use, and many TRS-80 functions and floating-point routines are supported, adding to its flexibility.

### UOLISP

UOLISP is not quite as complete an implementation as Supersoft's Lisp, but it has other attractions. UOLISP runs in a minimum of 16K, although 32K or 48K is recommended. The extras that come in the UOLISP package include a Lisp interpreter, a Lisp compiler, the Meta language writing system, an RLisp compiler, a Trace function, and a small Lisp editor. UOLISP does not support floating-point numbers.

The manual is not as easy to read as Supersoft's manual. If you are not an

See List of Advertisers on Page 451

experienced Lisp user, it will be very difficult to get through. It does sport an index, but lacks a table of contents.

UOLISP supports a large subset of Lisp functions. It does not, however, support Explode, Implode, or Expand. TRS-80 graphics are supported.

Disk I/O supports the standard ASCII Lisp source-code format file, similar to that used by Supersoft and a Fast Load format since it contains a compiler in addition to the interpreter. This kind of file loads and saves more quickly, it takes up less space in memory and on disk, and executes significantly faster.

The compiler requires UOLISP to be resident to run the compiled programs and is easy to use.

LAP stands for Lisp Assembly program, and is a Z80 assembler resident with the interpreter. It allows you to enter Z80 op-codes (which are slightly different for this assembler than for the standard assembler since TDL mnemonics are used), and these codes will be assembled into a function. This is useful for creating functions that must run quickly.

One of the nice features about UOLISP is that it contains an RLISP parser. RLISP is a cortex syntax for Lisp, and was created by A. C. Hearn in 1973. Lots of people are intimidated by the strange appearance of Lisp programs, and RLISP provides Lisp with a more understandable syntax. It contains a parser that reads in RLISP syntax and parses the expressions into Lisp format, and then passes it over to Lisp to evaluate. It is easy to move back and forth between RLISP and Lisp.

*"Supersoft's Lisp is not a 'tiny' Lisp, but a complete version allowing full recursion."*

RLISP is a structured language with If...Then, Do...While, and For constructs. The For construct lets you perform actions for each element in a list, as well as the standard FOR i := 1:2 DO. The syntax is FOR EACH i IN X DO..., where X is the list. RLISP functions can be compiled by the compiler as they are read in. The manual's chapter on RLISP is more readable than most, and beginners should be able to get through it with a minimum of work. RLISP is a useful addition to this package.

Another addition to UOLISP is the Little Meta Translator Writing System, or Meta. It also sits on top of the Lisp interpreter. The intended use for Meta in the UOLISP package is for language-design experimentation. Meta supports BNF-like syntax, recursive-descent parsing schemes, lexical primitives, pattern-directed code generation, and syntax-error message generation. Examples are given, and a calculator-like language is implemented before your very eyes. If you've ever wanted a simple way to translate Fortran into Basic or Pascal into PL/C, this is the proper environment. A separate manual accompanies Meta. Lisp is the ideal language environment for writing such a translator, and UOLISP is greatly enhanced by including it.

UOLISP lacks floating point and a few Lisp functions, but is fairly complete and easy to use, once you figure out how. It also features a function compiler, the LAP assembler, RLISP, and the Meta Translator system. If you have experience in Lisp and like to experiment, UOLISP is a great choice. ■

---

★★★★

**Quest**
**Aardvark-80**
**2352 S. Commerce**
**Walled Lake, MI 48088**
**Color Computer, 16K**
**$14.95 cassette**

**by Beth Ann Norman**

Quest is an adventure game written in Basic. Your goal is to travel through the lands of Alesia to Moorlock's citadel and, ultimately, defeat the forces of evil within. What makes Quest unique is that it's different every time you play it. It also makes good use of the Color Computer's graphics.

The game starts with the computer drawing a randomized map of Alesia. Certain features of the map are always the same; a river always runs straight across the middle of the screen, and Moorlock's citadel is always in the lower right corner. Also, two cities, one on each side of the river, a mountain range, and a swamp are always present, although randomly placed. Your character is represented by a colon and is placed randomly at the beginning of each game somewhere in the upper half of the screen.

The bottom of the screen is the command line. Your major commands, with which you answer the "What now, Sire?" prompt, are: N, E, W, and S for the directions, M for redrawing the map, I for taking inventory of your possessions, and U for when you wish to use an item. There are other commands for special situations. For example, you might meet a band of well-armed ogres. The computer asks you if you want to attack, run, or be friendly. You reply by giving the first letter of your choice.

You start each adventure with 10 loyal companions. Your object is to gather the men, weapons, armor, and food capable of defeating the forces at the citadel. You gain men and money by fighting or coming to terms with bands of ogres, wizards, pilgrims, and soldiers. You can also find money in caves, ruins, castles, and towers. All are randomly placed with each new game, and do not appear on the screen until you approach them while exploring Alesia. You can also buy food, armor, and weapons at the marketplace in each city.

You will need certain items to gain entrance to many places. For instance, when you find a cave, the computer draws a picture of the entrance and tells you, "You're at a cave. It's too dark to see! What now, Sire?" Your response might be to type in U for a listing of everything your men are carrying to see if you've brought a light source along. If you have, use it to enter the cave. You then are given the choice of searching or leaving.

I'll give you a hint: It you don't find anything at first, keep searching—you *will* find something eventually, even if it's a band of soldiers. If you re-enter a building after having removed the goods, though, you won't find anything new.

When you buy needed items in the

# ACEMAIL

**The Evolution of Low-Cost Electronic Communications**

We believe AceMail is probably the most powerful, automated two-way message system available for anywhere near the price. But don't let the price fool you.

The Automated Computer Electronic Mail software offers some unprecedented clout in communications across the phone lines. From a completely unattended TRS-80, AceMail can look up a phone number, dial it, log/on to the system, upload and download ASCII files, log/off and wait until later to call another computer. You may also call AceMail from a remote location and exchange information.

A world of information is at your fingertips as AceMail offers speed and accountability in accessing branch office correspondence, sales reports, data banks and time sharing systems nationwide.

The multifunctional system is easy to use and comes with a complete users manual allowing even the inexperienced user to send and receive electronic mail in minutes!

If your TRS-80 is due to evolve in the world of electronic mail, then choose the communications system with ease and power—AceMail.

Written exclusively for the Hayes Stack Smartmodems and the TRS-80 Model I/III 48K with disk. AceMail operates on NEWDOS/80 or DOSPLUS and comes supplied on 'tiny' DOSPLUS ready to run.

Hayes Stack Smartmodem 1200—$619.00
Hayes Stack Smartmodem 300—$239.00
ACEMAIL 1200 Software—$119.00
ACEMAIL 300 Software—$79.00

**ace computer products of florida, inc.**

**1640 n.w. 3rd street, deerfield beach, florida 33441 voice (305) 427-1257 data (305) 427-6300**

**ORDER TOLL FREE 1-800-327-2283**

~197

marketplaces, you find that the list of what is on sale is randomized and changes with each visit. Some items have many uses, and there's one that I still haven't figured out. One of the funniest parts of the game is haggling with the merchants.

A realistic feature of the game is the weight system used. Each man can carry only 100 pounds. The computer keeps track of the total weight of your items (plate mail—50 pounds, broadsword—10 pounds, and so on) and compares it to the amount your men can carry. If the total is too high, you must drop items until the weight is acceptable. You cannot ignore these warnings, since you will be unable to move until enough material has been left behind.

When you think you've found or bought everything you can in the upper half of the map, you're ready to cross the river. Beware traveler; your men can't swim!

Once you have crossed the river and you've gathered what you consider to be enough men, weapons, armor, and food, attack the citadel. You will find it difficult to defeat Moorlock's forces unless you have a certain item, which of course I cannot reveal. The display for a fight (either at the citadel or with anyone you encounter) lists everchanging tallies of your men, weapons, experience level, luck, and survivors, against theirs. To stop the fight, hit the space bar when you think your luck number is higher than your opponent's, although this is just one factor in deciding the outcome. You can next see how many survivors each party has. You can fight, run, or come to terms whenever a fight has been stopped. If you run, your foes may catch you, and just because you want to come to terms doesn't mean that they will.

*"Once you have crossed the river and you've gathered what you consider to be enough men, weapons, armor, and food, attack the citadel."*

If you kill all your enemies in a fight outside the citadel, salvage their equipment. If you come to terms, some will join you, bringing equipment, food, and money. If all your men are killed, of course, the game ends.

Quest is a wonderful, well-thought-out game. It is a terrific game for beginning adventurers, because there are no sudden unreasonable ways to lose. For example, if you are losing men in a fight, you can always try running away. You can save a game on tape at any point (as long as the "What now, Sire?" prompt is on the screen), and return to it later. Quest is also fairly short for an adventure—playable in two to five hours. Since the game is different every time, you are not likely to lose interest in it the first time you defeat the forces of evil. ■

★ ★ ★ ★ ½

**LSI Soft-view CRT**
**Langley-St. Clair Instrumentation Systems Inc.**
**132 West 24th St.**
**New York, NY 10011**
**$80–$100**

**by Dave Smith**

Remember the night you worked into the wee hours coding and debugging that super adventure on your TRS-80? Or the time you had to turn out that 30-page, end-of-the-year report with your word processor? Remember how tired your eyes became?

The reason for such fatigue stems from the cathode ray tube (CRT) Tandy uses in the manufacture of the TRS-80. It is basically a tv-type picture tube on which the display strobes 60 times per second. The screen is coated with rapid-decay white phosphors that fade quickly and are re-excited upon the next strobe, causing a repeated fading and brightening. The LSI Soft-view CRT combats this problem.

These replacement CRTs are available with either a slow-decay green phosphor or a medium-decay amber (orange) phosphor. The medium-decay phosphors fade at a rate nearly 100 times slower than the white phosphors. For this review I used the amber display CRT.

The CRT comes with an eight-page installation instruction booklet. All the appropriate warnings regarding health, safety, and the invalidation of the Tandy warranty are prominently displayed in the booklet. There is also a handy list of the tools needed if you plan to do your own installation. (The alternative is to rely on your local computer center or tv repair shop for the installation.)

The instructions include fully detailed, step-by-step procedures, locations and types of screws to be encountered, colors of wires to be unplugged, and so on. They do not include any pictures or diagrams, but are very easy to follow just the same.

The hardest part of the installation is accepting the recommended one-hour wait following turning the power on before making further internal adjustments. (My installation required only one such adjustment.)

Finally, the moment of truth. It gave me a real lift to stand back and look at the familiar grey case with the totally unfamiliar amber letters and numbers on the screen. It took some time for my audiences' ooohing and aahing to subside, but I managed eventually to get down to some dispassionate assessment of performance.

The amber phosphors do not glow with the same intensity as the white. (I was later informed that they are only 42 percent as efficient in their light-emitting properties.) Therefore, I spent some time adjusting and readjusting the brightness and contrast controls for acceptable character clarity. In addition, the new CRT features both the double-dark leaded glass (standard) and the anti-glare surface (optional), further reducing the brightness of the screen. The



✓See List of Advertisers on Page 451

benefit of the leaded glass is the reduction of x-ray emissions from the screen.

My initial reaction was very favorable, but included a twinge of disappointment, since I had been expecting an amber duplication of the white phosphor screen. It was only after using the computer for an extended time that I fully appreciated the pleasure of fatigue-free viewing. I became accustomed to the subdued amber screen very quickly, tending to be unconscious of the color and of the absence of flicker and glare until forced to use a standard machine. Having used the amber CRT, I found the black-and-white screen harsh and discomforting.

*"I became accustomed to the subdued amber screen very quickly."*

If you use your TRS-80 frequently or for extended periods at a single sitting, the improved viewing comfort and health features of these replacement CRTs are very much worth the price. ■

★★★★

**FED File Editor for LDOS**
**Galactic Software Inc.**
**11520 N. Port Washington**
**Mequon, WI 53092**
**$40 disk**

**by Charles P. Knight**

FED is a file editor designed to run only under the LDOS operating system. It is a unique concept in zap programs in that it operates on the file level only, rather than the track/sector level. It is a great improvement over the usual zap program.

File level only does not mean that files like DIR/SYS and BOOT/SYS cannot be examined; they must be examined as a file rather than as a group of sectors. BOOT/SYS, for example, is a file of five sectors beginning at track 0, sector 0. If it is accessed as a file, it isn't necessary to know where it is on the disk. The password must be supplied. If you don't know it, RS0LT0FF gives you access to any file anywhere under LDOS. Because of this approach, many advanced features are available and you can perform functions that would be difficult if you had to know which sectors on the disk contained the file.

There are two display modes: One is a standard 256-byte mode with the ASCII display on the left instead of the usual right, and the other is an extended 128-byte mode with the ASCII across the top of the screen, the hex in the center, and plenty of space for other information (where the cursor is, name of file being worked on, hex and decimal value of byte under cursor, and so on).

Pressing the T key toggles between these two display modes. The latter is particularly nice for modifying ASCII files. Also available on the video in the 128-byte display mode are file name, relative byte under cursor, decimal and hex values of that byte, drive number, record number within the file, and the last command entered.

Graphics are not displayed and are represented by a graphic block instead of the usual period—a great help when dealing with files that contain both periods and graphics. I think a major improvement would be an optional graphics ASCII display, though. You can modify graphics by entering the appropriate hex character, but you can't view your changes directly by looking in the ASCII column.

*"The usual file modifications are available plus much more. You can modify in hexidecimal or ASCII."*

The usual file modifications are available plus much more. You can modify in hexadecimal or ASCII, or if you want to know what the character under the cursor is in binary, press the equal key (=) and it is immediately displayed at the lower right of the screen. The binary representation should display the value under the cursor all the time in the 128-byte mode, as there is enough room on the screen; however, this is not a serious drawback.

If you want to know where a byte will load into memory, press M and the load address of that byte will be displayed. This is useful particularly when converting X-type patches to D type. If you want to save the modified file, you must press the break key to exit the modify mode, and then press S followed by enter. By requiring that enter be pressed here, mistakes are more difficult to make and much easier to correct.

There are three commands for searching through the file (the whole file, not just the displayed sector like most zap utilities). The C command searches through the file for an ASCII string up to 15 bytes long in the 256-byte display mode or 30 bytes long in the 128-byte display mode. The search begins at the present cursor position and searches the rest of the file. Unlike many such features in other utilities, if the string searched for falls across sector boundaries, FED still finds it correctly.

The F command performs the above search for a string of hex bytes up to six characters long if in the 256-byte display mode, or up to 30 characters if in the 128-byte mode.

The L command, unlike any I've seen before, searches for a load address, and if found, positions the cursor over the byte that loads at that address. This search starts at the beginning of the file and continues until the end is reached.

The G command can be used after any one of the above three commands to go to the next occurrence of that byte, string, or load address in the file.

The D command prints the entire file to the line printer. If the printer is offline, FED waits until it is ready or until the break key is pressed. The format prints three sectors of information on a page. FED performs its own pagination, so if you have a printer filter resident for output formatting, you might want to disable it to prevent conflict. The output is well formatted and thoughtfully arranged.

The P command sends the current edit buffer to the printer rather than the entire file, while the O command outputs a top-of-form character (CHR$(12)) to the printer. This is a clear example of programming that keeps the user in mind. Another example of this is the Enter command. Simply pressing the enter key displays a menu of available commands and their functions.

For machine-language programmers, the Z command jumps from load block to load block within the file. The cursor must be on the type byte (usually 01) when the key is pressed. FED then finds and positions the cursor over the type byte of the next load module. This is very handy for working out patches to someone else's code when you don't have the source.

When you are done editing the file, you can press E and enter to exit FED, or N, enter, and the file name of another file to be worked on. During the entering of any command, if you change your mind or make a mistake, simply press the break key to abort the command sequence. To position the cursor to the beginning of the file or to its end, press the B or E key. To position the cursor at the beginning of a particular logical record (sector when LRL = 256), press the R key followed by the number of the logical record you want in hexadecimal. While editing, the arrow keys move the cursor and the semicolon and hyphen keys move the display forward or back one sector.

For $40 you receive a disk-file editor that has capabilities you'll wonder how you got along without. I use mine almost daily and keep a copy on my working system disk so that it is always available to me. If you're an LDOS user and have a need to modify files at the disk-sector level, you should have this program in your library.■

---

★ ★ ★ ½

**Bounceoids**
**Cornsoft Group Inc.**
**6008 N. Keystone Ave.**
**Indianapolis, IN 46220**
**Model I or III**
**$15.95 cassette**
**$19.95 disk**

**by Mark E. Renne**

The scenario for Bounceoids is a familiar one. You move your ship, located in the middle of the screen, by using a joystick or keyboard control. Your mission is to destroy the large, floating space boulders surrounding your ship before they destroy you. This game, however, is not another version of asteroids.

You can maneuver your ship in a variety of ways. The arrow keys, R, T, O, P, or a joystick can be used to turn the ship right or left, move it forward, or activate the shields. The shields protect the ship in the event a boulder gets too close.

Unlike an asteroid-type game, you cannot penetrate the boundaries of the screen. Boulders bounce off the walls of the screen and ricochet at 90-degree angles. Boulders form groups of four or more and move in unison. The groups can become very large at higher skill levels and form a straight line that crosses the entire screen.

If these groups of boulders aren't enough to throw you, a snake-like creature periodically appears from the side of the screen to confront you. In the higher levels, this multi-segmented creature becomes so long it is almost impossible to destroy.

If this still isn't enough trouble for the advanced alien fighter, there's more action ahead. At points directly north, south, east, and west, alien artillery posts appear. When playing in the lower skill levels they will wait a long time before firing at your ship, but as you move up in skill, they fire very rapidly. Your epitaph will be written quickly if you're not fast on the fire button.

Just for fun, there's also a wandering alien who bounces around causing more problems for would-be starfleet heroes. With all these different bad guys running around, it's not easy to survive. Should you accumulate 10,000 points a new ship is awarded.

After 20,000 points you reach the real challenge of this arcade madness. A string of unique aliens descend from the upper-left corner and move across the screen, allowing you to gather extra bonus points. At this point you also change skill levels.

Only the highest score is maintained on the system (and only until you turn off the power). You don't even get a chance to enter your initials. It does seem strange on a disk system that the top 10 scores aren't maintained.

Ten levels of play are available for one or two players. Each player may start at a different level of play. For example, player one may play at level 9 (expert) while player two plays at level 0 (beginner).

This game is fast and should provide the ardent arcader with enough action to provide a challenge and sore fingertips.■

---

★ ★ ★ ★

**The Disk Doctor**
**Superior Graphic Software**
**406 Little Mountain Road**
**Waynesville, NC 28786**
**Color Computer**
**$49.95**

**by John Steiner**

The Disk Doctor is a Basic program that assists in the reconstruction of a damaged Color Computer disk file.

To ensure the integrity of your files, the Disk Doctor manual starts off with this advice: Check files for accuracy by using the Verify On command before saving a file or program. Use Field commands from Basic, rather than from within a program. Keep your fingers, coffee, nail files, and so on away from the head opening on your disks. Don't put disks on top of your monitor or drive unit. The magnetic fields generated by these devices make quick work of your files. Back up anything you think is important, and do it often.

Another hint given by the Disk Doctor is to put a write-protect tab on any disk any time you receive an I/O error. Write-protecting a disk keeps the allocation table from becoming garbled and assists you in the reconstruction of the disk. The disk files that are still intact can be transferred to another preformatted disk using the Copy command.

Regardless of what the Color Computer disk manual says, Copy *will* work on a single-drive system; just enter COPY "FILENAME/EXT". You are prompted when to change disks, just as

# Telewriter-64™

## the Color Computer Word Processor

- **3 display formats: 51/64/85 columns × 24 lines**
- **True lower case characters**
- **User-friendly full-screen editor**
- **Right justification**
- **Easy hyphenation**
- **Drives any printer**
- **Embedded format and control codes**
- **Runs in 16K, 32K, or 64K**
- **Menu-driven disk and cassette I/O**
- **No hardware modifications required**

### THE ORIGINAL

Simply stated, Telewriter is the most powerful word processor you can buy for the TRS-80 Color Computer. The original Telewriter has received rave reviews in every major Color Computer and TRS-80 magazine, as well as enthusiastic praise from thousands of satisfied owners. And rightly so.

The standard Color Computer display of 32 characters by 16 lines without lower case is simply inadequate for serious word processing. The checkerboard letters and tiny lines give you no feel for how your writing looks or reads. Telewriter gives the Color Computer a 51 column by 24 line screen display with *true lower case characters*. So a Telewriter screen looks like a printed page, with a good chunk of text on screen at one time. In fact, more on screen text than you'd get with Apple II, Atari, TI, Vic or TRS-80 Model III.

On top of that, the sophisticated Telewriter full-screen editor is so simple to use, it makes writing fun. With single-letter mnemonic commands, and menu-driven I/O and formatting, Telewriter surpasses all others for user friendliness and pure power.

Telewriter's chain printing feature means that the size of your text is never limited by the amount of memory you have, and Telewriter's advanced cassette handler gives you a powerful word processor without the major additional cost of a disk.

*...one of the best programs for the Color Computer I have seen...*
— Color Computer News, Jan. 1982

### TELEWRITER-64

But now we've added more power to Telewriter. Not just bells and whistles, but major features that give you total control over your writing. We call this new supercharged version Telewriter-64. For two reasons.

### 64K COMPATIBLE

Telewriter-64 runs fully in any Color Computer — 16K, 32K, or 64K, with or without Extended Basic, with disk or cassette or both. It automatically configures itself to take optimum advantage of all available memory. That means that when you upgrade your memory, the Telewriter-64 text buffer grows accordingly. In a 64K cassette based system, for example, you get about 40K of memory to store text. So you don't need disk or FLEX to put all your 64K to work immediately.

### 64 COLUMNS (AND 85!)

Besides the original 51 column screen, Telewriter-64 now gives you 2 additional high-density displays: 64 × 24 and 85 × 24!! Both high density modes provide all the standard Telewriter editing capabilities, and you can switch instantly to any of the 3 formats with a single control key command.

The 51 × 24 display is clear and crisp on the screen. The two high density modes are more crowded and less easily readable, but they are perfect for showing you the exact layout of your printed page, *all on the screen at one time*. Compare this with cumbersome "windows" that show you only fragments at a time and don't even allow editing.

### RIGHT JUSTIFICATION & HYPHENATION

One outstanding advantage of the full-width screen display is that you can now set the screen width to match the width of your printed page, so that "what you see is what you get." This makes exact alignment of columns possible and it makes hyphenation simple.

Since short lines are the reason for the large spaces often found in standard right justified text, and since hyphenation is the most effective way to eliminate short lines, Telewriter-64 can now promise you some of the best looking right justification you can get on the Color Computer.

### FEATURES & SPECIFICATIONS:

**Printing and formatting:** Drives any printer (LPVII/VIII, DMP-100/200, Epson, Okidata, Centronics, NEC, C. Itoh, Smith-Corona, Terminet, etc).

Embedded control codes give full dynamic access to intelligent printer features like: underlining, subscript, superscript, variable font and type size, dot-graphics, etc.

Dynamic (embedded) format controls for: top, bottom, and left margins; line length, lines per page, line spacing, new page, change page numbering, conditional new page, enable/disable justification.

Menu-driven control of these parameters, as well as: pause at page bottom, page numbering, baud rate (so you can run your printer at top speed), and Epson font. "Typewriter" feature sends typed lines directly to your printer, and Direct mode sends control codes right from the keyboard. Special Epson driver simplifies use with MX-80.

Supports single and multi-line headers and automatic centering. Print or save all or any section of the text buffer. Chain print any number of files from cassette or disk.

**File and I/O Features:** ASCII format files — create and edit BASIC, Assembly, Pascal, and C programs, Smart Terminal files (for uploading or downloading), even text files from other word processors. Compatible with spelling checkers (like Spell 'n Fix).

Cassette verify command for sure saves. Cassette auto-retry means you type a load command only once no matter where you are in the tape.

Read in, save, partial save, and append files with disk and/or cassette. For disk: print directory with free space to screen or printer, kill and rename files, set default drive. Easily customized to the number of drives in the system.

**Editing features:** Fast, full-screen editor with wordwrap, block copy, block move, block delete, line delete, global search and replace (or delete), wild card search, fast auto-repeat cursor, fast scrolling, cursor up, down, right, left, begin line, end line, top of text, bottom of text; page forward, page backward, align text, tabs, choice of buff or green background, complete error protection, line counter, word counter, space left, current file name, default drive in effect, set line length on screen.

Insert or delete text anywhere on the screen without changing "modes." This fast "free-form" editor provides maximum ease of use. Everything you do appears immediately on the screen in front of you. Commands require only a single key or a single key plus CLEAR.

*...truly a state of the art word processor... outstanding in every respect.*
— The RAINBOW, Jan. 1982

### PROFESSIONAL WORD PROCESSING

You can no longer afford to be without the power and efficiency word processing brings to everything you write. The TRS-80 Color Computer is the lowest priced micro with the capability for serious word processing. And only Telewriter-64 fully unleashes that capability.

Telewriter-64 costs $49.95 on cassette, $59.95 on disk, and comes complete with over 70 pages of well-written documentation. (The step-by-step tutorial will have your writing with Telewriter-64 in a matter of minutes.)

To order, send check or money order to:

Cognitec
704 Nob Ave.
Del Mar, CA 92014 ✓121

Or check your local software store. If you have questions, or would like to order by Visa or Mastercard, call us at (714) 755-1258 (weekdays, 8AM-4PM PST). Dealer inquiries invited.

(Add $2 for shipping. Californians add 6% state tax. Allow 2 weeks for personal checks. Send self-addressed stamped envelope for Telewriter reviews from CCN, RAINBOW, 80-Micro, 80-U.S. Telewriter owners: send SASE or call for information on upgrading to Telewriter-64. Telewriter compatible spelling checker (Spell 'n Fix) and Smart Terminal program (Colorcom/E) also available. Call or write for more information.)

in the Backup command. Any Basic, ASCII, or binary file can be copied. Unlike Backup, programs in memory are not affected. You can use Copy from within a program, but you may have to switch disks often if the program is in memory or the file is large.

If you used the write-protect tab when you first received an I/O error, you only need to reconstruct that file containing the error. The Disk Doctor reads the allocation table and displays the locations of the files on the disk. A single file is easy to locate from the table. Just make a copy of the table that contains file-continuation information. If you have a printer, the documentation tells you how to make the program print the table or any program output on your printer.

Once you have this information, you can check the contents of any sector on the disk. The Disk Doctor transfers the file to graphics memory one sector at a time. When the file transfer is complete, install a formatted disk, and the menu-driven program transfers the file from graphics memory into a new disk file.

Reconstructing an entire disk involves a little more work, especially if the allocation table has been destroyed. The Disk Doctor lets you reconstruct any file type including machine-language programs.

The Doctor also contains a feature useful for good disks. If you locate the beginning of a binary file and read the first sector, you are given the start, end, and execute addresses. I recommend using the program on good binary files so you can find and store this information should you ever need to reconstruct.

*"The author has included a crashed disk with your program disk. If you... reconstruct the crashed disk, you get a back-up of the Disk Doctor..."*

Disk Doctor's written documentation is minimal, but the eight pages are packed with information on reconstructing a crashed disk and preventative care.

The author has included a crashed disk with your program disk. If you successfully reconstruct the crashed disk, you get a back-up of the Disk Doctor and a Packer program. Packer, a binary routine, removes excess spaces from any Basic program to make the program run faster. On the crashed disk, data and graphics files are included to demonstrate how they can be reconstructed. It takes very little time to learn the reconstruction process, if you do the assigned homework. The homework is an overview of the disk-allocation information in the Color Computer's disk operating manual.

I found one bug when I tried to transfer a large file. If the file you are transferring is larger than the eight graphic pages allocated (about 12K), the file must be recreated in parts. If you forget where you are and continue the transfer past the eighth page, you will overwrite the beginning of the Doctor and the program crashes. The only recovery is to reload the program and begin again. You will not encounter this bug very often (few files are longer than 12K) and a little care on your part easily avoids the problem.

The Disk Doctor teaches you disk anatomy as well as preventative medical information, allowing your disk files to lead long and fulfilling lives. I feel safer when creating important files and programs having the Doctor "on call."■

---

★★★★★

**LCA-47 Lowercase Adapter**
**Micro Technical Products Inc.**
**814 W. Keating Ave.**
**Mesa, AZ 85202**
**$75**
**$25 additional character generator ROMs**

**by Scott L. Norman**

The Color Computer is maturing; we are beginning to see hardware add-ons similar to those introduced for the Model I. One such useful device that improves the appearance of text on the video screen is the LCA-47 Lowercase Adapter.

The LCA-47 is a small printed circuit board that plugs into the socket normally occupied by the Color Computer's MC6847 video display generator IC; the IC, in turn, plugs into the LCA-47.

In an unmodified Color Computer, the 6847 uses its own circuitry to form and synchronize the video pulses that produce characters on the display. Lowercase letters are not included in the repertoire of this internal circuitry. When the LCA-47 is in place, the 6847 has access to an externally defined character set—in particular, one that is contained in a fast bipolar ROM. This set is then displayed by all programs that normally call upon the 6847 to perform its duties—the LCA-47 is invisible to such programs. However, Telewriter, Colorterm, and the like (packages that use the CC's high-resolution graphic modes to define new character sets) are unaffected, since they effectively bypass the 6847 anyway.

Installation of the LCA-47 requires that the Color Computer case and RF shield be opened, though, so all the usual precautions about the 90-day warranty apply.

There are only a couple of other things that should be mentioned. One is the possibility that your Color Computer will have its 6847 soldered into the motherboard, rather than socketed. (This was the case with some of the early machines.) Recognizing the difficulties inherent in unsoldering a 40-pin IC, Micro Technical Products offers a refund to purchasers who meet this particular stumbling block; they also describe the procedure for sacrificing the IC and piggybacking the LCA-47 on the remains of the connecting pins. You must buy a new 6847 to make the computer operational, of course.

The only other problem likely to arise is mechanical interference with a particular electrolytic capacitor on the Revision E motherboards. There are actually two versions of the LCA-47 available; one has a clearance hole for this component.

Open the computer's case and RF shield before ordering. If the 6847 is soldered in place, you may want to rethink the whole affair; if not, check for a small tubular electrolytic mounted vertically behind the 6847. If it's there,

# Words are not enough.


Experience goes beyond words. We believe adventures should too. The Asylum Series is Med Systems' premiere line of 3-D graphics adventures for the TRS-80 and IBM personal computers. What is an Asylum? An Asylum is a real-time simulation that takes place in a building with 1500 locations. Both are inhabited by crazed inmates, sadistic guards, and evil doctors. Your goal? ESCAPE!

Our 3-D graphics are recognized throughout the industry for their machine-language speed and clarity. Hallways recede into the screen as though you are actually there. Doors open and close. Beds, desks, and other inmates are drawn on the screen. Instantaneously!

Asylums understand **complete sentences**, not just choppy one and two word commands. Sentences like "Drop everything on the desk except the matches" are interpreted and acted on **instantly**. Further, the command "VOCABULARY" will show you the entire dictionary, eliminating the need to second-guess program designers you have never met.

But don't take our word for it. See the reviews in *'80 Microcomputing* (Feb. 1981, Aug. 1981, May 1982). And remember, Med Systems offers to refund your money if you aren't satisfied. Just return the game within 14 days of receipt.

Although Asylum II is the sequel to Asylum I, you need not have played one to play the other. Asylum I and Asylum II require at least 16K of RAM, and are available on tape or disk (please specify). Hint sheets are available for each Asylum for $1.00.

**Asylum I**
**(cassette or diskette) . . $19.95**
**Asylum II**
**(cassette or diskette) . . $19.95**
**Asylum I & Asylum II**
**(one diskette) . . . . . . . . $34.95**

**MED SYSTEMS SOFTWARE** 83
P.O. BOX 3558 CHAPEL HILL, NC 27514
TO ORDER, CALL 1-800-334-5470

**Or see your dealer.**

advise the vendor when ordering so he can supply you with the right board.

The LCA-47's manual is excellent, and gives detailed instructions for removing the 6847 and installing the plug-in board. The procedure isn't very difficult, but you should be cautious. Work slowly and carefully. When the board and IC are installed, you may find that one corner of the RF shield can no longer be pressed down as far as the others. I have never observed any increase in RF interference due to the misfit.

The LCA-47 board contains a pair of DIP switches, one to enable or disable the whole unit and another to reverse the foreground and background colors. The normal operation mode gives dark characters on a light background, which is very readable. If you prefer the reverse, just flip the switch. It is inconvenient to open the computer's case to operate the switches, so instructions are included for connecting outboard switches for remote operation. Once the unit is set up and enabled, the Color Computer's keyboard works in the usual fashion.

The LCA-47's output looks very fine indeed. Someone at Micro Technical Products has paid a lot of attention to the typographical aspects of the video display. You might expect that the lowercase characters would be good looking with real descenders, and so they are.

What comes as a pleasant surprise is the uppercase character set. A number of subtle and not-so-subtle changes have been made, with the result being improved readability over that of the Color Basic characters. For example, the letter O is now oval instead of rectangular; the numeral zero has a slash through it to avoid confusion. Uppercase letters S, E, and F, among others, have been improved, and even some of the numerals and other symbols have been touched up. My elderly tv receiver now has a much easier time resolving the asterisk; it no longer resembles a fuzzy plus sign.

### Word Processing With the LCA-47

These changes improve the readability of both source listings and text output of Basic programs. However, the real value of the LCA-47 lies in word processing. Users of Color Scripsit, C.C. Writer, Super Color Writer, and other color word-processing packages can now enjoy video output that looks like the printed page. This eases the chores of proofreading and editing.

The LCA-47 only solves the problem of character shape; line length and number of lines per screen are unchanged from whatever is standard for a given program. The appearance of printed characters is unchanged as well, although you can get an accurate reproduction of the video display with a screen-print routine. In fact, one of the pieces of literature included with the review copy of the LCA-47 was an electronic-circuit diagram produced on a Line Printer VII. The diagram included both uppercase and lowercase text and schematic symbols.

### Alternative Character Sets

Since the LCA-47 uses a separate ROM to define the characters, anything that can be drawn within an appropriate area can be generated. The area within which the character must be defined is eight dots (video resolution elements) wide and 12 scan lines tall. At present you can purchase four character-generator ROMs from the manufacturer: the standard upper/lowercase furnished with the board, upper/lowercase with lowercase Greek letters, upper/lowercase with thinline graphics and electronic symbols, and uppercase with selected Japanese katakana characters.

In each case, the English letters are compatible with Basic programs, while the special symbols are accessed by POKEing appropriate values into the display RAM. A blank bipolar PROM is available if you want to do your own programming, and additional character sets are promised. Custom programming is also available.

### In Summary

The LCA-47 is a well-made product that is supported by the manufacturer. The instruction manual is also of high quality.

It is interesting to speculate about possible enhancements. For example, I can imagine a larger board with at least two character generators, selectable by another DIP switch or from software.

For the present, the LCA-47 makes the Color Computer a more finished product, and deserves the consideration of all Color Computer users engaged in text processing. ■

# DAISY WHEEL PRINTERS...

- SMITH CORONA TPI
- COMRITER CR-1
- DAISYWRITER 2000
- DIABLO 620/630 KSR
- QUME SPRINT 9 KSR

# COMPUTER SYSTEMS...

- EPSON HX-20
- TRS-80 MODEL III

## $$$ CALL FOR PRICE $$$

**You won't believe it!!!**

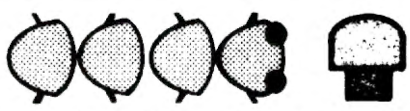
**Rainbow**
**P & P CORPORATION**
PO BOX 362 • HADDONFIELD, NJ 08033

264

**800-257-6170** *in NJ call* **609-428-3900**

---

# COLORPEDE



An exciting new game from the company that is setting the standards. Colorful, high scoring, fast action play with arcade quality sound effects. High resolution, multi-colored characters on a black background. Smooth accurate joystick control. Demonstration mode. Pause feature. 1 or 2 players. 100% machine language. Requires 16K color computer with joysticks.

Cassete—$29.95 Disc—$34.95
Add $1.50 for shipping; $3 outside U.S.; 4% tax in Mich. VISA, Mastercard or Money order. Please allow 2 weeks for checks.

DEALER INQUIRIES INVITED

**intracolor**
**COMMUNICATIONS**
**SETTING THE STANDARDS**
6048 Horizon Drive, East Lansing, MI 48823

TRS 80 Color Computer is TM of Tandy Corp 336

---

# DISKETTE JUNCTION


CALL NOW AND
**SAVE**
ON THIS SPECIAL INTRODUCTORY OFFER

## Scotch DISKETTES

**$19.95 per box of 5¼"**

1 side/D Density (744D-O) diskettes compatible with Verbatim (MD525-01)
Limit two per order.

## Scotch Head Cleaning Kits

5¼" and 8" Kits
**$19.95 each**
Limit one per order.

DISKETTE-JUNCTION stocks a complete line of 3M magnetic media including 96TPI Quad Density. Call now to order the Scotch diskette for your equipment.

**TOLL FREE**
**800-321-5134**
**OHIO RESIDENTS CALL**
**800-362-5134**

**DISKETTE JUNCTION** 5
5918 Smith Rd. • Cleveland, Ohio 44142

WE ACCEPT
VISA • MASTER CHARGE • CHECKS
MONEY ORDERS • C.O.D.

- Add $3.00 for shipping & handling.
- $2.00 EXTRA for U.S. Mail delivery.
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6.5% sales tax.

Authorized Distributor
Information Processing Products
3M

★★★★½

**Hypergate Centurion**

★★★★

**Hypergate Patrol**
**Synergistic Solar Inc.**
**Box 560595**
**Miami, FL 33156**
**16K**
**Model I or III, cassette or disk**
**$39.95**

**by Mark E. Renne**

Synergistic has developed two very sophisticated space simulations that require only 16K and yet are stimulating to even the best space cadets. They propel you into a universe filled with monsters, exploding asteroids, marauding pirates, and anxious allies.

The games, which are sold together, are related to each other. In Hypergate Patrol, you are the captain of a small patrol vessel wandering through the galaxy at the command of the Hypergate Centurion. In Hypergate Centurion, you control several Hypergate Patrol ships and a plasma weapon that spans the universe. The inhabitants of both games are identical, but the strategy is very different.

Hypergate Patrol is a program written in Basic with machine-language subroutines. Each galaxy is comprised of nine quadrants, and five galaxies make up your patrol universe. The galaxies are all joined in the center by a Hypergate. By entering the Hypergate at different angles, you can travel from one galaxy to another. For example, if you're in the Androm galaxy and receive a distress call from the Solan galaxy, enter the Hypergate at a 238 - 274 degree angle. Your destination will be Solan.

Your performance is rated on a merit-demerit system. Killing the bad guys rates five merits while killing a friend rates five demerits. You need a ratio of seven to one to survive. The game is played in real time and you must complete your mission within a specified amount of cycles.

Although the program is written mostly in Basic, it moves quickly and features 50 different aliens. At any given time a certain alliance will be the enemy of the guild and you must destroy any of that alliance's warships. Also, you must destroy pirates and monsters. Beware, though; pirates might be disguised as other ships and are only discovered by interrogation.

The video display is divided into seven sections, providing all the vital information. Your ship features a variety of weapons, and a call to the Hypergate Centurion is always possible in a tight spot.

Hypergate Centurion is a machine-language, multi-level, space simulation of the highest degree. You are buried deep within an asteroid and control 15 quadrants of space. You have a fleet of five seekers in the hangar bay, ready to be dispatched with any of five different orders.

As each object enters your jurisdictional area it is assigned a number. You can scan the area in four different ways—area, quadrant, tactical, or interrogation. This gives a variety of magnification and types of information depending on the scan.

Documentation for these two games consists of 45 pages of instructions in the form of an operations manual full of Centurion secrets. These are not arcade-type games, but simulations requiring skill, logic, and mental expertise. Even though they require only 16K, they are first class and offer a challenge that won't be mastered in one sitting. ■



★★★★

**Date-O-Base Calendar**
**Custom Software Engineering**
**807 Minutemen Causeway**
**Cocoa Beach, FL 32931**
**Color Computer**
**$19.95 disk**
**$16.95 cassette**

**by Scott L. Norman**

Keeping a calendar is a worthwhile task for any home or business computer. Date-O-Base is an inexpensive program that turns the Color Computer into an electronic datebook able to search and display short memos previously stored for a particular date. It also includes a calendar-page display mode that takes advantage of the Color Computer's high-resolution graphics capabilities.

Date-O-Base, or DOB, is available for cassette and disk systems; disks are best for the file handling this type of program requires. I reviewed disk DOB, intended for a 32K system with a single drive.

The program comes on tape, so your first task is to copy it onto a blank disk. There isn't much to this, since DOB is written in Basic: just CLOAD and save. When the program is first run, two ASCII files called MEMO/IND and MEMO/CAL are automatically created. These expand as the calendar fills with memos. A disk can hold more than 4,000 one-line memos of up to 28 characters each. There are restrictions, though: no more than 12 memos for any single day, no more than 300 for any month. This should be more than adequate for any household or small office. The program even allows multiple lines for complicated messages.

Although DOB's date-computation routine can handle any day of any month between the years 1700 and 2099, the program will probably be used for events in the near future. The input routines have been fine-tuned a bit; any year in the 20th century can be specified by giving its last two digits. Months can be specified by a number from 1–12 or by a three-letter abbreviation. The months, days, and years required as input must be separated by commas.

When DOB is run (you have to enter the Run command twice when first starting a session), a master menu containing six options presents itself. This is home base; you can get back to it by hitting enter once or twice. Here are the details of operation:

● Option 0, End, must be used for all normal exits from DOB. Besides being used at the end of a session, you should select Option 0 after you have entered a series of memos that you will later want to search. It closes the disk files and carries out other housekeeping chores. It also terminates the program and returns Basic's OK prompt, so that an additional Run command is necessary to get back into action.

I found this to be the only tricky aspect of DOB's operation. I forgot to use Option 0 a few times, and wiped out some test files before I could check out the rest of the system.

● Option 1, Display Month, prompts you for a month and year, and then draws and numbers the appropriate calendar page using high-resolution graphics. You can enter memos for any day during the chosen month by typing in the date. The numbers you type will not show on the screen, but hitting enter causes the display to switch to a text screen that does show the date, together with 12 numbered lines for memos. You are then prompted for a memo number. When you have finished your entries, a null line (the enter key by itself) returns the calendar display. Now, however, the box for the date in question contains a reminder spot for each memo. When you have finished your month's entries, another null line gets the master menu back.

The data on any given page can be changed by overwriting. A memo can be deleted by specifying its number and entering a null line. This is fine for individual deletions, but there are easier methods for handling blocks of data.

It is possible to get a printout of the calendar display, but this requires an external screen-print routine—Radio Shack's, for instance, or one sold by Custom Software themselves.

● Option 2, Display/Change Memos, bypasses Option 1's calendar-drawing routine and saves some time. Option 2 lets you specify a date and jump right to the text screen. The entry, modification, and deletion of memos proceeds just as for Option 1.

*"Date-O-Base... fills a genuine need at a reasonable price."*

● Option 3, Search Memos, displays or prints memos falling within a specified period of time. You can read all the memos, or use keywords and search logic to restrict the output.

When you select Option 3, you are first prompted for start and finish dates for the search interval. Next, specify the search logic: AND, OR, or ALL. ALL bypasses the use of keywords and outputs everything found in the interval. Both AND and OR allow you to specify up to 10 keywords. The AND option only displays memos containing all of the keywords, while OR selects memos containing any one of them. If you are interested in only one keyword, either option will do.

● Option 4, Delete Memos, is a global delete function, eliminating all the data for a specified interval. The minimum interval is one month; both starting and finishing dates must be specified in MM/YY format, but you can specify the same month for both. DOB is a friendly program, and gives you a last chance by asking if you are sure before actually deleting memos.

Option 4 provides an alternative to the one-memo-at-a-time deletion capability of Options 1 and 2. You can also use Disk Basic's Kill command to eliminate the MEMO/IND and MEMO/CAL files. Of course, this wipes out everything but the original DOB program itself.

● Option 5, Date Computation, employs DOB's date-computation routine to compute the elapsed time between two specified dates. The output takes four forms: days, weeks/days, months/days, and years/months/days. There is one quirk: The first two display modes takes account of February 29 for leap years, while the last two do not (unless the elapsed time is less than one month).

Date-O-Base has quickly become a member of my Color Computer's assortment of household programs. It is easy to use (provided that you remember Option 0), and fills a genuine need at a reasonable price. If you can live with the short memos imposed by the program, it can do a good job for you. ■



✓See List of Advertisers on Page 451

★ ★ ★ ½

**Drive Control Unit**
**Optronics Technology**
**2990 Atlantic Ave.**
**Penfield, NY 14526**
**$39.95, assembled**
**$29.95, kit**

**by David DuPuy**

It has long bothered me that my eight-inch disks continue to spin in the drive after I have accessed the files I need. The wear on the disk drive, and shortening the life of the floppy disk, seems unnecessary.

Now, eight-inch drive systems can be adapted to turn the drives off when they are not needed (see New Products, *Microcomputing*, May 1982, page 140). Optronics Technology's Drive Control Unit (DCU) is available assembled and tested, or in kit form. The DCU can be installed in existing eight-inch drive enclosures with a minimum of effort.

### Concept and Features

I ordered the Drive Control Unit in kit form from Optronics. A six-page manual, with two pages of drawings and diagrams and a photograph showing how the unit mounts in the rear of an eight-inch Shugart drive comes with the kit.

The DCU's concept is simple. The 110 volts ac that is normally connected to the synchronous motor for rotating the disk is turned on or off by a triac mounted on the Optronics DCU board. A trigger signal from the Shugart PC board controls the triac, so the computer needs no extra wiring. The triac is opto-isolated from the logic, as you might expect. To minimize electrical switching noise, the ac voltage turns on or off at zero crossing. To turn a second drive on and off, Optronics provides a second control input and a second output. I could not test that option, since I have only one disk drive. After you access and release the drive (per normal operation before inserting the DCU in the ac line), the disk spins for about nine seconds after the LED goes out before shutting off. This nine-second interval can be changed when building the unit by changing one resistor.

### Construction

The printed circuit board is about 2 3/8 by 3 1/2 inches in size and comes predrilled. There are about 20 parts to mount on the PC board. The instructions are adequate, and no one who has any experience at the Heathkit level should have any problems assembling the board. A drawing shows the placement of parts to be mounted, and I found no ambiguities in the instructions.

I do have a couple of suggestions for comments to be included in the next version. I saw no mention that parts mounted on the board should be bent parallel to the board (it is obvious that the triac should be bent over); the board layout is arranged so that the capacitors bend over to conserve vertical room, and the manual should mention this. Mechanical interference is possible between the molded nylon socket for ac input and the wires to the ac output socket. It would be simple to move the leads to the ac output socket 1/4 inch and avoid this problem. Also, the diagram showing hookup to the trigger signal on the Shugart PC board doesn't look like the PC board on my Shugart 801. Finally, although all necessary information has been included, the manual could use a little more organization.

The manual slants toward wiring that conforms either to Shugart or Siemens drives. If you have either of these, every detail you need to build and install this device is in the manual. The manual makes clear those changes that have to be made for different types of drives, and offers several options. A few extra components are included (signal diodes) in case you have a drive that requires slight wiring changes on the PC board.

Installing the PC board in the Shugart drive is easy. The molded nylon plugs included are identical to those used to connect 110 volts ac to the Shugart drive motor, and this PC board is inserted in series in the 110-volt ac line. The molded nylon plugs are mounted on the PC board so that the 110-volt leads support the board, and the cables in the drive need little rearrangement to accommodate the board. Five-volt and ground leads attach to the Shugart PC board, and the only other lead to connect is the trigger signal, which also goes to the Shugart PC board. For the Shugart drive, the manual instructs you to connect this lead to test point 11 (TP11) on the Shugart board. As I mentioned before, the diagram showing this connection does not resemble my Shugart 801 PC board. However, I located a pin marked 11, and that evidently was the right connection, since the unit works as advertised.

### Performance

Upon installation in my system (with a CCS disk controller and Shugart 801 drive), the Drive Control Unit worked fine the first time without fine tuning or troubleshooting. The lack of drive noise and continual disk spinning is a delight!

I did experience a few problems. A disk-error message occasionally shows up when I first boot up my system, and typing B a second time (my monitor's command to boot from the disk) boots the system from the disk. The DCU manual recommends you lengthen the delay on the disk controller's HLD signal to about 360 ms. This did not solve the error message on boot-up, and I decided to leave the delay around 300 ms. This particular error might not show up on other systems. In the auto-boot mode, this problem did not occur, although I had to press reset after turning on the machine while the disk was running to boot the system. The manual gives suggestions for troubleshooting in case other disk errors occur.

I have one problem that was probably not anticipated by the designers, and that is because I'm using only one disk drive. In copying from one disk to another using a single-disk, file-transfer program, problems occur if the nine seconds expires before changing the disks. The error message "Drive A Not Ready" appears. Nothing serious happens; you just have to reboot and try again. Switching disks within the nine-second time limit is easy (you don't have to rush), and then the single-drive copy routine performs satisfactorily. You can lengthen the nine-second interval before turnoff if desired.

### Summary

This product serves a needed function. The price is reasonable and the device works with no debugging. The disk-drive motor now shuts down nine seconds after being disengaged. Disk and drive wear has been greatly reduced. The unit mounts easily in the rear of a Shugart eight-inch drive without modification to the drive. You must solder three wires to the Shugart PC board to complete the installation. The instruction sheets are detailed enough to guide even a novice. ■

# REVIEW DIGEST

*Understanding Computer Science*, Roger S. Walker, Tandy/Radio Shack, Catalog #62-1383, $2.95.

"I picked up *Understanding Computer Science* while waiting in line at Radio Shack, and I couldn't put it down. . . . I'm convinced that it's a little-known treasure that belongs on any computer owner's bookshelf. This book is an inexpensive overview of all the major areas of computing, written at a level that any interested person can understand, and the information that it contains is perfectly packaged for home computerists." *Compute*, September, p. 100.

---

**Postman**, Soft Sector Marketing, Garden City, MI 48135, Model I or II, $125, $175 (form letter option), $225 (hard-disk version).

"Postman fills a variety of small-business applications that require large mailing lists. It is written entirely in machine language and is menu-driven for simplicity. . . . There is only one way to describe the performance of Postman: 'Flawless!' " *InfoWorld*, September 20, p. 47.

---

*Basic For Business*, Douglas Hergert, Sybex Books, Berkeley, CA, $12.95.

"*Basic For Business* is written for the businessperson who wants to know how and why a program works. It approaches the study of Basic from a businessperson's perspective; that is, it guides a reader through the development of several practical business programs, including a cost of goods sold inventory program, break-even point analysis, T-ledger accounts, depreciation, and basic statistical analysis programs." *Personal Computing*, October, p. 149.69.

---

**Master Control**, Soft Sector Marketing, Garden City, MI 48135, Color Computer, $24.95.

"Alan Schwartz is to be congratulated on an innovative and highly useful program. It is written in machine language and from the first worked as the very brief, but adequate, instructions indicated. . . . The first few days living with Master Control the kids and I keyed in some programs . . . Master Control makes that task much easier and faster." *Color Computer News*, September, p. 29.

---

**Pro-80 Extended Basic**, Pro-80 Systems, Cedar Falls, IA, Model II and 16, $195.

"Pro-80 Basic does what it claims it can do: It expands the capabilities of Radio Shack's Basic interpreter. Before you buy, you should be aware that some of the features contain potential hazards, and the sort is not as fast as some I've seen. However, my overall recommendation is that the package is worth its price." *two/sixteen*, July-August, p. 16.

---

*Big Things from Little Computers: A Layperson's Guide to Personal Computers*, Dale Peterson, Prentice-Hall Inc., Englewood Cliffs, NJ, $12.95.

"Peterson discusses a variety of unique uses for personal computers in education, art, music and business, in the home, and by the physically handicapped. . . . It is clear that *Big Things From Little Computers* is not a how-to book, but rather a resource book of better personal computing ideas and applications that can be implemented with a little effort and imagination." *Personal Computing*, September, p. 156.

---

**MicroPilot**, Tandy/Radio Shack, Model I and III, 32K disk format, $79.95.

"The strengths of MicroPilot lie in its programming efficiency for educational/interactive programming. With MicroPilot you can ask a question, get an answer, evaluate the answer and appropriate feedback with three of four simple commands. . . . The features that make MicroPilot shine are its edit/run capability, its graphics, its sound generation capability, and its record keeping commands." *Creative Computing*, October, p. 34.

---

**The Tiny Compiler**, Aardvark-80, Walled Lake, MI 48088, Color Computer, $24.95.

"The name is appropriate because the Tiny Compiler does not pretend to be a sophisticated big-time compiler that will take whatever code you wish to write in Basic and generate machine-language instructions. Instead, and to the credit of Aardvark-80 which markets it, Tiny Compiler is billed as a limited compiler which, frankly, can speed up a lot of things that needed speeding up in the first place." *The Rainbow*, September, p. 89.

---

**Stellar Escort**, Big Five Software, Van Nuys, CA 91409, Model I and III, $95.

"Your job is to protect the defenseless craft you are towing from the vile Cretonians who are out to destroy it. It is to your advantage to protect your craft well, because if it is destroyed the disruption of the force field immediately destroys your escort craft. . . . As compared to other games on the market in this price range, you will definitely get your money's worth from this one." *80 U.S.*, October, p. 102.

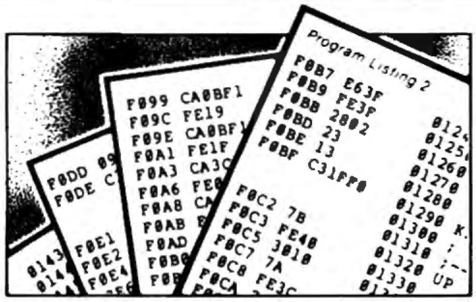
---

**RAM Slam**, DSL Computer Products, Dearborn, MI 48121, Color Computer, $49.95.

"At $49.95 the RAM Slam upgrade kit from DSL Computer Products is one of the least expensive kits of the 'solderless' type I have seen advertised. . . . Not being the technical type I was very pleased to see how easy it was to upgrade to 32K . . ." *The Rainbow*, September, p. 88.

---

by Roger Fuller




# Display bits from bytes

Last December, I discussed bits and bytes. This month's column provides a short program that lets you examine the bits in the bytes of binary numbers.

The Assembly program takes advantage of a ROM routine. When the Basic interpreter encounters a USR function, it evaluates the function's argument. This sets the variable type flag at 40AFH and leaves the type in the A register. Therefore you need only access the A register to get the type, which also happens to be the number of bytes used to represent the number.

The type flag for an integer is two; for a single-precision number, four; and for a double-precision number, eight. If the binary number is an integer or a single-precision number, the bytes representing the number are stored starting at 4121H. A double-precision number is stored starting at 411DH. Table 1 summarizes this.

The Assembly code that displays the bits on the screen's top row is shown in Program Listing 1. These bits are displayed as ones if they are on, zeros if they are off. To determine the number of bits, multiply the type by eight. A double-precision number uses the entire top row of 64 spaces to display its bits. The bits are displayed with the least significant bit at print position 63 on the screen. The more significant bytes in a number are displayed to the left of the less significant bytes, the opposite of the way they are stored in memory. A byte of value 3 would be displayed as 00000011. Since the type flag is equal to the number of bytes in a number, the Assembly program uses it to set up a loop. This loop determines how many times to call the subroutine that displays the bits. A DJNZ loop structure accomplishes this. The HL register pair is initialized to the memory address where the argument of the USR function was placed after evaluation by Basic. The CP 8 in line 150 allows the adjustment for double-precision numbers to be made in line 170.

The DE register pair is used to point to the screen.

In line 190 the #-bits-to-display counter is saved by the PUSH BC. Line 200 loads a byte from the number. Next, a Call is made to the display bits subroutine. Here a bit mask is created in the C register by loading it with a one in line 280.

The mask works by ANDing it with the target byte stored in A. If the bit in the mask and the bit in the target byte are not on, the result is zero. Since the AND instruction destroys the value of the A register, the value is saved in the B register in line 290 and recovered each time it is needed in line 300. The load in line 320 does not alter the flags, so the jump in line 330 is still valid. The ASCII zero, or if the bit was on, the ASCII one is displayed on the screen.

Next, the screen pointer, DE, is backed up and the mask rotated left, which also performs a test for zero. This is a convenient way to exit the subroutine at line 390.

The Basic program (Program Listing 2) assumes you have loaded the machine code at address 7F00H. You will need to POKE the values for the USR address if you do not have Disk Basic. The program requests the type length in line 40 and validates the answer. Then a subroutine is called in line 50 to set the variable's type appropriately. This allows the program to easily change variable types.

The TYPE/2 + (TYPE = 8) expression in line 50 evaluates to a 1, 2, or 3 for integer, single-precision, or double-precision variables respectively. When (TYPE = 8) is false it is evaluated as zero; when true, it is evaluated as -1. Thus, 8/2 + (-1) equals 3. This works with almost all Basics. (Level I is not one of them.)

Line 60 requests the initial value of the number you wish analyzed. Line 70 requests the step value. If you want to examine just one number, answer with a zero.

Displaying double-precision numbers is impressive and demonstrates why processing them takes so long. ■

```
                00100           ;BIT DISPLAYER BY ROGER FULLER   PUBLIC DOMAIN
                00110           ;
7F00            00120           ORG     7F00H
7F00 47         00130           LD      B,A         ;SAVE NUMBER LENGTH
7F01 212141     00140           LD      HL,4121H    ;POINT TO NUMBER STORAGE
7F04 FE08       00150           CP      8           ;CHECK FOR DBL LENGTH
7F06 2002       00160           JR      NZ,$+4      ;IF NOT DBL THEN SKIP
7F08 2E1D       00170           LD      L,1DH       ;ELSE BACKUP 4 BYTES
7F0A 113F3C     00180           LD      DE,3C3FH    ;063 ON VIDIO SCREEN
7F0D C5         00190 NUMBER    PUSH    BC          ;SAVE LOOP COUNTER
7F0E 7E         00200           LD      A,(HL)      ;GET A BYTE
7F0F 23         00210           INC     HL          ;POINT TO NEXT BYTE
7F10 CD177F     00220           CALL    BYTE        ;DISPLAY 8 BITS
7F13 C1         00230           POP     BC          ;RESTORE LOOP COUNTER
7F14 10F7       00240           DJNZ    NUMBER      ;LOOP TILL DONE
7F16 C9         00250           RET                 ;DONE
                00260           ;
                00270           ;
7F17 0E01       00280 BYTE      LD      C,1         ;MASK FOR BIT POSITIONS
7F19 47         00290           LD      B,A         ;SAVE BYTE
7F1A 78         00300 BITOUT    LD      A,B         ;RECOVER BYTE
7F1B A1         00310           AND     C           ;TEST FOR BIT ON
7F1C 3E30       00320           LD      A,'0'       ;LOAD A ASCII ZERO
7F1E 2801       00330           JR      Z,$+3       ;IF BIT IS OFF
7F20 3C         00340           INC     A           ;IF BIT ON MAKE '0' A '1'
7F21 12         00350           LD      (DE),A      ;DISPLAY '0' OR '1'
7F22 1B         00360           DEC     DE          ;POINT TO NEXT SPACE
7F23 CB21       00370           SLA     C           ;MOVE MASK BIT LEFT ONE
7F25 20F3       00380           JR      NZ,BITOUT   ;IF NOT DONE
7F27 C9         00390           RET
                00400           ;
0000            00410           END
00000 TOTAL ERRORS
```

*Program Listing 1*

```
10 ' BIT DISPLAYER BY ROGER FULLER   PUBLIC DOMAIN
20 CLS : DEFUSR1=&H7F00
30 PRINT@128,"ENTER TYPE OF VARIABLE (2=INT, 4=SNG, OR 8=DBL)";
40 INPUT TYPE : IF TYPE<>2 AND TYPE<>4 AND TYPE<>8 THEN 30
50 ON TYPE/2 +(TYPE=8) GOSUB 200  ,400  , 800
60 PRINT@256,"ENTER INITIAL VALUE"; : INPUT NV
70 PRINT@384,"ENTER STEP SIZE"; : INPUT NS
80 PRINT @ 512, "NUMBER = ";
90 PRINT @ 520, USR1(NV);
100 NV = NV + NS : GOTO 90
200 DEFINT N : RETURN
400 DEFSNG N : RETURN
800 DEFDBL N : RETURN
```

*Program Listing 2*

MSB = Most Significant Byte
NMSB = Next Most Significant Byte
LSB = Least Significant Byte
EXP = Exponent Value
SIGN = Sign Bit

| | INTEGER | SINGLE | DOUBLE |
|---|---|---|---|
| 411D | | | LSB |
| 411E | | | NMSB |
| 411F | | | NMSB |
| 4120 | | | NMSB |
| 4121 | LSB | LSB | NMSB |
| 4122 | SIGN/MSB | NMSB | NMSB |
| 4123 | | MSB | MSB |
| 4124 | | SIGN/EXP | SIGN/EXP |

*Table 1*





✓See List of Advertisers on Page 451


*THE SAD PLIGHT OF A FROG. HE FACES WILD DRIVERS AS HE TRIES TO CROSS THE ROAD. SINKING TURTLES, FLOATING LOGS AND HUNGRY ALLIGATORS STAND BETWEEN HIM AND HOME. HE NEEDS YOUR HELP!!* ***ORDER SOON BEFORE ITS TOO LATE!!***

SOUND EFFECTS

TOP TEN SCORES SAVED ON TAPE OR DISK

**TRS-80* MODEL I - III**

Disk $19.95 Tape $15.95

Joysticks available for the above program for $39.95

TO ORDER SEND CHECK OR MONEY ORDER OR CALL (517) 542-3280

**Visit Our Two RETAIL LOCATIONS:**
*111 Marshall St.
Litchfield, MI 49252
517-542-3280
886 Ecorse Rd., Ypsilanti, MI 48197
313-482-4424

*Authorized Radio Shack Dealership

62

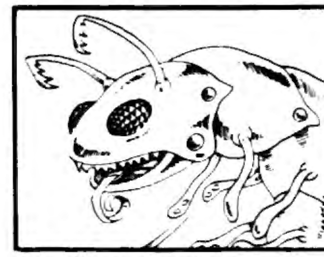
DISPLAYED VIDEO

**DISTRIBUTED EXCLUSIVELY BY DISPLAYED VIDEO**
**DEALER DISCOUNTS AVAILABLE**
**WRITTEN BY: DuBois and McNamara**

ADD $2.00 for Shipping. Add $1.50 for COD. Michigan Res. Add 4 percent.

*TRS-80 Trademark of the Tandy Corporation

# Extra! Extra!



Insect Frenzy - Page 5

Volume 4 | All The News That's Fit To ***ZAP!***

# *Users vote no to the same OLD ARCADE GAMES*








**(DV 1982)** Blurry eyed users have turned to **DISPLAYED VIDEO** to answer their need for new and exciting arcade games. In response, **DISPLAYED VIDEO** has announced eight new programs for the TRS-80* Models I-III. These arcade type games feature sound, graphics, joystick compatibility and are written in machine language for maximum speed! Both disk and tape versions allow the user to save high scores, a feature not usually found on cassette based games. Maze enthusiasts seem to like Ghost Hunter and Killer Beetles, while gun slingers look toward Insect Frenzy, Jungle Raiders, Space Shootout, Alien Cresta and Battle Stations for excitement. A Game that does not fall into these categories is Hoppy. It features wild drivers, sinking turtles, and hungry alligators. These programs are distributed exclusively by **DISPLAYED VIDEO** and written by Dubois and McNamara. Pricing for these programs is **$15.95** for tape and **$19.95** for disk. Reliable sources inside the company indicate Killer Gorilla will be available by the time you read this.

TO ORDER SEND CHECK OR MONEY ORDER OR CALL (517) 542-3280

**Visit Our Two**
**RETAIL LOCATIONS:**
111 Marshall St., Litchfield, MI 49252
(517) 542-3280
or
886 Ecorse Rd., Ypsilanti, MI 48197
(313) 426-5086/(313) 482-4424

*Authorized Radio Shack Dealership

DISPLAYED VIDEO

DEALER DISCOUNTS AVAILABLE

Joysticks for above programs **$39.95**
Add **$2.00** for Shipping
Add **$1.50** for COD
Mich. res. add 4 percent tax

*TRS-80 is a trademark of the Tandy Corp
Prices subject to change without notice

# The GAMER'S CAFE

by Rodney Gambicus

## How long can you tread water?


We awoke to the dull staccato of rain beating on the van roof, as we had every morning during our stay in Eugene, OR. The doors had rusted shut overnight and Winthrop started screaming, "We're all gonna die, we're all gonna die," so I strapped him to his bean-bag chair and cranked up Hellfire Warrior on the Model I, and he quickly slipped back into his coma.

It was Winthrop's idea to come to Eugene because he'd heard that there were a lot of crunchy-granola types running around, and thought maybe we could find some New Age health freak who could figure out what was wrong with the PMC (which we affectionately call the Model ½) we'd found at the Goodwill in Pocatello, ID. I tried to tell him that foot reflexology wouldn't work on a computer, but Winthrop is a stubborn sort. It took me six weeks to potty-train him.

It sure didn't look like we'd be opening the cafe that day. So I wiped the green slime from my glasses and crawled out the window and padded on over to the Atrium, where I grabbed a cup of espresso and a Prince Puckler's chocolate milk shake for breakfast. I had Queenie, one of our Model IIIs, along, so I plugged her in and booted up Adventure International's Sea Dragon.

Well, naturally several shoppers wandered over to see what was happening, and before I knew it they'd pushed me aside and were cruising along like so many underwater Ahabs. What with all of the rain perpetually dropping from the sky, these people seemed to have a particular affinity for the video deep, and pretty soon they'd blown Winthrop's high score off the board. Some fellow who called himself Mad Max wiped out the reactor five times and racked up 147,910 points.

Mad Max was a hippie tree planter who hadn't been out of the woods since 1976, which was when he finally got back from Woodstock. He thought Queenie was pretty interesting (he called her a "souped-up tv"), and wanted to know if anybody'd ever gotten a score higher than his. I had no idea. So we went back to the van and woke up Winthrop and asked him, but he had no idea either.

Well, says I, why don't we put out a call in Gamer's Cafe? But Winthrop was a bit miffed that this guy, who didn't know a chip from cheddar cheese, had cleaned his clock, and wasn't too receptive. Then Mad Max said, why not publish Winthrop's high scores for other games? This suggestion thawed Winthrop pretty quickly.

So I started scrounging around the van for the bits and pieces of paper Winthrop filed his scores on, and typed them on a separate sheet of paper, which I imagine the guys at *80* will print as a table or figure or something. (See Something 1.) If any of you out there have done better, or have a super score for a game not on the list, send a photo of the screen to the Gamer's Cafe, care of *80 Micro*.

* * * * *

About the time we heard a knock on the window. It was the mailman. Lord only knows how the *80* office manages to track us down. Winthrop was all excited—he loves answering mail from our readers—but the mail consisted of a flyer from the Church of Otherly Love, a free sample of a new fingernail remover, and a press release from Hayden Software. None of it interested Winthrop much, so when Mad Max suggested they go out for a mushroom pizza, Winthrop jumped at the offer.

Since it seemed likely that the Hayden release would have something to do with micros, I opened it up, and found an interesting couple pages on their Kamikaze Shootout. Kamikaze is a new arcade game for the Apple, and Hayden is sponsoring a contest that will culminate (i.e., end) in a showdown in San Francisco for 5,000 smackers. The finalists will be the five people who send in the highest scores by January 31, 1983; they have to verify their scores with a photo of the screen.

You might be wondering what's to stop somebody from messing around with the code and posting a high score illegally. I wonder if you're wondering that because I wondered it too. So I picked the lichen from my pants and headed for the nearest phone to call Jim Hunter, software editor at Hayden.

Well, the honchos at Hayden are pretty laid back about the whole thing. Jim says they've got a protection scheme built into the program, and a

| Game | Player | Score |
|---|---|---|
| Eliminator | Winthrop | 59,600 |
| Sea Dragon | Mad Max | 147,910* |
| Swamp Wars | Winthrop | 39,200** |
| Armored Patrol | Winthrop | 81,000† |
| Galaxy Invaders | Winthrop | 1,000,000†† |
| Cosmic Fighter | Winthrop | 103,980 |

*Novice mode. Max got 69,480 in the expert mode
**Winthrop got through all nine swamps, too.
†Method I. Winthrop racked up 281,000 points using Method II.
††Winthrop still had six ships left, but he got bored

*Something 1. A list of the most-frequently played games at the Gamer's Cafe and their high scores.*



✓70

✓See List of Advertisers on Page 451

message that will flash onto the screen at 50,000 points. But basically they're relying on the integrity of the contestants. Besides, he points out, who's going to want to get to San Francisco by cheating and then get humiliated at the national finals? (He's lucky he didn't ask Winthrop that question.)

In any case, Jim and I marveled at what a great idea it was, and wondered why other companies didn't do it. Jim pointed out that such a contest requires a lot of up-front cash for advertising and promotion (not to mention the prizes), and that only the larger companies could afford to risk the loss. Still, it sure would be nice if somebody did something similar for us TRS-80 pilots.

Speaking of Hayden, Jim also says that they won't be coming out with too much TRS-80 software in the near future. It seems they've got eight or nine Apple and Atari games in the works that they need to polish off. Jim acknowledged, however, that the company can't afford to ignore three quarters of a million of us, so maybe some time in the future they'll toss something our way.

While I was on the phone, I decided to make one of my rare yet infrequent calls to *80 Micro* headquarters. One of the editors answered with, "Casey's Bar and Grill, Casey speaking." Jeez, I thought, those folks at *80 Micro* sure are a pack of cards. Anyway, I asked about these rumors going around about Spectral Associates, and the editor said sure enough, they're true. Spectral is coming out with a machine-language hi-res graphics adventure for the Color Computer called Revenge of the Hawk, and the people at Spectral say it's going to set a new standard in adventure games. We've all heard that one before, so we'll see. They've also got Omega Race ready to go on line. Both Hawk and Race should be out at any time. In addition, they have a machine-language hi-res Donkey Kong play-alike in the works for you weird arcade parlor bums.

After I got all the straight dope on Spectral, I asked how things were going at *80 Micro*, and the guy on the line said "80 what?" I thought at this point that the joke had gone too far, so I shouted at the guy for a little while, and he shouted back. It finally came to light that I really *had* called Casey's Bar and Grill. These bartenders think they know everything.

* * * * *

When I got back to the van, Winthrop and Mad Max were staring happily at a blank VDT, giggling and passing a bottle of margaritas back and forth. I asked what was going on, and Winthrop said something about the hilarious ultrasonic hum. Mad Max asked me if I wanted a piece of mushroom pizza. Only thing was, they'd somehow lost the pizza and only the mushrooms were left. I politely declined.

Well, I was getting pretty sick of the rain, so I figured that this was as good a time as any to hit the road. We needed some place nice and sunny where we could set up the cafe once again and sip coffee to the sounds of twangy Spanish guitars. So I wiped the moss from the windshield and headed for Route 5 South.

It wasn't until Klamath that I realized we still had Max in the van. I thought of saying something, but he was asleep, and I didn't want to wake him. Besides, this'll be good for his soul. How long can a guy plant trees before he becomes one?

That's it for this month. Keep those cards and letters pouring in. ■

# I've paid off the costs on my Model II TRSDOS* systems, too.

# So now they're 50% cheaper.

A couple of months ago, I realized I'd paid off the development costs on my Model I and Model III programs. I could lower the price without cutting back one bit on my support.

The response was fantastic. Enough so that I can do the same for the TRSDOS versions of my Model II/16 programs.

These are my systems, and my new prices.

**General Ledger**
It gives year-to-year comparisons in dollars and percentages. It figures budgets and even has a report generator. It was $299, it's now $150.

**Accounts Payable/Purchase Order**
It generates purchase orders and posts the items to payable when the goods come in. It calculates and prints checks and aged ledger reports, linking fully to the General Ledger. Was $349, now $175.

**Accounts Receivable**
You can choose either an open item system or a balance forward system which works on a cash or an accrual basis. The open item system does invoicing and sales analysis by product code and figure in salesmen's commissions. They both generate mailing lists by customer code and zip code for up to 2000 customers. Open Item/Invoicing was $349, it's now $175. Balance Forward was $399, it's now $200.

**Payroll/Job Costing**
A huge capacity. It accommodates up to 300 employees in multiple departments, with any state tax routine. It also figures piecework, overtime and tips. Was $299, now $150. With job costing option, was $399, now $200.

**Inventory Control**
It stores up to 5000 items. It reports by vendor, tells you when you're out of stock, or when you need to reorder. It updates price or cost automatically and integrates fully with my invoicing system. Originally $399, now $200.

These programs all work with one, two, three or four-drive and hard disk systems. They're designed to integrate with the General Ledger, and, where it helps, with each other.

They also get what I firmly believe is the most thorough support in the microcomputer industry. If you have a problem, call us and we'll straighten it out. Even if I have to do it myself, personally, right there on the phone.

Michael Tannenbaum, the "80 Accountant" thought my systems were "a very impressive product at a very reasonable price." Even when they cost twice as much as they do now.

Just call, and take advantage of me.

✓70

**Taranto**
& ASSOCIATES, INC.

121 Paul Drive, San Rafael CA 94903
Outside California, toll free (800) 227-2868. In California, (415) 472-2670.

*A trademark of the Tandy Corporation.

✓ See List of Advertisers on Page 451

# Voice-Controlled Typewriter

by Mike Rigsby

**The physically handicapped can produce printed text with this sound-controlled program.**

Some handicapped people have disabilities that prevent them from writing or typing, yet their intellect is normal. This program lets you produce printed text by verbal control. Required equipment includes a TRS-80 Model I or III, a cassette recorder, and a printer. You don't have to be able to utter words; controlled verbal utterances or the tapping of a pencil upon the table will suffice.

A special feature of this software is the speed-control function. You can move the cursor (the heart of the operation) faster or slower by verbal command. You set your own pace.

To start the program, load Voice-controlled Typewriter, type Run, and press enter. The program begins and indicates that the auxiliary microphone and plugs, as well as any tape, should be removed from the tape recorder. Next press the record and play keys while simultaneously pushing in the silver interlock button in the left rear of the recording well. Press any key and the program will operate, under sound control.

A cursor moves across the top of the screen and when it is over a row with a desired letter, utter a short sound and it will travel down that row. When the cursor is at the left of the desired letter, utter another sound and that letter will

*The Key Box*

**Model I or III**
**16K, 32K RAM**
**Cassette or Disk Basic**
**Printer**

*Program Listing*

```
1 GOTO 622
2 CLEAR 400
3 B=0
4 DIMA$(70)
5 Q1=30
6 Q=50
7 FOR A=0 TO 69
8 A$(A)=" "
9 NEXT A
10 CLS
11 PRINT@5,"*":PRINT@13,"*":PRINT@21,"*":PRINT@29,"*":PRINT@37,"
*":PRINT@45,"*":PRINT@53,"*"
12 PRINT@133,"A":PRINT@141,"D":PRINT@149,"C":PRINT@157,"B":PRINT
@165,"SPACE":PRINT@173,"0":PRINT@181,"5"
13 PRINT@197,"E":PRINT@205,"H":PRINT@213,"F":PRINT@221,"J":PRINT
@229,".":PRINT@237,"1":PRINT@245,"6"
14 PRINT@261,"I":PRINT@269,"N":PRINT@277,"G":PRINT@285,"K":PRINT
@293,",":PRINT@301,"2":PRINT@309,"7"
15 PRINT@325,"O":PRINT@333,"S":PRINT@341,"L":PRINT@349,"M":PRINT
@357,"PRINT":PRINT@365,"3":PRINT@373,"8"
16 PRINT@389,"U":PRINT@397,"T":PRINT@405,"P":PRINT@413,"Q":PRINT
@421,"ERASE":PRINT@429,"4":PRINT@437,"9"
17 PRINT@469,"R":PRINT@477,"V":PRINT@485,"X":PRINT@501,"SPEED"
18 PRINT@533,"Y":PRINT@541,"W":PRINT@549,"Z"
19 SET(106,41):SET(107,41):SET(108,41):SET(109,41):SET(110,41):S
ET(111,41):SET(112,41):SET(113,41):SET(114,41):SET(115,41):SET(1
16,41):SET(117,41):SET(118,41):SET(119,41):SET(120,41):SET(121,4
1):SET(127,41):SET(126,41)
20 POKE 15364,191
21 POKE15412,128
22 FOR X=1 TO Q
23 OUT255,00
24 Y=INP(255)
25 IF Y>200THEN 70
26 NEXTX
27 POKE 15372,191
28 POKE15364,128
29 FOR X=1 TO Q
30 OUT255,00
31 Y=INP(255)
32 IF Y>200 THEN 127
33 NEXT X
34 POKE15380,191
35 POKE 15372,128
36 FOR X=1 TO Q
37 OUT255,00
38 Y=INP(255)
39 IF Y>200 THEN  184
40 NEXT X
41 POKE 15388,191
42 POKE15380,128
43 FOR X=1 TO Q
44 OUT 255,00
45  Y=INP(255)
46  IF Y>200 THEN 263
47 NEXT X
```

Listing continues

# MAILING LIST SYSTEM $119.95

**FOR TRS-80 (Tandy Trademark) Model I and III**

## WHAT SETS OUR SYSTEM APART?...

- Our system is configured specifically for large mailing lists (or small) on **floppy disk drives.** Some other major systems **run** on floppies but are really intended for use on hard disk drives. Such a system assumes that you have vast amounts of on line disk storage capacity - the continuity of the data is limited to what you can have on line at one time. To get the real benefit of such a system, one usually has to purchase expanded track/density floppy disk drives and even then the problem occurs when all the drives are filled with data. We have neatly solved this problem by allowing all your data disks to be maintained in continuous order - even though, due to the limitations of your drives, the list may be too large to all be "on line" at one time. Thus our system accomodates extremely large lists using your existing drives and yet avoids the "segmented" data problems of the hard disk approach.
- While it is fashionable to advertise all-machine-code systems, our system is primarily written in BASIC - with embedded machine code for the speed sensitive areas. What this means is that our system is **easy** to **modify**, yet extremely fast. This is very important since many users like to have custom modifications made (either by them or us) so as to fit some unique requirement. Our manual has a section devoted exclusively to such modifications. Remember all-machine-code systems are extremely difficult to modify.
- Continuity of the ordered data (even data spanning many disks) is not limited to a "session", but is permanent.
- Optional "backing up" of your data as-you-go is an integral part of the system and is not restricted to the end of a session. This is true even for deletions.
- The length of our data fields are more than adequate to accomodate even your longest names/addresses.
- Adjusts to a 32K memory although full use is made of a 48K memory. Can be used with any DOS including TRSDOS.
- The program disk does not have to stay on line, thus freeing more space for data storage.
- Load and scroll through entire entries or selected fields. Edit as you scroll or go directly (takes about 2 sec.) to a specific entry and edit or delete.
- Our automatic repeat feature allows often used names/addresses to be entered with a single key stroke.
- Each disk entry optionally "remembers" how many mailings have been made for that particular entry. Can be tied in with purge/select.
- Continuing **expert** support just a phone call away. You will be able to discuss your problems/modifications with the authors.
- **Money back guarantee if not fully satisfied.**

## ADDITIONAL FEATURES:

- Simple to use, even for the novice - menu oriented.
- Permits **2260** names "on-line" with 40 track double density drives and almost **5000** names with 80 track drives. The older 35 single density drives permit 1025 on-line entries - made possible with our unique data compression techniques.
- Super fast sort by alph. or zip order (8 sec. for 1000 entries). Both orders can exist **simultaneously** on disk.
- High speed recovery of entries from disk - speed of sort is meaningless if retreival from disk is slow. Ours pulls in over 11 per sec!
- Master list printout of your list in several formats (not just a rehash of labels) - extremely useful.
- Zip order is "sub-alphabetized".
- Less than 5 digit zips have leading 0's appended.
- Supports 9 digit zips, **Canadian zips**, and foreign abbrev.
- Optional second address line.
- Optional reversal of names about commas. This permits disk storage in last-name-first order to facilitate meaningful ordering while the printout will be in "natural" order.
- Permits telephone, account, and/or serial numbers, etc.
- Prints on envelopes or labels 1, 2, 3 or 4 across.
- Can print individual labels at time of creation or editing.
- Test label/envelope printing allows you to make vertical and horizontal adjustments with ease.
- Transfers old files to our system ← **LOOK!**
- Selective printout by specific zips or zip ranges.
- Plenty of user defined fields with provisions for simultaneously purging and/or selecting the printout - even allows for inequalities. Powerful and easy to use.
- Editing is simple and fast - direct access or automatic search. Batch transfer or edited entries to backup disks.
- Optionally provides for duplicate labels.
- Deleted entries have "holes" on disk filled automatically and alph. order is still maintained!
- All labels optionally support an "Attn:" line with provisions for multiple entries. This permits mail to be sent to several people at given addresses - conserves disk space.
- All Ø's are replaced by easier to read 0's in addresses.
- Continuous display of number of labels/envelopes printed.
- Extensive use of error traps - even recovers from a power failure during a printout.
- Extensive assortment of extra cost options for customized master list printout (in addition to the standard one mentioned above), transfer of entries between disks, summary reports, and "publisher's" type multiple list label printouts.
- Hardware requirements - 32K, printer, and 1 or 2 drives.

## FORM LETTER **(Use with Mail List System)** $39.95

Create letters and store on disk with provisions for later retreival and additions. Then print your letters using your mailing list.

- Same select and purge features as mailing list system.
- Select either continuous fanfold or "cut sheet" paper.
- Selectable tabing, test printing, and paging.
- Allows regular or legal size pages.
- Greetings are selectable by codes on mailing list. Options include Mr., Mrs., First/Last Name, global, or user defined.

## SUPER CALENDAR **(Supplied on tape only)** $19.95

Prints out calendars of individual months of years ranging from 1583 to any time in the future. Standard banker's holidays are noted. Additionally prints out large "graphics" type wall calendars with memos under each day. Use as a planning calendar with optional disk storage. Requires 16K and a printer.

## Loan Amortization **(Supplied on tape only)** $29.95

Achieves pin point accuracy with a built in calendar. This sophisticated program produces an exceptionally professional looking printout that includes yearly summaries as well as "totals-to-date". Several options for calculating interest including one that pushes the payment date ahead to the next business day if the regular pay date falls on a weekend or holiday.

## FOOTBALL SCOUT **(disk only)** $89.95

Charge local schools up to $1000 per season for these sophisticated reports. Analyze the tendancies of opposing teams. Equivalent to that used by the pros.

☐ **Mailing List System** ☐ **Loan**
☐ **Form Letter** ☐ **Football Scout** ☐ **Super Calendars**

Model I or III? ____________
Total (Add $3.00 for Shipping & Handling) ____________
Check ☐ C.O.D. ☐ Visa ☐ MC ☐
Card No. ____________ Exp. ________
Name ____________
Address ____________
City, State, & Zip ____________

**PRECISION PROTOTYPES** ✓ 63
**410 E. Roca Refugio, TX. 78377 (512)526-4758**

*Listing continued*

```
48 POKE 15396,191
49 POKE15388,128
50 FOR X=1 TO Q
51 OUT255,00
52  Y=INP(255)
53 IF Y>200 THEN 342
54 NEXT X
55 POKE 15404,191
56 POKE15396,128
57 FOR X=1 TO Q
58 OUT 255,00
59 Y=INP(255)
60 IF Y>200 THEN 419
61 NEXT X
62 POKE 15412,191
63 POKE 15404,128
64 FOR X=1 TO Q
65 OUT255,00
66 Y=INP(255)
67 IF Y>200 THEN 476
68 NEXT X
69 GOTO 20
70 POKE 15364, 128
71 POKE 15492,191
72 FOR X=1 TO Q1
73 NEXT X
74 B$="A"
75 C=15492
76 FOR X=1 TO Q
77 OUT255,00
78 Y=INP(255)
79 IF Y>200 THEN 543
80 NEXT X
81 POKE 15556,191
82 POKE 15492,128
83 FOR X=1 TO Q1
84 NEXT X
85 B$="E"
86 C=15556
87 FOR X=1 TO Q
88 OUT255,00
89 Y=INP(255)
90 IF Y>200 THEN 543
91 NEXT X
92 POKE 15620,191
93 POKE 15556,128
94 FOR X=1 TO Q1
95 NEXT X
96 B$="I"
97 C=15620
98 FOR X=1 TO Q
99 OUT255,00
100 Y=INP(255)
101 IF Y>200 THEN 543
102 NEXT X
103 POKE 15684,191
104 POKE 15620,128
105 FOR X=1 TO Q1
106 NEXT X
107 B$="O"
108 C=15684
109 FOR  X=1 TO Q
110 OUT255,00
111 Y=INP(255)
112 IF Y>200 THEN 543
113 NEXT X
114 POKE 15748, 191
115 POKE 15684,128
116 FOR X=1 TO Q1
117 NEXT X
118 B$="U"
119 C=15748
120 FOR X=1 TO Q
121 OUT255,00
122 Y=INP(255)
123 IF Y>200 THEN 543
124 NEXT X
125 POKE 15748,128
126 GOTO 20
127 POKE15372,128
128 POKE15500,191
129 FOR X=1 TO Q1
130 NEXT X
131 B$="D"
132 C=15500
133 FOR X=1 TO Q
134 OUT255,00
135 Y=INP(255)
136 IF Y>200 THEN 543
137 NEXT X
138 POKE  15564,191
139 POKE  15500,128
140 FOR X=1 TO Q1
141 NEXT X
142 B$="H"
143 C=15564
144 FOR X=1 TO Q
145 OUT255,00
146 Y=INP(255)
147 IF Y>200 THEN 543
148 NEXT  X
149 POKE 15628,191
150 POKE 15564,128
151 FOR X=1 TO Q1
152 NEXT X
153 B$="N"
154 C=15628
155 FOR X=1 TO Q
156 OUT255,00
157 Y=INP(255)
158 IF Y>200 THEN 543
159 NEXT X
160 POKE 15692,191
161 POKE 15628,128
162 FOR X=1 TO Q1
163 NEXT X
164 B$="S"
165 C=15692
166 FOR X=1 TO Q
167 OUT255,00
168 Y=INP(255)
169 IF Y>200 THEN 543
170 NEXT X
171 POKE 15756,191
172 POKE 15692,128
173 FOR X=1 TO Q1
174 NEXT X
175 B$="T"
176 C=15756
177 FOR X=1 TO Q
178 OUT255,00
179 Y=INP(255)
180 IF Y>200 THEN 543
181 NEXT X
182 POKE 15756,128
183 GOTO 20
184 POKE 15380,128
185 POKE 15508,191
186 FOR X=1 TO Q1
187 NEXT X
188 B$="C"
189 C=15508
190 FOR  X=1 TO Q
191 OUT255,00
192 Y=INP(255)
193 IF Y>200 THEN 543
194 NEXT X
195 POKE 15572,191
196 POKE 15508,128
197 FOR X=1 TO Q1
198 NEXT X
199 B$="F"
200 C=15572
201 FOR X=1 TO Q
202 OUT255,00
203 Y=INP(255)
204 IF Y>200 THEN 543
205 NEXT X
206 POKE  15636,191
207 POKE 15572,128
208 FOR X=1 TO Q1
209 NEXT X
210 B$="G"
211 C=15636
212 FOR X=1 TO Q
213 OUT255,00
214 Y=INP(255)
215 IF Y>200 THEN 543
216 NEXT X
217 POKE 15700,191
218 POKE 15636,128
219 FOR X=1 TO Q1
220 NEXT X
221 B$="L"
222 C=15700
223 FOR  X=1 TO Q
224 OUT255,00
225 Y=INP(255)
226 IF Y>200 THEN 543
227 NEXT X
228 POKE 15764,191
229 POKE 15700,128
230 FOR X=1 TO Q1
231 NEXT X
232 B$="P"
233 C=15764
234 FOR X=1 TO Q
235 OUT255,00
236 Y=INP(255)
237 IF Y>200 THEN 543
238 NEXT X
239 POKE 15828,191
240 POKE 15764,128
241 FOR X=1 TO Q1
242 NEXT X
243 B$="R"
244 C=15828
245 FOR X=1 TO Q
246 OUT255,00
247 Y=INP(255)
248 IF Y>200 THEN 543
249 NEXT X
250 POKE 15892,191
251 POKE  15828,128
252 FOR X=1 TO Q1
253 NEXT X
254 B$="Y"
255 C=15892
256 FOR X=1 TO Q
257 OUT255,00
258 Y=INP(255)
259 IF Y>200 THEN 543
260 NEXT X
261 POKE 15892,128
262 GOTO 20
263 POKE 15388,128
264 POKE 15516,191
265 FOR X=1 TO Q1
266 NEXT X
267 B$="B"
268 C=15516
269 FOR X=1 TO Q
270 OUT255,00
271 Y=INP(255)
272 IF Y>200 THEN 543
273 NEXT X
274 POKE 15580,191
275 POKE 15516,128
276 FOR X=1 TO Q1
277 NEXT X
278 B$="J"
279 C=15580
280 FOR X=1 TO Q
281 OUT255,00
282 Y=INP(255)
283 IF Y>200 THEN 543
284 NEXT X
285 POKE 15644,191
286 POKE 15580,128
287 FOR  X=1 TO Q1
288 NEXT X
289 B$="K"
290 C=15644
291 FOR X=1 TO Q
292 OUT255,00
293 Y=INP(255)
294 IF Y>200 THEN 543
295 NEXT X
296 POKE 15708,191
297 POKE 15644,128
298 FOR X=1 TO Q1
299 NEXT X
300 B$="M"
301 C=15708
302 FOR X=1 TO Q
303 OUT255,00
304 Y=INP(255)
305 IF Y>200 THEN 543
306 NEXT X
307 POKE 15772,191
308 POKE 15708,128
309 FOR X=1 TO Q1
310 NEXT X
311 B$="Q"
312 C=15772
313 FOR X=1TOQ
314 OUT255,00
315 Y=INP(255)
316 IF Y>200 THEN 543
317 NEXT X
318 POKE 15836,191
319 POKE 15772,128
320 FOR X=1 TO Q1
321 NEXT X
322 B$="Q"
323 C=15836
324 FOR X=1 TO Q
325 OUT255,00
326 Y=INP(255)
327 IF Y>200 THEN 543
328 NEXT X
329 POKE 15900,191
330 POKE 15836,128
331 FOR X=1 TO Q1
332 NEXT X
333 B$="W"
334 C=15900
335 FOR X=1 TO Q
336 OUT255,00
337 Y=INP(255)
338 IF Y>200 THEN 543
339 NEXT X
340 POKE 15900,128
341 GOTO 20
342 POKE 15524,191
343 POKE 15396,128
344 FOR X=1 TO Q1
345 NEXT X
346 B$=CHR$(176)
347 C=15524
348 FOR X=1 TO Q
349 OUT255,00
350 Y=INP(255)
351 IF Y>200 THEN 543
352 NEXT X
353 POKE 15588,191
354 POKE 15524,128
355 FOR X=1 TO Q1
356 NEXT X
357 B$="."
358 C=15588
359 FOR X=1 TO Q
360 OUT255,00
361 Y=INP(255)
362 IF Y>200 THEN 543
363 NEXT X
364 POKE 15652,191
365 POKE 15588,128
366 FOR X=1 TO Q1
367 NEXT X
368 B$=","
369 C=15652
370 FOR X=1 TO Q
371 OUT255,00
372 Y=INP(255)
373 IF Y>200 THEN 543
374 NEXT X
375 POKE 15716,191
376 POKE 15652,128
377 FOR X=1 TO Q1
378 NEXT X
379 C=15716
380 FOR X=1 TO Q
381 OUT255,00
382 Y=INP(255)
383 IF Y>200 THEN 553
384 NEXT X
385 POKE 15780,191
386 POKE 15716,128
387 FOR X=1 TO Q1
388 NEXT X
389 C=15780
```

*Listing continues*

✓See List of Advertisers on Page 451

*Listing continued*

```
390 FOR X=1 TO Q
391 OUT255,00
392 Y=INP(255)
393 IF Y>200 THEN 568
394 NEXT X
395 POKE 15844,191
396 POKE 15780,128
397 FOR X=1 TO Q1
398 NEXT X
399 B$="X"
400 C=15844
401 FOR X=1 TO Q
402 OUT255,00
403 Y=INP(255)
404 IF Y>200 THEN 543
405 NEXT X
406 POKE 15908,191
407 POKE 15844,128
408 FOR X=1 TO Q1
409 NEXT X
410 B$="Z"
411 C=15908
412 FOR X= 1 TO Q
413 OUT255,00
414 Y=INP(255)
415 IF Y>200 THEN 543
416 NEXT X
417 POKE 15908,128
418 GOTO 20
419 POKE  15532,191
420 POKE15404,128
421 FOR X=1 TO Q1
422 NEXT X
423 B$="0"
424 C=15532
425 FOR X=1 TO Q
426 OUT255,00
427 Y=INP(255)
428 IF Y>200 THEN 543
429 NEXT X
430 POKE 15596,191
431 POKE 15532,128
432 FOR X=1 TO Q1
433 NEXT X
434 B$="1"
435 C=15596
436 FOR X=1 TO Q
437 OUT255,00
438 Y=INP(255)
439 IF Y>200 THEN 543
440 NEXT X
441 POKE 15660,191
442 POKE 15596,128
443 FOR X=1 TO Q1
444 NEXT X
445 B$="2"
446 C=15660
447 FOR X=1  TO  Q
448 OUT255,00
449 Y=INP(255)
450 IF Y>200 THEN 543
451 NEXT X
452 POKE 15724,191
453 POKE 15660,128
454 FOR X=1 TO Q1
455 NEXT X
456 B$="3"
457 C=15724
458 FOR X=1 TO Q
459 OUT255,00
460 Y=INP(255)
461 IF Y>200 THEN 543
462 NEXT X
463 POKE 15788,191
464 POKE 15724,128
465 FOR X=1 TO Q1
466 NEXT X
467 B$="4"
468 C=15788
469 FOR X=1 TO Q
470 OUT255,00
471 Y=INP(255)
472 IF Y>200 THEN 543
473 NEXT X
474 POKE 15788,128
475 GOTO 20
476 POKE 15540,191
477 POKE 15412,128
478 FOR X=1 TO Q1
479 NEXT X
480 B$="5"
481 C=15540
482 FOR X=1 TO Q
483 OUT255,00
484 Y=INP(255)
485 IF Y>200 THEN 543
486 NEXT X
487 POKE 15604,191
488 POKE 15540,128
489 FOR X=1 TO Q1
490 NEXT X
491 B$="6"
492 C=15604
493 FOR X=1 TO Q
494 OUT255,00
495 Y=INP(255)
496 IF Y>200 THEN 543
497 NEXT X
498 POKE 15668,191
499 POKE 15604,128
500 FOR X=1 TO Q1
501 NEXT X
502 B$="7"
503 C=15668
504 FOR X=1 TO Q
505 OUT255,00
506 Y=INP(255)
507 IF Y>200 THEN 543
508 NEXT X
509 POKE 15732,191
510 POKE 15668,128
511 FOR X=1 TO Q1
512 NEXT X
513 B$="8"
514 C=15732
515 FOR X=1 TO Q
516 OUT255,00
517 Y=INP(255)
518 IF Y>200 THEN 543
519 NEXT X
520 POKE 15796,191
521 POKE 15732,128
522 FOR X=1 TO Q1
523 NEXT X
524 B$="9"
525 C=15796
526 FOR X=1 TO Q
527 OUT255,00
528 Y=INP(255)
529 IF Y>200 THEN 543
530 NEXT X
531 POKE 15860,191
532 POKE 15796,128
533 FOR X=1 TO Q1
534 NEXT X
535 C=15860
536 FOR X=1 TO Q
537 OUT255,00
538 Y=INP(255)
539 IF Y>200 THEN 573
540 NEXT X
541 POKE 15860,128
542 GOTO 20
543 GOTO544
544 POKEC,128
545 PRINT@(896+B),B$
546 IF B$=CHR$(176) THEN B$=" "
547 A$(B)=B$
548 B=B+1
549 IF B=62 THEN 553
550 FOR X=1 TO 20
551 NEXT X
552 GOTO 20
553 C$=A$(0)+A$(1)+A$(2)+A$(3)+A$(4)+A$(5)+A$(6)+A$(7)+A$(8)+A$(
9)+A$(10)+A$(11)+A$(12)+A$(13)+A$(14)+A$(15)+A$(16)+A$(17)+A$(18
)+A$(19)
554 D$=A$(20)+A$(21)+A$(22)+A$(23)+A$(24)+A$(25)+A$(26)+A$(27)+A
$(28)+A$(29)+A$(30)+A$(31)+A$(32)+A$(33)+A$(34)+A$(35)+A$(36)+A$
(37)+A$(38)+A$(39)
555 E$=A$(40)+A$(41)+A$(42)+A$(43)+A$(44)+A$(45)+A$(46)+A$(47)+A
$(48)+A$(49)+A$(50)+A$(51)+A$(52)+A$(53)+A$(54)+A$(55)+A$(56)+A$
```

*Listing continues*

*Listing continued*

```
(57)+A$(58)+A$(59)
556 POKE15716,128
557 F$=A$(60)+A$(61)+A$(62)
558 G$=C$+D$+E$+F$
559 LPRINTG$
560 PRINT@896,"
                "
561 FOR A=0TO69
562 A$(A)=" "
563 NEXT A
564 B=0
565 FOR A=0 TO 500
566 NEXT A
567 GOTO 20
568 B=(B-1):IF B<0 THEN B=0:A$(B)=" "
569 POKE 15780,128
570 PRINT@(896+B)," "
571 GOTO 20
572 GOTO 628
573 CLS
574 PRINT@20,"TIME CONSTANT"
575 PRINT@35,Q
576 PRINT@133,"FASTER"
577 PRINT@153,"SLOWER"
578 PRINT@173,"RETURN"
579 POKE15492,191
580 POKE15532,128
581 FOR X=1TO20
582 NEXT X
583 FOR X=1 TO 50
584 OUT255,00
585 Y=INP(255)
586 IF Y>200 THEN 607
587 NEXT X
588 POKE 15512,191
589 POKE 15492,128
590 FOR X=1 TO 20
591 NEXT X
592 FOR X=1TO50
593 OUT255,00
594 Y=INP(255)
595 IF Y>200 THEN 614
596 NEXT X
597 POKE 15532,191
598 POKE 15512,128
599 FOR X=1 TO 20
600 NEXT X
601 FOR X=1 TO 50
602 OUT255,00
603 Y=INP(255)
604 IF Y>200 THEN 620
605 NEXT X
606 GOTO579
607 Q=(Q-5)
608 IF Q=0 THEN Q=5
609 POKE15492,128
610 PRINT@35,Q
611 FOR X=1 TO 50
612 NEXT X
613 GOTO 579
614 Q=(Q+5)
615 POKE 15512,128
616 PRINT@35,Q
617 FOR X=1 TO 50
618 NEXT X
619 GOTO 579
620 CLS
621 B=0:GOTO7
622 CLS
623 PRINT"1.   YOUR PRINTER SHOULD BE ATTACHED AND ON."
624 PRINT"2.   REMOVE THE 'AUX' AND 'MIC' PLUGS FROM THE CASSETT
E        RECORDER."
625 PRINT"3.   THE ROOM MUST BE QUIET AND THE RECORDER ON A HARD
 FLAT     SURFACE FOR BEST RESULTS."
626 PRINT"4.   DEPRESS 'RECORD', 'PLAY', AND THE SMALL PIN AS IN
DICATED    IN THE FLASHING DRAWING."
627 PRINT"5.   DEPRESS ANY KEY TO CONTINUE."
628 POKE15916,151:POKE15917,135:POKE15918,131:POKE15919,131:POKE
15920,131:POKE15921,131:POKE15922,131:POKE15923,171:POKE15980,14
9:POKE15987,170
629 POKE16044,149:POKE16045,191:POKE16047,191:POKE16051,170
630 M$=INKEY$
631 IF M$=""THEN633
632 GOTO2
633 POKE15917,131:POKE16045,128:POKE16047,128
634 GOTO 628
```

appear at the bottom of the video screen. The cursor returns to the home position in the upper left and circulates again.

The Space function makes a small block appear on the video display (the printer will print a space). The Print function prints the video display line, and Erase deletes the last character (or space) from the video screen. If you reach the end of the video line (the gap in the indicating bar above the line), the line is automatically printed.

The Speed function produces a new display that has the commands Faster, Slower, and Return. These commands increase or decrease the speed at which the cursor moves. Return sends the program back to the original mode. Anything on the video display line will be lost upon exiting to the Speed control display. You can change the speed as often as you wish.

This program is written in Basic and some delay is allowed after a Print command so the printer noise will not be "heard" by the tape recorder. If your printer is slow, you may want to increase this delay by changing the 500 in line 565 to a greater number, such as 2000. ■



✓See List of Advertisers on Page 451

# TWO MILLION PEOPLE COULD USE YOUR SOFTWARE.

# TOO BAD THEY'VE NEVER HEARD OF YOU.

At last count, more than a couple of million Americans owned personal computers. And that number is expected to triple by 1985.

That's a fertile environment for software sales. But a lot of personal computer software organizations are discovering it's not as easy as they might have thought.

The problems boil down to these:

Lack of awareness on the part of personal computer buyers. Lack of understanding on the part of personal computer salespeople. Lack of unlimited dollars to advertise in the myriad magazines that reach this small segment or that small segment of the total market.

If any of this sounds like a familiar frustration to you, we've got good news. *LIST* is here. And its advent heralds a new era in cost-effective software marketing.

*LIST* isn't another guide. It's not another directory. It's an informative new publication that puts software first. And puts you in touch—directly, inexpensively—with the fastest growing segments of the personal computer market.

With *LIST*, you'll be able to reach business and professional people at a critical point in the purchase cycle—*before* they've bought their hardware.

At the same time, you'll be able to impact another significant market segment—those people who already own personal computers, and are eager to learn how they can do more with them.

To find out how *LIST* can expand the awareness of your software for less than $200, send us this coupon. (Hurry! We go to press December 15.)

Or don't. And go on being less well known than you deserve to be.

*LIST is published by Redgate Publishing Co., an affiliate of E F Hutton & Co.*

**1 800 327-1300**

In Florida call: 1 305 231-6904

**"I WANT TO MAKE MORE PEOPLE AWARE OF MY SOFTWARE."**


☐ Please send me literature that explains how *LIST* can help.

☐ I don't want to wait on the mails. Call me right away.

Send to *LIST*, Redgate Publishing Co., 3407 Ocean Drive, Vero Beach, FL 32960. **Or phone 1 800 327-1300.**

In Florida call: 1 305 231-6904

NAME TITLE

COMPANY TELEPHONE

ADDRESS

CITY STATE ZIP

MI

**LIST**™

The Software Resource Book For Personal Computer Users

# The Color Computer Goes Forth

by Scott L. Norman

★ ★ ★

**Colorforth**
**Armadillo International Software**
**Austin, TX 78712**
**Color Computer, 16K**
**$49.95**

★ ★ ★ ★ ½

***Starting Forth***
**Leo Brodie**
**Prentice-Hall**
**Englewood Cliffs, NJ 07632**
**softcover, 348 pp.**
**$15.95**

**by Scott L. Norman**

Colorforth is an implementation for the Color Computer's 6809 microprocessor of FIG-Forth, the standard promoted by the Forth Interest Group (FIG) of San Carlos, CA. The documentation is a 12-page Armadillo Software pamphlet and a detailed FIG glossary. The pamphlet contains details on changing the memory maps for disk and cassette systems, as well as a few special features of Colorforth. The FIG publication includes the manual for the source-code text editor.

### *Starting Forth*

There's no way a stranger to Forth could learn the language from this material. Armadillo Software suggests you purchase Leo Brodie's book. There are tutorial reviews, problems with solutions, and carefully marked sections for novices and relative experts.

**Programming in Forth is now possible for Color Computer owners; read on and find out how**

The pages of *Starting Forth* are inhabited by marvelous cartoon characters who illustrate the functions of Forth words and features; there's a robed monk with a quill pen for the compiler, a hooded medieval executioner for the word execute, and a slick-haired, mustachioed character who constantly refers to his dictionary. Naturally, he's the interpreter. My favorite is the two-headed flying reptile responsible for exchanging the top two numbers on the parameter stack. By an amazing coincidence, the word he represents matches the sound of his wings: swap!

None of this hides the fact that Brodie has produced a serious book. It is best read at the keyboard where you can test the examples interactively, since certain sections are geared to Forth Inc. products and discuss features not relevant to Colorforth. An example is the discussion of the text editor, for which Colorforth uses very different syntax. The FIG notes provide a clear guide to this aspect of the software. In most respects, *Starting Forth* is excellent in both style and content, and I recommend it.

### Colorforth

The Colorforth software comes on cassette, with versions for 16K systems with or without disk drives. There are complete instructions for making cassette back-ups and for modifying and storing disk and cassette versions for 32K systems.

There is also a discussion of the simulated disk screens used by cassette-based systems. Forth uses virtual storage, shuffling 1,024-character screens between disk and RAM as needed. Colorforth reserves RAM for the purpose of simulating such storage units: four of them in 16K, 12 in 32K. (A real disk holds about 135K.)

By using the starting and finishing addresses specified in the documentation, you can store the simulated screens' contents on tape so your newly defined text isn't lost at the end of a terminal session. You can also define new words without using a screen, although you can't use the text editor to correct errors. Words that compile correctly become part of Colorforth's dictionary, at least for the duration of the session. The Colorforth pamphlet also gives the addresses used in saving a dictionary that has been extended.

I added the word CLS to the dictionary, to clear the screen as in Color Basic. Colorforth doesn't have a built-in function like this. My word was constructed as a colon definition; i.e., the defining name, operands, and previously defined (stock) words out of which it was constructed were enclosed between a leading colon and trailing semicolon, like this: : CLS 1024 512 143 FILL ;. Hitting enter compiles the word into the dictionary.

This illustrates a number of qualities about typical Forth words, so let's take it apart slowly. It starts with the name I choose, which can be up to 31 characters long. The use of long names makes programs somewhat self-documenting, but remember that what's stored is just the first three characters plus a count of the word's length. This means that WORD1 and WORD2 are indistinguishable as names; both are five characters long and both begin with WOR. It is acceptable to use 1WORD and 2WORD, however.

The definition follows the name of CLS; in this case, "Starting at address 1024, fill 512 memory locations with the byte pattern that translates as decimal number 143." (The definition of the word Fill tells us which operand corresponds to each of these parameters.)

Fill is a general Forth word, and can perform this memory-stuffing operation anywhere in RAM. Recall, however, that in the Color Computer location 1024 is the start of the low-resolution video screen. When you store numbers in such locations, the corresponding graphics character shows up on the video display, so that my Colorforth word is the equivalent of the Color Basic loop:

```
100 FOR I = 1024 TO 1024 + 511
100 POKE I, 143
120 NEXT I
```

Referencing Color Computer documentation verifies that here, 143 is a solid green graphics block.

You can see the extent Colorforth represents a customization of Forth to the Color Computer. You can capitalize on the memory-mapped video of the 6809, and use the predefined graphics characters at will.

For example, here is one way to generalize CLS to a new word, YELSCRN, which clears the video display to solid yellow instead of solid green: : YELSCRN 1024 512 159 FILL KEY ;. Here, 159 represents the solid yellow graphics block denoted as CHR$(159) in Color Basic. Another stock Forth word, Key, has been introduced. It causes the computer to wait until a key is depressed before continuing (before signifying the complete execution of YELSCRN). The signal takes the form of OK printed on a green field, which spoils the appearance of the solid yellow screen. Thus Key is similar in function to the Basic statement 200 A$ = INKEY$ : IF A$ = "" THEN 200, or to the indefinite loop 200 GOTO 200, often used to freeze a video display.

Once they are defined, invoke the new words by calling them; to clear the screen to green at any time, enter CLS from the keyboard or include it in a program statement.

**Text Screen Operations**

You can define a whole series of colored backgrounds if desired. The best way is to use the text editor, entering a whole group of definitions on a screen. Remember, words defined this way don't have to be sequential statements in any one program. When using a disk, prepare the system for screen input by entering "empty-buffers," followed by "disk." The command Editor invokes the editor; screen # 1 can then be loaded with blanks and selected for editing by 1 CLEAR.

A text screen consists of 16 lines of a maximum of 64 characters each. The lines are numbered 0–15, and you enter text on line #n by prefixing your material with n P. Thus typing the following two lines loads the defining words for blue and red backgrounds into virtual memory:

```
2 P : BLUSCRN 1024 512 175 FILL KEY ;
3 P : REDSCRN 1024 512 191 FILL KEY ;
```

Line numbers 2 and 3 refer only to locations being edited within the screen; there are no line numbers in a Colorforth program itself.

The last entry on a text screen should be ;S (note the lack of a space between semicolon and S), which stops interpretation when the screen is loaded. You need not devote an entire line to this two-character word; Colorforth can handle multiple-word (multiple-statement) lines, as long as they're less than 64 characters. If a single definition is longer than this, press enter when nearing the end of the first 64 characters and continue on the next screen line. A right square bracket, generated by pressing the shift and right arrow keys simultaneously, denotes the start of a continuation line.

The FIG documentation discusses

# What Forth Is, and Isn't

I knew very little about Forth before being introduced to the Colorforth package, though I had a feeling that it was useful for computer control of machinery. The arrival of the Colorforth cassette motivated me to learn more about the background and capabilities of the language.

I learned that Forth—at least in its original form—was largely the invention of Charles H. Moore, who now heads Forth Inc. in CA. Forth has become a favorite for computer control of instruments among radio astronomers, although since its inception it has found a wider variety of applications.

It's compact, and has been implemented on a range of computers, from mainframes to micros. The microcomputer versions are not necessarily poor relations; a very capable Forth package can be written for a 16K machine.

One of the problems in writing about Forth is that, strictly speaking, it isn't a language, but rather a complete programming environment. A typical package includes a core dictionary of words, a compiler, an assembler, and a text editor for working on source code. For brevity, though, I'll call it a language in this review. It's easy to customize Forth for any particular application. Most versions, however, start with an unusual degree of commonality. The Forth community has shown a genuine interest in setting standards, which makes it easy to use general texts like Brodie's with other vendors' software packages.

But what is it like to program in Forth? You begin with a dictionary of 150 or so high-level words that act like subroutine calls. These can be added to. In fact, much of what you do when programming in Forth is to define more and more powerful words, combining prior definitions with embedded data. Ultimately, you define a word that, when called, executes the entire program.

Languages implemented in this fashion are said to be threaded; you can visualize a thread of subroutine addresses linking the program words together, eventually ending at the machine-language level where the most basic words (the ones that were in the dictionary to begin with) are defined. The commands a programmer generates can be tested immediately

Continues on p. 82

the text editor. It performs most functions you expect from an editor, although some operations aren't as convenient. For example, it takes two commands to change one text string into another: X oldtext, C newtext.

The word Forth causes the system to leave the editor. All that remains is to compile the words defined on the screen into the dictionary, done easily with the Load command.

**Dealing With Numbers**

All the examples so far have used decimal numbers; Colorforth initializes in decimal, just as Basic does. It does have a built-in hexadecimal capability, however, and it's easy to define the system you want and have Colorforth operate in any other base.

You can switch between decimal and hexadecimal with the words Decimal and Hex. These are toggled commands, so exercise a little caution if you change bases in the middle of a routine. Here's a definition for a new word, Convert, which takes a decimal number from the keyboard and prints its hex equivalent: : CONVERT CR HEX . DECIMAL CR ;. This uses two more predefined words: CR, which generates a carriage return; and ".", the period, which causes the preceding number to print according to the most recently defined base.

You can see how the liberal use of punctuation marks as words makes Forth hard to write about. Assume that you start out in the decimal system; you invoke the new word by entering n Convert, where n is an appropriate number. The entry process puts n on top of the stack. Hex converts the top number on the stack to hexadecimal, the period prints it, and Decimal returns you to base 10 for the next keyboard entry. The initial and final carriage returns provide a decent display.

From this example, you can see how the stack gets involved in Colorforth operations, including I/O.

The next step is the definition of other numbering systems. This involves the predefined variable BASE, which stores the decimal value of the number used for conversions. Here are the definitions of two new systems:

```
: TERNARY 3 BASE ! ;
: 25-ARY 25 BASE ! ;
```

These aren't especially useful, but are different from the Binary and Octal that everyone defines when writing about Forth.

These definitions introduce the ex-

Continued from p. 81

from the keyboard, since they are compiled as they are entered. Alternatively, you can use the text editor to write chunks of source code called blocks or screens. These can be stored on disk or tape, or even in RAM, and entered as a unit.

The Forth incremental compiler also compiles this code as it is entered. Once compiled, new Forth words have the same status as those that came with the original package; there's no difference between the original building blocks and any particular program statement. This lack of a fixed instruction set has caused purists to object to calling Forth a language.

The extendable dictionary is one of Forth's principal features. It takes a little getting used to the fact that the words are not as limited in form as they are in Basic. A Forth word can begin with (or even be) a number. There are even common words that are punctuation marks. It's a little bizarre, but let me get ahead of myself: This is a valid Forth statement, defining one standard word in terms of others: : ? @ . ;.

It's true that some of this shorthand makes Forth hard to read. There are a couple things that initially seem strange, at least if you've come through the Basic ranks, and it is worthwhile to mention them before going into the details of Colorforth itself.

The first is the use of a stack. This is a portion of RAM set aside for the sequential storage and retrieval of data in a last-in, first-out (LIFO) fashion. Forth uses the stack, or more properly the parameter stack, to pass operands and data from one word or subroutine to another.

When writing Forth code, it is convenient to keep intermediate results bouncing up and down on the stack from one operation to another. That doesn't mean that you can't name variables and call them as needed; but using the stack intelligently can speed program operation. You can't avoid keeping track of things on the stack anyway, since operands have to be presented to Forth words in the correct order.

Finally, it can be just plain fun to define your own commands and incorporate them into the dictionary; it's the easiest way I know to produce your own personalized computer language.

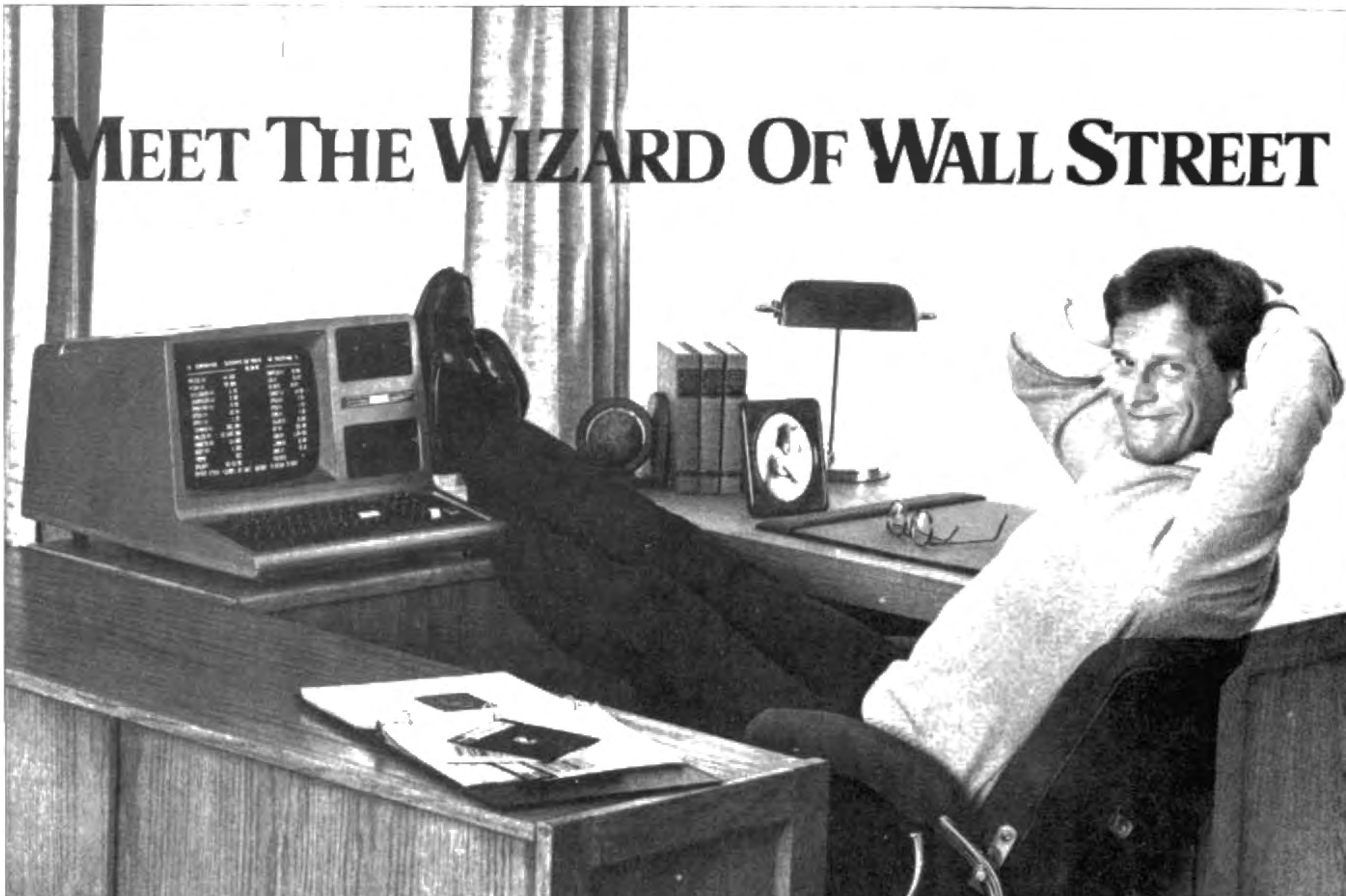
Forth uses a second (or return) stack to keep track of procedure call information. Programmers use the return stack to test the current value of the running index in a DO loop, Forth's equivalent of the For...Next loop.

Because of the parameter stack, Forth employs reverse Polish, or postfix, notation (RPN). In this system, arithmetic expressions are written with operations following the operands to which they apply, so that A + B * (C-D) / E becomes CD-B * E / A +. Operations are carried out immediately as the expression scans from left to right. No parentheses are needed, and there is no implied hierarchy of mathematical operations. Compilation or interpretation can be very fast.

Hewlett-Packard calculator users will recognize this, as their machines use RPN. In fact, Forth-like languages are used in the ROMs of some calculators, as well as in scientific instruments, machine-tool controllers, and so on. Still, before plunging into the Colorforth version, ask what Forth can do for a microcomputer owner, especially a Color Computer owner. After all, it requires a lot of attention to detail, the necessity to move numbers to and from the stack makes it less than ideal for number crunching, and its structure and syntax are radically different from those of Color Basic.

To some extent, this difference is responsible for some of Forth's appeal. It allows the programmer close control over the computer (for I/O operations, for example) that you associate with Assembly language, but it's easier to work with. It's fast, and it's easy to incorporate Assembly-language subroutines into a Forth program for the most time-critical applications.

# MEET THE WIZARD OF WALL STREET



## Standard & Poor's STOCKPAK system can turn any investor into a whiz

Whether you're an active investor, or just considering the best ways to invest, STOCKPAK can turn you into a Wall Street Wizard.

## Sophisticated software that's easy to use

STOCKPAK is a sophisticated software system that offers you an easy and economical way to obtain and evaluate important investment information to help you make better investment decisions. In fact, in virtually no time at all, you can put STOCKPAK to work for you performing a wide range of investment functions.

## Screening power is at your command

Our Screen and Select system lets you apply your own investment criteria to a 900-company database. For example, you can tell STOCKPAK to find only large companies, or only small ones...only those with high earnings growth, or low P/E's, or high dividends... only those in certain industries, or on specific exchanges...only those above or below a certain price... or only those with high sales, or low debt, and so on.


In fact, you can draw from over 200 specific information items to compose a single screen. And, you can establish your own targets and cut-off points in each of these 200 areas. STOCKPAK will oblige by searching through the entire database to find only those companies that satisfy your criteria.

## More feats of wizardry with STOCKPAK

Standard & Poor's time-tested software also lets you evaluate and manage actual or hypothetical portfolios of up to 100 stocks each. And, you can create your own customized reports—displayed the way you want them.

Whether you use STOCKPAK to help improve your investment decisions, or just for the fun of it, it's good to know the source of this wizardry: Standard & Poor's, one of the world's leading financial publishers with a 120-year reputation of service to the investment community.

## Visit your local Radio Shack Computer Store today

STOCKPAK is designed exclusively for use on the TRS-80 Models III and I computers with 32K business systems and two disk drives. The basic software and sample database plus complete documentation is available from your Radio Shack computer store for only $49.95. The STOCKPAK monthly updating service can be ordered directly from Standard & Poor's for only $200 annually. Visit your local Radio Shack today for additional information and a demonstration.

**STANDARD & POOR'S CORPORATION** ✓261

**25 BROADWAY, NEW YORK, N.Y. 10004**



✓ See List of Advertisers on Page 451

clamation point, which Forth uses to denote storing a number at a specified address. The syntax is such that you provide the address by specifying the variable BASE; variables return their addresses to the stack, while constants return their values.

Earlier, I referred to a number n as an "appropriate" number. What does this mean? Colorforth is set up to do fixed-point arithmetic only, using signed 16-bit numbers (32767 to -32768). This is a general property of Forth, even when used on 8-bit microprocessors. Some double-length operations are available, and it's possible to write floating-point routines, but this is one reason Colorforth wouldn't be the language of choice for scientific number crunching.

To enter a double-length number, you punctuate it with a period; you can enter 100000 as 100.000, for example. Colorforth also has formatting commands similar to Print Using for outputting large numbers.

Although the range of numbers usable for calculation is limited, some operators do interpret 16 bits as an unsigned number (0–65536). As you'll see, this is what you need to address memory-mapped I/O in the Color Computer.

**Comparison and Control Structures**

Colorforth has a fairly complete set of comparison operators, which are used to control program branching within If...Then...Else statements. Because of RPN, the syntax of these commands takes some getting used to, although the ideas are familiar enough from Basic. Here are Color Basic and Colorforth program fragments that accept a number from the keyboard, test it to see if it is positive or negative, and print the result:

Color Basic

```
100 INPUT N
110 IF N > 0 THEN PRINT "POSITIVE"
ELSE PRINT "NEGATIVE"
```

Colorforth

```
: TEST 0 > IF ." POSITIVE "
ELSE ." NEGATIVE " THEN ;
(Used by entering N TEST)
```

Notice that in Colorforth, the words If and Then span the possible branches the program might take, but not the decision-making portion. That comes before the If, in the form required by RPN; for example, (N > 0) becomes (N 0>), where, in my example, N is put on the stack from the keyboard. This example also introduces dot-quote, the word used to print literals. Neither example considers the possibility that N equals zero.

In the Basic example, control passes to the statement following line 110 after a message has been printed. Similarly, Colorforth passes to whatever follows Then. There is a difference, of course, in that the Basic If...Then could have altered program flow by using statement numbers (implied GOTOs), instead of the Print commands. Because Colorforth is a structured language without line numbers, there are no GOTO statements. You can use other words within the If...Then to execute actions other than printing, however. After they are executed, control eventually returns to the word following Then.

*"Colorforth has a fairly complete set of comparison operators, which are used to control program branching within If...Then...Else statements."*

Colorforth comparison expressions can be quite lengthy, as they are in Basic. Simple algebraic comparisons can be logically combined, using AND, OR, or XOR operators.

Color Basic has only one kind of repetitive structure, the definite For...Next loop where the index's lower and upper bounds are specified. Colorforth has a similar construct, called a DO loop, plus a pair of indefinite forms, the Begin...Until and Begin...While...Repeat loops. The DO loop cannot be executed interactively; it must be compiled into a word, although you can specify one or both limits from the keyboard when the word is executed. Typical syntax is: : TEST b a DO (expression) LOOP;, where the loop index will run from a to (b–1).

For this example, b must be greater than a, and the index is incremented by one for each pass through the loop; the more general form is: : NEWTEST b a DO (expression) c +LOOP ;, where c is the step size, which can be negative, and where the loop is terminated by the word +LOOP.

The current value of the loop index is always kept on the second (return) stack, from which it can be obtained by the Colorforth word I. Here's a simple example that loops 10 times, printing the number of each cycle as it goes along: : COUNT 11 1 DO CR I . LOOP ;.

Loops can also be nested. Here is a pair that together make a software timer: : TIMER 0 DO 15500 0 DO LOOP LOOP ;. To time out N seconds, enter N TIMER. N serves as the upper limit for the outer loop, while the inner loop does nothing 15,500 times per second. This gives you an idea of Colorforth's relative speed. Color Basic's For...Next timing loop only executes 460 times per second. Notice that each nested loop terminates on its own Loop, in contrast to Basic, where loops can share a Next statement.

The indefinite loops have no simple counterparts in Basic, although Pascal and other structured languages have similar constructs. These loops are used to repeat some action or calculation until a particular event occurs, where the programmer has no control over how long that might take. The general syntax for a Begin...Until loop is: BEGIN (expression to be repeated) f UNTIL, where f represents a logical flag. The main expression is repeated as long as the flag is zero (false). The loop ends when the flag takes on the value one (true). Mathematical comparisons such as N 0 > return flags of this sort to the top of the stack.

The inverse behavior, the repetition of an operation as long as something is true rather than until it is true, is obtained with the Begin...While...Repeat loop, where the logical test is performed part way through the cycle. The construction is: BEGIN (first part) f WHILE (second part) REPEAT. After the first part of the computation, the program tests a flag. If it is true, execution continues to the second part of the computation and the whole thing repeats. As soon as the flag tests false, execution exits from the loop.

**Constants, Variables, and Arrays**

Although Colorforth uses the stack for temporary storage, it also lets you name constants and variables for future reference. The naming phrases are: VARIABLE (name) and (value) CONSTANT (name). These phrases are not colon definitions, like those I used for CLS. It is not necessary to specify the functions of Variable and Constant; they're predefined words, and their machine code takes care of that.

Earlier, I said that using the name of a variable in another statement puts that variable's address on top of the stack. If you use the name of a constant, its value, not its address, is placed there.

Look again at the word YELSCRN. You can define the solid yellow graphics block this way: 159 CONSTANT YELBLOCK, so the word that sets the screen to a solid yellow becomes: : YELSCRN 1024 512 YELBLOCK FILL KEY;. This isn't very profound, but it's a necessity for high-level languages. Even Assembly language lets you use labels.

If using a variable's name returns its address, how do you get its value? For that matter, how do you set its value? As you might expect, Forth dialects have other words for storing and retrieving the variables' values. The bad news is that these particular words use punctuation marks, so they are difficult to write about clearly, and their forms don't necessarily suggest their definitions. Still, Color Basic programmers get used to a leading question mark for Print and an apostrophe for REM.

The word that stores a 16-bit number at any given address is an exclamation point. So, the definition of TERNARY, "3 BASE !" actually consists of the phrase, "Store the 16-bit representation of the decimal number 3 in the address referred to by the predefined variable BASE." There is another word, " + !", that adds a number to whatever is at a given address, similar to the add-to-memory key on a calculator.

The operation of retrieving the value from a particular address and putting it on the stack is performed by the at symbol (@), pronounced fetch. A useful extension, symbolized by the question mark, is fetch followed by a print command. The operand precedes this word; to find out what number system Colorforth is in at any time, enter DECIMAL BASE?. A little thought shows you why the word Decimal is needed. The definition of the word "?" is the cryptic example I gave in the section "What Forth Is."

These words are useful to Forth programs running on 16-bit machines, although they also exist in the Colorforth dictionary. There are analogous words, C! and C@, that store and fetch 8-bit values and are more interesting to Color Computer programmers. The C prefix stands for character, because these words are useful for manipulating ASCII characters (even on larger machines).

Here's a routine that exercises some of the concepts and Colorforth words I've introduced so far. It displays each of the Color Computer's 256 distinct printable characters, including graphics blocks, in turn at the middle of the screen (location 1295). Each character shows on the screen for two seconds.

```
1295 CONSTANT MIDSCRN
: CHARS CLS 256 0 DO I
MIDSCRN C! 2 TIMER LOOP;
```

C! acts like POKE, rather than PRINT CHR$, as far as video memory goes; the table on page 211 of the June 1981 *80 Micro* spells out the differences.

Forth dialects also support array definitions. These are treated as variables with extended memory areas set aside for the data values' storage. The equivalent of Basic's DIM statement is the word Allot, used this way: VARIABLE (array name) (n) ALLOT, where n is the number of bytes of storage reserved, in addition to the two bytes that are automatically set aside by the word Variable. Remember, even Colorforth assumes you're working with 16-bit numbers unless otherwise

```
HEX (Square Wave Sound)
FF01 CONSTANT AD1
FF03 CONSTANT AD2
FF23 CONSTANT AD3
FF20 CONSTANT D/A
: 1SEL   AD1 C@ F7 AND AD1 C! ; (Puts 0 in bit 3 of $FF01)
: 2SEL   AD2 C@ F7 AND AD2 C! ;
: SNDENABLE   AD3 C@ 8 OR AD3 C! ; (Puts 1 in bit 3 of $FF23)
: TIME   F 0 DO LOOP ; (Approx. 0.001 sec. delay)
: SOUNDOFF   HEX 1SEL 2SEL SNDENABLE BEGIN FC D/A C! TIME
] 0 D/A C! TIME 0 UNTIL ; (Infinite loop)
;S
```

*Figure 1*



See List of Advertisers on Page 451

informed. So to set up a 10-member array named Info filled with 16-bit numbers, specify an additional (9 × 2) bytes: VARIABLE INFO 18 ALLOT.

If you use an array for 8-bit numbers such as graphics or ASCII codes, you can cut down on the amount of memory you must reserve. Variable gives you space for the first two numbers, and you just specify one byte for every additional storage position.

Storing data in an array or retrieving it is more complicated than in Basic, because Colorforth doesn't support subscripted variables. You specify the first storage location by using the array name (it's a variable, remember), and then add an offset to get to the desired location. For example, you can locate the first cell (the Forth name for a 16-bit-wide storage location) in our array Info just by calling Info; offset the Nth cell by 2x (N–1) bytes, so you can find out what's in, say, the fifth cell with the phrase INFO 8 + ?.

Of course, if you use an array for 8-bit numbers, there's no need to double the subscript N to get the offset. You must use the store and fetch commands—! and @, C! and C@—appropriate to the length of the numbers you are working with; Colorforth has no other way of knowing.

Colorforth can also handle multidimensional arrays. You must decide on a convenient method for converting multiple subscripts to single address offsets. Brodie's book gives a nice example for two-dimensional arrays, which wind up being stored columnwise. All the elements of the first column, starting with the uppermost, are stored, then all the elements of the second column, and so on. This is actually the way many high-level languages treat such data structures.

**Something Completely Different**

Here are a couple examples of the way you can use Colorforth with the Color Computer's I/O structure.

Anyone who has used the Color Computer's joysticks knows about memory address 65280, which is PEEKed to test the status of the trigger buttons. You can give this address a name, such as Port: 65280 CONSTANT PORT. Colorforth interprets this properly (the internal representation is treated as an unsigned 16-bit number). The word Looksee prints the contents of this location:: LOOKSEE 2 TIMER PORT C@ .;.

Entering Looksee from the keyboard causes the system to wait two seconds (assuming the previous definition of Timer, of course) and then prints the contents—255 or 127 if you press neither joystick trigger, 254 or 126 for the right trigger, and so on. Since a single memory location can only contain an 8-bit number, the proper fetch command is C@. It is easy to incorporate Looksee into an If. . .Then statement, causing a program branch, rather than having it print and stop.

While Port is a constant whose value is put on the stack when used in another word, there's nothing wrong with this value representing the address of something else. Remember, if Port is defined as a variable, its address is put on the stack, not the address of the trigger-button sensor.

The final example uses this sort of labeling, and is less trivial: a sound generator. This program generates a square wave at approximately 500 Hz and routes it to the Color Computer's composite video output. If you are interested in this sort of thing, you need detailed reference material. I recommend the Radio Shack reference or service manuals, or the *Facts* book from Spectral Associates. In any event, here's a rundown on the relevant aspects of Color Computer operation.

Internally synthesized sound uses the Color Computer's D/A converter. To be specific, a six-bit number loaded into the six high-order positions of address $FF20 is converted into analog voltage—the larger the binary number, the larger the voltage. The D/A is also used for joystick input and for the cassette interface.

To ensure that signals are routed properly, a few addresses associated with the computer's peripheral interface adapter (PIA) circuits have to be properly programmed ($FF20 also belongs to a PIA). Briefly, you need zeros in bit 3 of $FF01 and $FF03 (analog multiplexer), and a one in bit 3 of $FF23 (six-bit sound enable). To set 1 bit at an address, logically AND or OR the address's contents with an appropriate binary mask, and the program takes care of that. Then you need a loop to load those six bits of D/A input at an audio rate.

The Program Listing has the line numbers you might see if you composed it on a screen. (The parenthetical expressions are Forth's version of REM statements; they can appear anywhere, even within another statement.)

Line 0 tells the Colorforth interpreter that hex numbers are coming in at compilation time. Lines 1–4 label the addresses of interest to us, in hex. Lines 5–7 each fetch an eight-bit number from one of the locations, set bit 3 to what you need for sound output, and store the new number in the old number's location. The mnemonics represent select 1, select 2, and sound enable. You can verify that ANDing and ORing with these operands affects only bit 3 of the appropriate word. Just remember that the rightmost position is bit 0.

Line 8 defines a one-millisecond timing loop, with the upper limit in hex. Lines 9 and 10 contain the definition of Soundoff, which is the whole program. Two screen lines were used because of the definition's length. The square bracket that begins line 10 signals the interpreter that this is a continuation. Finally, line 11 contains the word that terminates compilation when the screen is loaded.

A few things about the colon definition of Soundoff: The definition includes hex so that it works properly even if called when the system has been operating in another base. Then it calls the words that set the software latches for sound output and enters an infinite Begin. . .Until loop. (The loop is guaranteed infinite because the logical flag tested to terminate the loop is defined to be 0, rather than being set as the result of any other process.)

The operations within the loop consist of setting the six high-order bits of $FF20 to ones and zeros, holding each setting for about one millisecond. The reason that you use $FC, and not $FF, for the high value is that the two low-order bits of this address are held at zero when you use the D/A in the six-bit sound output mode. ANDing and ORing are used to set $FF20.

**Summary**

Colorforth may seem complex when compared with Color Basic, but it's closer to Assembly language in the level of detail required. Remember, with Colorforth you get a higher degree of control over machine operations than Basic offers, coupled with high speed. I have not covered text manipulation, but the words for sophisticated text I/O are present in the starting vocabulary. This is where I should encourage you to get the software and the book and find out for yourselves.

Just one more thing: You can apply the bit twiddling used in the definition of Soundoff to program the SAM and VDG circuits for graphics. ■

*Scott Norman lives at 8 Doris Road, Framingham, MA 01701.*

# DISCOUNT SOFTWARE FOR THE TRS-80

### FARM and RANCH MANAGEMENT

| | | |
|---|---|---|
| BEEF PROJECTION | | $20.00 |
| BEEF RATION | | $15.00 |
| DAIRY COW | | $15.00 |
| FEED LOT | Models I and III | $35.00 |
| | Model II | $100.00 |
| FARM RECORD | | $35.00 |
| GOVERNMENT PROGRAM | | $15.00 |
| GRAIN STORAGE | | $15.00 |
| LOAN | | $15.00 |
| MACHINE COST | | $20.00 |
| RECORD MANAGEMENT | | $15.00 |
| ESTATE TAX | | $25.00 |
| FEED FORMULATOR | | $35.00 |
| PASTURE PROJECTION | | $20.00 |

HARDWARE REQUIREMENTS FOR THE ABOVE PROGRAMS (MINIMUM) TRS-80 Model I or III with 48K and one disk drive. For Model II Versions of the above programs please add $5.00 per program (Feed Lot Program Add $65.) Buy all the above programs and save even more. Model I and III Versions only $250.00 (Save $30) Model II Versions Only $350 (Save $55.)

## NEWDOS, DOSPLUS

### AND NOW

## L DOS!!

## YOUR CHOICE, ONLY $129 EACH.

NEWDOS, DOSPLUS AND L DOS are three of the most advanced operating systems ever available for the TRS-80. After spending thousands of dollars on your computer, why limit yourself by using an inferior operating system?

Order Today!!!!

| | | |
|---|---|---|
| LAZY WRITER | | $175 |
| ALCOR PASCAL | | $199 |
| MICRO CLINIC | | $24.95 |
| CHEXTEXT MICROSOFT | | $34.95 |
| LEVEL III BASIC | | $50 |
| TYPING TUTOR | | $15 |
| EDITOR ASSEMBLER + | | $30 |
| ADVENTURE | | $30 |
| MOD I FORTRAN | | $87.95 |
| A.L.D.S. | | $85 |
| MOD I BASIC COMPILER | | $195 |
| MOD I MuMATH | | $75 |
| OLYMPIC DECATHLON | | $24.95 |
| EDITOR ASSEMBLER (Disk) | | $49.95 |
| CP/M FORTRAN-80 | | $475 |
| CP/M BASIC COMPILER | | $375 |
| CP/M MuMATH-MuSIMP-80 | | $250 |
| CP/M COBOL COMPILER | | $699 |
| CP/M BASIC-80 | | $350 |
| BASIC COMPILER II | | $395 |
| COBOL-80 MOD II | | $699 |
| M/SORT | | $195 |
| ASPEN: | | |
| GRAMMATIK I/III | | $75 |
| GRAMMATIK II | | $99 |
| PICKLES & TROUT: | | |
| RANDOM HOUSE PROOF READER | | $50 |
| CP/M FOR MODEL II | | $185 |
| CP/M FOR MODEL 16 | | $220 |
| MISC: | | |
| BOOKKEEPER I/III | | |
| Cassette $49.95 | Disk | $59.95 |
| MICRO SEABATTLE | | |
| Cassette $19.95 | Disk | $24.95 |
| LIST: 1 (Telephone List) | | |
| Cassette $9.95 | Disk | $14.95 |
| COMPUTER BASEBALL | | |
| Cassette $14.95 | Disk | $19.95 |

## RADIO SHACK SOFTWARE

### MODEL I, III

| | | |
|---|---|---|
| Cassette Scripsit | 26-1505 | $34.95 |
| Cassette Portfolio | 26-1506 | $24.95 |
| Stock Pack | 26-1507 | $44.95 |
| Disk Mailing List | 26-1551 | $34.95 |
| General Ledger I | 26-1552 | $87.95 |
| Inventory Control I | 26-1553 | $87.95 |
| Accts. Payable | 26-1554 | $138.00 |
| Accts. Receivable | 26-1555 | $138.00 |
| Disk Payroll | 26-1556 | $175.00 |
| Business Mailing List | 26-1558 | $87.95 |
| Profile | 26-1562 | $69.95 |
| Scripsit | 26-1563 | $87.95 |
| Visicalc-Mod I | 26-1566 | $87.95 |
| Visicalc-Mod III | 26-1569 | $175.00 |
| Project Manager | 26-1580 | $87.95 |
| Checkwriter | 26-1584 | $87.95 |
| Super Scripsit | 26-1590 | $175.00 |
| Scripsit Dictionary | 26-1591 | $131.00 |
| Profile III Plus | 26-1592 | $175.00 |
| Desktop Plan-80 | 26-1594 | $175.00 |
| Budget Mgmt | 26-1603 | $22.00 |
| Advanced Stat. Analysis | 26-1705 | $39.95 |

### MODEL II

| | | |
|---|---|---|
| Gen. Ledger 1 Disk | 26-4501 | $175.00 |
| Inventory Mgmt. System | 26-4502 | $175.00 |
| Mod II Visicalc | 26-4511 | $263.00 |
| Profile II | 26-4512 | $170.00 |
| Profile II Plus | 26-4515 | $275.00 |
| Mod II Scripsit 2.0 | 26-4531 | $375.00 |
| Scripsit Dictionary | 26-4534 | $175.00 |
| Inventory Mod II | 26-4602 | $275.00 |
| Order Entry ICS | 26-4607 | $465.00 |

## GAMES FOR THE COLOR COMPUTER

## CHRISTMAS SPECIAL!!

### COSMIC DOGFIGHT, OFFENDER, PACKETMAN

GET ALL THREE OF OUR NEWEST
COLOR COMPUTER GAMES ON EITHER
DISK OR CASSETTE

## DISK $59.00   CASSETTE $44.00

### ★ ★ NEW RELEASES ★ ★

| | | |
|---|---|---|
| COSMIC DOGFIGHT | CASSETTE | 14.95 |
| | DISK | 19.95 |
| OFFENDER | CASSETTE | 29.95 |
| | DISK | 34.95 |

### OTHER FAVORITES

| | | |
|---|---|---|
| PACKETMAN | CASSETTE | 24.95 |
| | DISK | 29.95 |
| MOONLANDER | CASSETTE | 14.95 |
| | DISK | 19.95 |
| METEROIDS | CASSETTE | 21.95 |
| SPACE INVADERS | CASSETTE | 21.95 |

THESE ARE OUR CASH DISCOUNTED PRICES. C.O.D. AND CHARGE ORDERS ARE 3% HIGHER. ALL COMPUTERS ARE SHIPPED FREIGHT COLLECT. PRICES, SPECIFICATIONS AND AVAILABILITY ARE SUBJECT TO CHANGE WITHOUT NOTICE. IBM & IBM PERSONAL COMPUTERS ARE TRADEMARKS OF INTERNATIONAL BUSINESS MACHINES, INC. TRS-80 AND RADIO SHACK ARE TRADEMARKS OF TANDY CORPORATION.

918/825-4844

AMERICAN Small Business COMPUTERS

118 SO. MILL ST
PRYOR, OK 74361

# The Calculating Genius—Part II

by Rick Cook

**Charles Babbage's Analytical Engine was the ancestor of the modern computer. It was designed to run with punched-card control.**

*Ed. note: This is the second of a two-part series on Charles Babbage, inventor of the modern-day computer. The first part covered the development of the Difference Engine. This part focuses on the Analytical Engine.*

From 1834 on, Babbage's main concern was his Analytical Engine, a mechanical digital computer capable of performing elaborate mathematical analysis more or less automatically.

The idea of the Analytical Engine seems to have taken firm hold in 1833 when he took the small calculating device he had built from parts of the Difference Engine and hooked it up to feed the number appearing in the result column back into the machine as a difference, calculating a whole new series.

By the 1830s, mechanical methods of doing arithmetic had been known for a long time. Since multiplication can be accomplished by repeated additions, and division is the inverse of multiplication, theoretically any machine that could add could be made to do the other operations as well. But Babbage was looking for something more sophisticated. He wanted a machine that could perform these operations on large numbers at reasonable speed with a minimum of human intervention.

Babbage realized the major problem with this conception was not arithmetic, but control. The earlier machines had shown ways to handle arithmetic and the Difference Engine showed how to store numbers, but no one had ever tackled the control problem. To be useful, the machine would have to be able to perform operations in whatever sequence might be appropriate to the problem at hand.

The solution lay in two things: the use of punched cards to supply data and instructions and the use of a chain —a mechanical AND circuit.

Punched-card control was familiar to Babbage from the Jacquard looms that were busy by the hundreds in Europe weaving elaborate textiles. Jacquard hadn't invented the punched-card loom, but he had come up with a way of feeding cards into the loom so that a large number of cards could be used to control the pattern. By 1835 the French were using steam-powered Jacquard looms to weave very fine tapestries. On his travels, Babbage acquired a portrait of Jacquard woven to such size and complexity that it took 24,000 cards to program the loom.

The essence of Babbage's chain was the same as the plug boards that were used to program the earliest electronic computers. The Analytical Engine was actually a mass of separate mechanisms that could be coupled or uncoupled at the command of the instruction cards. Only those parts needed for a particular instruction would be coupled at any one time.

Since the machine was completely modular, wear and waste motion were held to a minimum. The mass in motion at any instant would be low and the chains of driven parts would be short. What's more, the Analytical Engine could be built and tested a piece at a time, and a component that failed could be taken out of service without shutting down the rest of the machine.

Clearly, Babbage had learned from his experiences with the Difference Engine. But in the end flexibility was a mixed blessing since it encouraged Babbage to keep tinkering with the design for the rest of his life. Babbage kept adding new features to enlarge its capacity, even though the basic outline was fixed by about 1840.

If the Analytical Engine had ever been built, it would have been even more impressive than the Difference Engine. Babbage's ultimate design called for it to be able to store 1,000 50-digit signed numbers, each in a column of disks like those used for the

Difference Engine. Numbers would be read into and out of memory by rack-and-pinion gears. The memory area was called the store and the collection of mechanisms that did the computing was called the mill.

**Programming the Analytical Engine**

The machine was programmed with three sets of punched cards. Each set was tied together to form a chain that could be pulled through the machine in the manner of the cards controlling a Jacquard loom. The machine could move the cards back and forth through the reader so that a set of instructions could be repeated—the equivalent of a Do. . . Until statement.

The first set of cards, the number cards, contained the data to be acted on. The second set, the operations cards, directed the operations to be performed. The third set, the directive cards, told the machine where to store the numbers and where to store the results. In effect, they acted like the stack register in a microprocessor.

Programming the Analytical Engine was done the same way a machine-language program is written today. Memory was at a premium, so a high-level language like Basic was out of the question. On the other hand, the speeds for reading a card and reading a number from memory were roughly comparable, so the distinction we make between main and auxiliary memory didn't exist.

This makes it hard to compare the Analytical Engine's storage capacity meaningfully with a modern computer. The memory locations probably would have only been used for intermediate results in calculations. The data being operated on would not have been read into memory first.

When it came to operation, the machine was not as automatic as a modern computer. A number of cases required the operator's intervention. The Analytical Engine would not normally calculate trig functions in the course of a program. Instead, it would signal the operator what it needed. The operator would take the card bearing the needed function out of the precalculated library and feed it into the machine. The engine would verify that the card it had been given was the right one and then go on with its calculations.

The machine also had a provision for indicating a variety of errors, including overflow and underflow.

On the whole, the Analytical Engine would probably have been about as flexible and powerful as such first-generation machines as ENIAC, although very much slower.

**The Later Years**

After his experiences with the Difference Engine, Babbage was not inclined to go to the government for money again. He told the government what he was up to, but the officials responded scathingly to what they considered another request for money. For the rest of his life Babbage bore a grudge against officialdom. He went ahead with the work at his home in London, concentrating on perfecting the design and only building those parts he needed to make sure they would work.

*"Time and again Babbage was balked by bad management practices."*

In 1842, L.F. Menabrea, an Italian mathematician and engineer, published an article in a Swiss magazine describing the Analytical Engine and how it worked. Babbage had talked to Menabrea about the engine when he went to Italy in 1839. In 1842 the paper was translated into English by Ada Agusta, Countess Lovelace, another friend of Babbage's and a considerable mathematical talent in her own right. With Babbage's encouragement, she prepared a series of extensive notes on the Analytical Engine which she attached to Menabrea's paper. Those notes included a description of machine-language programming that can still serve as a nontechnical introduction to the subject.

The paper and the generally positive reaction to it were the closest things Babbage got to support for his work. Once again there was wide admiration for his genius, but this was tempered by a near-total lack of comprehension of what the Analytical Engine was and how it would operate. For every General Menabrea or Lady Lovelace, a hundred scientists or mathematicians found the Analytical Engine foolish, impossible or merely incomprehensible.

This was especially true in England. On the Continent and in the United States, Babbage's stock stood considerably higher. In his later years, Babbage received and cherished many letters from Americans, including requests from the U.S. Government for advice on cryptography and calculating machines.

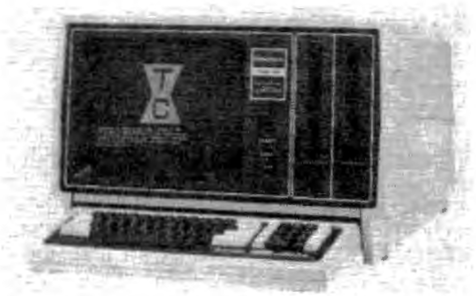
Babbage kept himself busy with many projects besides the Analytical Engine. At one point he conceived the idea of building a machine to play tic-tac-toe and exhibiting it to raise money to finance the Analytical Engine. He got as far as outlining the basic rules for teaching a machine to play any game and devising a tic-tac-toe strategy based on the magic square before he discovered that the public preferred General Tom Thumb to even the cleverest automata. In addition, he invented an occulting light for lighthouses, the opthalmoscope, published a book on ciphers, papers on statistics and geology, and pamphlets advocating life peerages and the income tax.

However, as time went on his life tended to become more and more circumscribed. His friends died off and the bitterness he felt over his disappointments deepened. Still he labored over his Analytical Engine, refining and redesigning it, although he knew by now it would probably never be built.

In his book on the history of computing, Herman H. Goldstine quotes Lord Moulton recounting a visit to Babbage in his last years:

> "One of the sad memories of my life is a visit to the celebrated mathematician and inventor, Mr. Babbage. He was far advanced in age but his mind was still as vigorous as ever. He took me through his workroom. In the first room I saw parts of his original Calculating Engine [the Difference Engine], which had been shown in an incomplete state many years before and had even been put to some use. I asked him about its present form.
>
> "'I have not finished it because in working at it, I came on the idea of my Analytical Engine, which would do all that it was capable of doing and much more. Indeed, the idea was so much simpler that it would have taken more work to complete the Calculating Machine than to design and construct the other in its entirety, so I turned my attention to the Analytical Machine.' I asked if I could see it. 'I have never completed it,' he said, 'because I have hit upon a different and far more effective method and this rendered it useless to proceed on the old lines.' Then we went into the third room. There lay scattered bits of mechanism, but I saw no trace of any working machine. Very cautiously I approached the subject and received the dreaded answer. 'It is not constructed yet, but I am working on it and it will take less time to construct it altogether than it would to complete the Analytical Machine in the state that I left it.'

# TRS-80 MODEL 16



**MODEL 16-1 DRIVE** **$4499**

**MODEL 16-2 DRIVE** **$4995**

**DT-1 VIDEO DATA TERMINAL** **$620**

For use with Model 16 or Mainframe Computer

# TRS-80 MODEL II

## Now Only $2895

NEW!

# HARD DISK DRIVES

## HARD DISK SYSTEM FOR THE MODEL III

**6 MEG $1795**

**12 MEG $1995**

**19 MEG $2195**

COMPLETE, READY TO RUN. NOW AVAILABLE!!

**LDOS REQUIRED — $129**

*unformatted capacity

## CAMEO SUBSYSTEMS FOR THE MODEL II

**DRIVE MOUNT ONE** **$4995—**
- Western Dynex D6222 Drive
(5 mb Fixed/5 mb Removable)
Controller, Adaptor and Cables

**1010 SUBSYSTEM** **$4995—**
- Data Peripheral Lynx Drive.
(10 mb Removable Winchester)
Controller, Adaptor and Cables

**1040 SUBSYSTEM** **$5995—**
- Data Peripheral Puma Drive
(40 mb Fixed Winchester)
Controller, Adaptor and Cables

**1020 SUBSYSTEM** **$7995—**
- 2 Data Peripheral Lynx Drives
(20 mb Removable Winchester)
Controller, Adaptor and Cables

**1050 SUBSYSTEM** **$8995—**
- Data Peripheral Lynx and Puma Drive
(40 mb Fixed/10 mb Removable)
Controller, Adaptor and Cables

**—EXTRA DRIVES—**

| | |
|---|---|
| WESTERN DYNEX D622 | $4495 |
| DATA PERIPHERAL LYNX | $3995 |
| DATA PERIPHERAL PUMA | $4995 |

**MULTIPLEXORS**

**MX200-2 PORT MULTIPLEXOR $495**
(Adaptors Not Included)

**MX400-4 PORT MULTIPLEXOR $750**
(Cables and Adaptors Included)

THESE ARE OUR CASH DISCOUNTED PRICES. C.O.D. AND CHARGE ORDERS ARE 3% HIGHER. ALL COMPUTERS ARE SHIPPED FREIGHT COLLECT. PRICES, SPECIFICATIONS AND AVAILABILITY ARE SUBJECT TO CHANGE WITHOUT NOTICE. IBM & IBM PERSONAL COMPUTERS ARE TRADEMARKS OF INTERNATIONAL BUSINESS MACHINES, INC. TRS-80 AND RADIO SHACK ARE TRADEMARKS OF TANDY CORPORATION.

**918/825-4844**

# AMERICAN Small Business COMPUTERS

118 SO. MILL ST
PRYOR, OK 74361

219

"I took my leave of the old man with a heavy heart. When he died a few years later, not only had he constructed no machine, but the verdict of a jury of kind and sympathetic scientific men who were deputed to pronounce upon what he had left behind him, either in papers or in mechanism, was that it was too incomplete to be capable of being put to any useful purpose."

Lord Moulton's last statement is somewhat inaccurate. Seven years after Babbage's death, the Royal Society appointed a committee to look at the Analytical Engine and decide whether it should be completed. The committee reported that all, or almost all, of the basic design work was done, but the cost of translating the design to working drawings and actually building the machine would be enormous. For this reason the committee recommended letting the Analytical Engine lie.

Charles Babbage died on October 18, 1871, at the age of 79 without ever seeing an Analytical Engine. After his death, his son, Henry, published a book on the engines that included many Analytical Engine details. Henry also commissioned the building of the Analytical Engine's mill and presented it to the Science Museum in London.

Probably the best epitaph that Charles Babbage could have comes from his autobiography, *Passages in the Life of a Philosopher*, written in 1864:

"The great principles on which the Analytical Engine rests have been examined, admitted, recorded, and demonstrated. The mechanism has now been reduced to unexpected simplicity. Half a century may elapse before anyone without these aids, which I leave behind me, will attempt so unpromising a task. If, unwarned by my example, any man should undertake and shall succeed in really constructing an engine embodying in itself the whole of the executive department of mathematical analysis upon different principles or by simpler mechanical means, I have no fear of leaving my reputation in his charge, for he alone will be fully able to appreciate the nature of my efforts and the value of their results."

### The Final Questions

In any event, it was nearly 70 years after Charles Babbage's death before his dreams were to be realized. In response to the need for new kinds of tables—firing tables for artillery—other men built machines like the one he had envisioned. These new analytical engines were far faster than Babbage's and used electronics rather than mechanics, but the first generation was remarkably similar in size and general structure to what Babbage had tried to build.

With the invention of these machines, Babbage was lifted out of obscurity and his memory honored for his ideas and attempts.

But still the questions remain. Could his calculating machines have worked? If so, why did Babbage fail?

The usual answer is that they could not have worked. Babbage was too far ahead of his time; the materials and techniques for shaping the parts weren't available; the designs were too complex. Yet the evidence doesn't support any of these conclusions.

Granted that the machines were complex and that Babbage was pushing the state of the art, but the designs were exceedingly clever and Babbage took great pains to see that they could be manufactured and would function in spite of their complexity.

Certainly the Difference Engine should have worked. By the time work stopped, all the technical problems had been solved and its complexity was the complexity of a single, fairly simple assembly repeated many times over.

The Analytical Engine is not as certain since it was more complicated and never got beyond the design stage. Still, it displays the same ingenuity of design and attention to reliability that marked the Difference Engine. In addition, the Analytical Engine was actually a collection of separate mechanisms that only joined to become a larger mechanism on command. Given the modular principles on which it was designed, it should have been possible to build a basic Analytical Engine out of the best-proved various mechanisms and then modify it as the more elaborate and troublesome parts were built and tested. This is essentially the way ENIAC and a great many homebrew microcomputer systems were built.

Then, if the machines could have been constructed and if they would have worked, why did Babbage fail? I think the answer lies in what he could not invent.

Babbage pioneered in cybernetic theory, machine and tool design, operations research, and a number of other engineering fields. But he was less advanced than many of his contemporaries in what we now know as project engineering. It was this lack that doomed his efforts. Babbage was capable of inventing the machines, but he could not invent the procedures needed to make their construction go smoothly.

Time and again, Babbage was balked by bad management practices. His failure to get the original agreement with the government in writing, his failure to set up an adequate system of progress payments, his inability to realistically estimate costs and construction times, his failure to secure ownership of crucial tools and fixtures, and his inability to stop designing and get on with the building all contributed to the failure of his projects.

To us these procedures are as commonplace as the idea of a digital computer. To Babbage and his contemporaries, both concepts were blindingly original. It was the genius of Charles Babbage that he was able to grasp one idea and his downfall that he did not understand the other. ■

*Rick Cook can be reached at Box 15193, Phoenix, AZ 85060.*

## BIBLIOGRAPHY

Babbage, Charles; *Charles Babbage and His Calculating Engines*, edited and with an introduction by Phillip and Emily Morrison; Dover Publications, New York, 1961.

Babbage, Charles; *On the Economy of Machinery and Manufactures*, Reprints of Economic Classics; Augustus M. Kelley Publishers, New York, 1963.

Dubbey, J.M.; *The Mathematical Work of Charles Babbage*; Cambridge University Press, Cambridge, 1978.

Goldstine, Herman H.; *The Computer from Pascal to von Neumann*; Princeton University Press, Princeton, NJ, 1972.

Lardner, Dr. Dionysius; "Babbage's Calculating Engine;" *Edinburgh Review*, July 1834, Vol. 59, pp. 263-327.

Moore, Doris Langley; *Ada, Countess of Lovelace: Byron's Legitimate Daughter*; Harper & Row, New York, 1977.

Moseley, Maboth; *Irascible Genius: The Life of Charles Babbage*; Henry Regnery Company, Chicago, 1964.

Smiles, Samuel; *Industrial Biography: Iron Workers and Tool Makers*, Reprints of Economic Classics; Augustus M. Kelley Publishers, New York, 1968.

*A graphics program for your Color Computer.*

# Flaky

*Sidney C. Garrison*
*1406 Sunny Court*
*College Station, TX 77840*

This graphics program shows off the power of the Extended Basic Color Computer. The three-dimensional and moire pattern effects which appear from the strict geometric shapes can mesmerize an audience.

The designs produced depend on two parameters, $K \geq 0$ and $0 \leq J \leq (K-2)/2$, both integers. On running the program a random pattern is shown and then the menu appears. Menu option one constructs and displays, for a short time, a previously selected list of patterns. You are then returned to the menu. Menu option two increments K and J in turn so that all possible patterns are seen; press Break to stop. Menu Option three lets you input a K and J value of your choice. The design is constructed and displayed until you press any key except break; and then you're returned to the menu. Before you enter the program, do not PCLEAR since it is assumed the graphics variables are at their default settings.

In mathematical terms the program adds a certain complex $K^{th}$ root of unity to a run-

***The Key Box***

**Color Computer**
**16K RAM**
**Extended Color Basic**

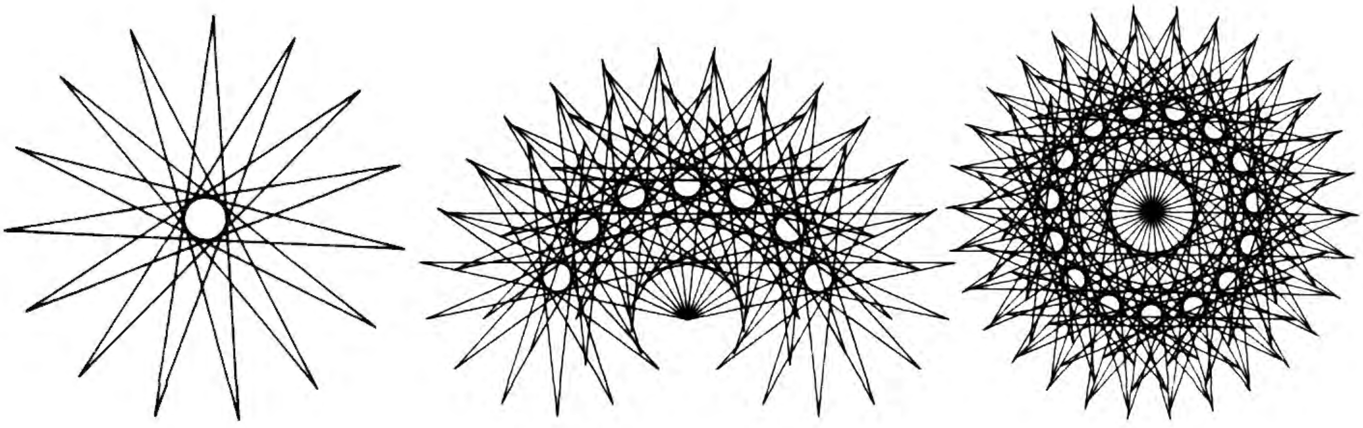


*Fig. 1. Stages in the construction of the design with K = 15 and J = 6*

# WE STOCK PRINTERS!

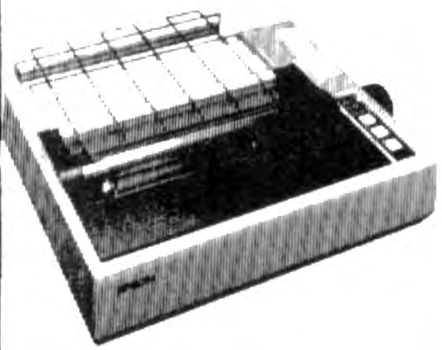
**EPSON • OKIDATA • C. ITOH • RADIO SHACK • SMITH CORONA**

## • EPSON •

| | |
|---|---|
| MX-80 | $459 |
| MX-80 F/T | $525 |
| MX-100 | $685 |

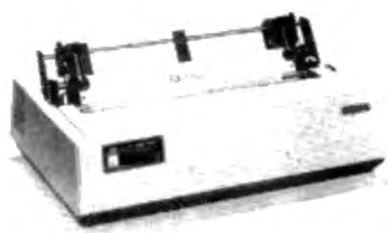
WE STOCK A COMPLETE LINE OF EPSON ACCESSORIES

CALL FOR PRICES!!



## • OKIDATA •

| | |
|---|---|
| ML-80 | $350 |
| ML-82A | $425 |
| ML-83A | $699 |
| ML-84P | $1049 |
| ML-84S | $1299 |
| OKIGRAPH | $99 |



## • C. ITOH •

### PROWRITER

| | |
|---|---|
| 8510 PARALLEL-10" | $489 |
| 8510 SERIAL-10" | $639 |
| 1550 PARALLEL-15" | $719 |
| 1550 SERIAL-15" | $1295 |

### STARWRITER

| | |
|---|---|
| F-10 40 CPS, PARALLEL | $1450 |
| F-10 40 CPS, SERIAL | $1599 |
| F-10 55 CPS, PARALLEL | $1699 |
| F-10 55 CPS, SERIAL | $1850 |

## RADIO SHACK PRINTERS

**DMP-100** (26-1253) **$339.95**
50 CPS, UPPER & LOWER CASE, 10" CARRIAGE

**DMP-200** (26-1254) **$679.95**
120 CPS, BIDIRECTIONAL, ADJUSTABLE TRACTOR, 10" CARRIAGE

**DMP-400** (26-1251) **$1015.00**
140 CPS, ADJUSTABLE, REMOVABLE TRACTOR FEED, 15" CARRIAGE

**DMP-500** (26-1252) **$1525.00**
220 CPS, HEAVY DUTY DESIGN, REMOVABLE ADJUSTABLE TRACTOR, 15" CARRIAGE

**DWP-410** (26-1250) **$1269.95**
300 WORDS PER MINUTE, 15" PLATEN

**DAISYWHEEL II** (26-1158) **$1695.00**
TRACTOR FEED (26-1447) **$275**

## DAISY WHEEL PRINTER

**$599.00**

## SMITH CORONA TP-I

(35-001)

THESE ARE OUR CASH DISCOUNTED PRICES. C.O.D. AND CHARGE ORDERS ARE 3% HIGHER. ALL COMPUTERS ARE SHIPPED FREIGHT COLLECT. PRICES, SPECIFICATIONS AND AVAILABILITY ARE SUBJECT TO CHANGE WITHOUT NOTICE. IBM & IBM PERSONAL COMPUTERS ARE TRADEMARKS OF INTERNATIONAL BUSINESS MACHINES, INC. TRS-80 AND RADIO SHACK ARE TRADEMARKS OF TANDY CORPORATION.

**918/825-4844**

# AMERICAN Small Business COMPUTERS

118 SO. MILL ST
PRYOR, OK 74361

221

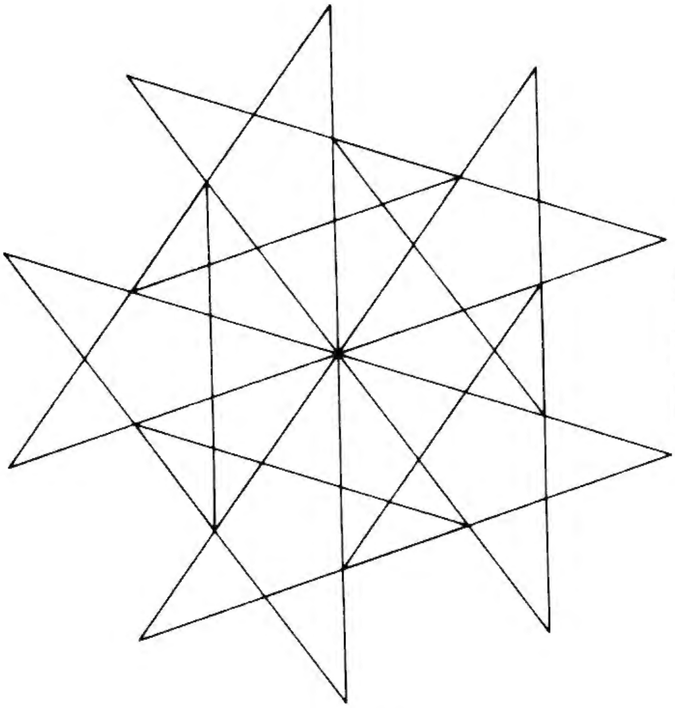


*Fig. 2. K = 5 and J = 1*

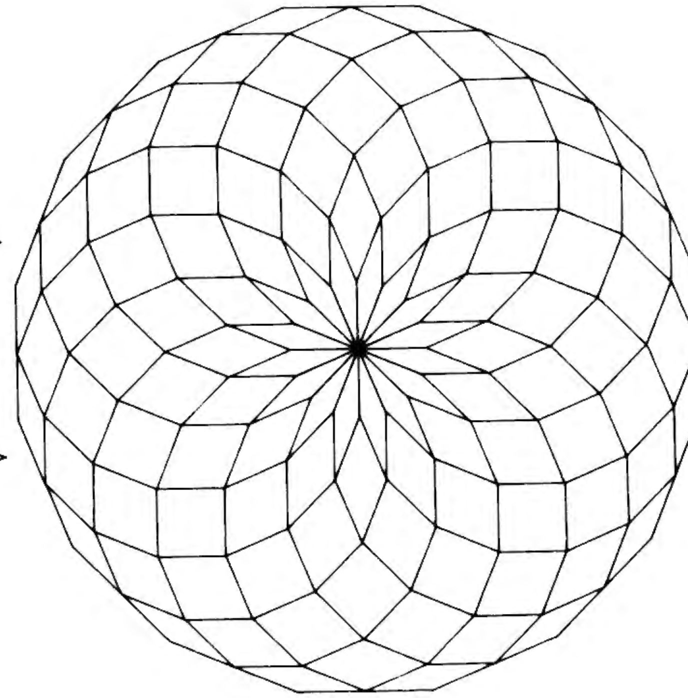


*Fig. 3. K = 16 and J = 0*

ning sum. Successive values of this sum are multiplied by a scaling factor and connected by a line on the screen which represents the complex plane. The root added at each stage depends on the J value. The sequence of pictures in Fig. 1 show stages in plotting the design with K equals 15 and J equals 6. The final graph is 15 copies of the first, each with the center rotated at an angle of 24 degrees (= 360/15). The calculation of the scaling factor is the only difficult mathematics involved, which is necessary so the pattern fills the screen for any value of the parameters. For a more detailed discussion of the mathematics involved see "Picturesque Exponential Sums, I" by D. H. and Emma Lehmer in the November 1979 issue of the *American Mathematical Monthly*.

Another very striking pattern can be formed with this program. I call these medallions; the effect depends on the eye's ability to pick out regularity from complex data. If K is large, say $60 \leqslant K \leqslant 400$, and J equals $(K-2)/2$, then a series of equally spaced rays from center screen outward are formed. These rays are actually stair steps. The eye organizes the steps from adjacent lines into curves. Use the program to experiment and find other patterns. The program organization is parallel to the description given in paragraph two. I added spaces and comments to make the structure clear. Copy the print statements exactly to avoid split words. The A$ equals INKEY$ at line 3050 allows you to temporarily stop execution using Shift@ without omitting the pause after construction of the design in menu option three. At line 10030 a procedure for calculating the greatest com-

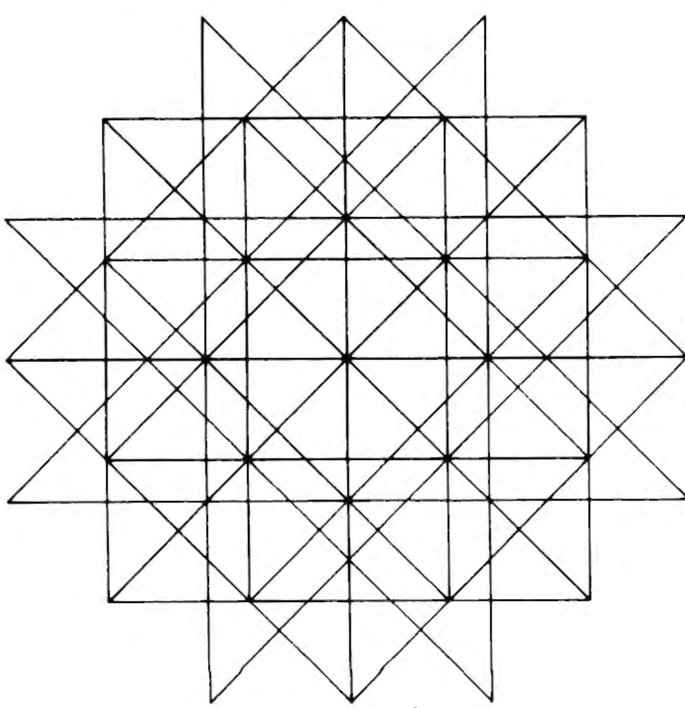


*Fig. 4. K = 8 and J = 2*

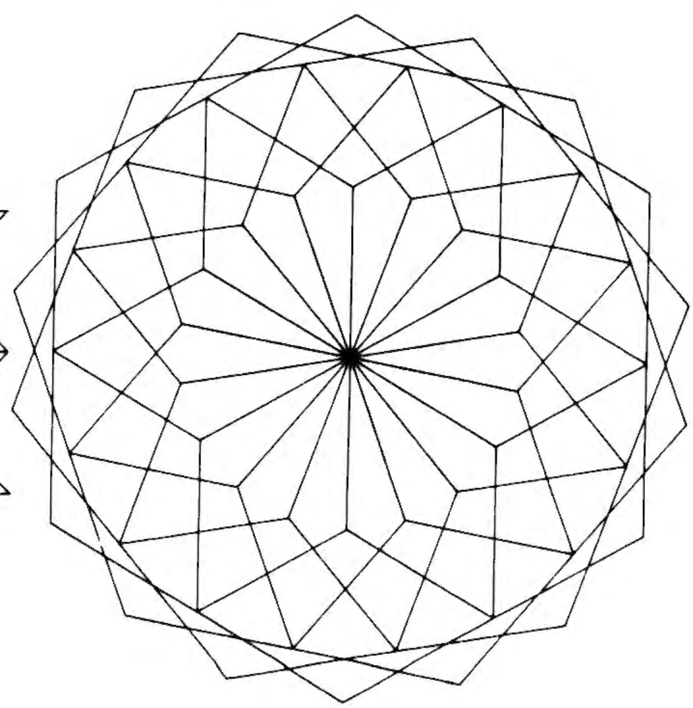


*Fig. 5. K = 18 and J = 2*

# A Computer That Writes Programs For You.

### What will they think of next..?


Your computer is fantastically fast.. once it knows what to do. You probably realize that a computer is really the combination of hardware and software, working together smoothly, to give you what you want. Either one alone is useless. Software is really the key the "mind" of a computer system. Every project or task you want to do requires a new specific software application to make it behave exactly the way you desire.

Of course, you may be able to "force-fit" an application into some existing canned program you have, but to really get results, you need a separate application program to run on your computer.

Until now, that meant you were forced to pay money for application software off the shelf, or if you could afford it, have it custom written for you, or, if you are qualified, do it yourself...spending endless hours figuring it out and writing it. Now, your computer can write individual application programs for you. These programs are each separate, unique software programs that run in standard Basic on your computer.

A company named FutureSoft has developed this exciting and long awaited remarkable working tool for you. There are two versions called *Quikpro+Plus* and standard *Quikpro*. Both of them create unique separate Basic programs for you...to do exactly, precisely, what you want to do. And listen to this.. you create a new program in minutes instead of hours.

You can quickly generate a new program when you want it. You can generate thousands of different unique programs, each one standing alone as a complete program that runs in Basic. Best of all, *you* do not have to be a programmer to do it. The *Quikpro* software becomes your personal programmer, waiting to do your work for you any time of day or night you choose to use it.

The custom programs you generate from this software provide for: Data Entry, Additions, Changes, Record Locating & Searches, great variety of Computations, and Report Printing (if you have a printer). It lets you decide what data to manipulate and how to manipulate it. It lets you decide the formats you want to appear on your screen and/or to print out in a report. It lets you use differing formats on the same data base. It lets you make calculations from data within records without altering the data base. It lets you report results with or without including the base data from which results were calculated.

All this is included in the ability/power of the program you create. You do it by simply answering questions that appear on your screen. Instantly, the *Quikpro* software instructs the computer to perform complex and error free instructional sequences. You get the immediate benefits of professionally written software for your application.

The resulting custom program is truly a separate Basic program. You can list it, you can modify it, you can actually see what makes it tick. You can even ask it to print out its own operating instruction manual so others can run it for you. Finally, you can really tap the speed and power of your computer the way you really want. You can create new programs for every use you have in Business, Science, Education, and Hobby areas. And you can start now.

The software is available immediately from the creators. It comes in two versions. If you want to generate separate Basic programs with all the data handling plus Calculations and Report Printing features, you want *Quikpro+Plus*. Specify to run on TRS80 Model I and Model III at only $149; to run on TRS80 Model II at $189.

If you do not need Calculation ability or Report Printing in the separate Basic programs you will create from this program generating software, then standard *Quikpro* will do the job for you. Standard *Quikpro* to run on TRS80 Model I or Model III is $89; to run on TRS80 Model II is $129. (Later on you can always trade up to the Plus Versions for only the cost difference between the two)

Both programs are available to run on many other computers besides TRS80. Details are available by calling or writing.

You can order right now by phone or mail. If you have Visa/Mastercard, call toll-free from:

| | |
|---|---|
| All States except CA-AL-HI | 1-800-824-7888 OP# 441 |
| From California call | 1-800-852-7777 Op# 441 |
| From Alaska/Hawaii call | 1-800-824-7919 Op# 441 |

Operators on duty 24 hours daily. Operators can not answer technical questions. If you need technical information or want to order from Canada or other nations, call 1-904-269-1918 during office hours, Eastern Time, Mon.-Fri.

Send mail orders with check, money order or credit card information to: FutureSoft, P.O. Box 1446-D, Orange Park , Florida 32073. FutureSoft gives you a satisfaction or your money refunded guarantee for 10 days from delivery. You can run the software yourself on your own computer and see with your own eyes what it can do for you. Order now. ✓66

*ADVERTISEMENT*

mon divisor of integers TK and TJ is used in the scaling mentioned above and to avoid retracing lines. The examples in Figs. 1-5 were done on an HP7225A plotter, but Extended Basic Color Computer's PMODE4 resolution (256 by 192) gives a clean display for reasonable values of K and J. ■

*Sidney Garrison is employed by Texas A&M University.*

```
100 '* * * * * * * * * * * * * * * *
110 'FLAKY  BY S.GARRISON  12/81
120 '* * * * * * * * * * * * * * * *
130 PI=3.14159265                  : HALF=.5
  : HH=95                          : HW=128
  : UNIT=1
200 ' GREETING * * * * * * * * * *
210 DATA 15,1,5,0,7,2,16,0,200,     99,8,2,5,1,13,5,8,0,9,1,6,0,
    120,39,9,3
220 UL=RND(12)
230 FOR I=1 TO UL                  : READ K,J
  : NEXT I
240 READ K,J                       : GOSUB 10000
  : GOSUB 10500                    : RESTORE
300 ' MENU * * * * * * * * * * * *
310 CLS
320 PRINT@39,"FLAKY: SNOW FLAKES"
330 PRINT@96,"   GIVEN PROPER VALUES OF K AND J THE COMPUTER PLO
TS A DESIGN.  VALUES MAY BE PICKED AUTOMATICLYOR BY THE USER:"
340 PRINT"  1.AUTOMATIC-A FEW NICE VALUES.  2.AUTOMATIC-SHOWS AL
L FLAKES.   3.MANUAL-YOU PICK K AND J.       4.END PROGRAM."
350 PRINT"HIT 1,2,3 OR 4."
360 A$=INKEY$                      : IF A$="" THEN 360
370 V=VAL(A$)                      : IF V*(5-V)<1 THEN 300
380 ON V GOSUB 1000,2000,3000,      4000
390 GOTO 300
1000 ' A FEW NICE FLAKES * * * *
1010 FOR N=1 TO 13                 : READ K,J
  : GOSUB 10000                    : GOSUB 10500
  : NEXT N
1020 RESTORE                       : RETURN
2000 ' SHOWS ALL FLAKES * * * *
2010 K=3
2020 FOR J=0 TO (K-2)/2            : GOSUB 10000
  : GOSUB 10500                    : NEXT J
2030 K=K+1                         : GOTO 2020
3000 ' PICK YOUR DESIGN  * * * *
3010 CLS                           ;PRINT
3020 INPUT"   INPUT PARAMETERS K (INTEGER  K>2) AND J (INTEGER 0
<=2J<=K-2). ";K,J
3030 CLS                           ;IF INT(K)<>K OR INT(J)<>J OR
    K<3 OR J<0 OR J>(K-2)/2 THEN    PRINT
  : PRINT "ILLEGAL K OR J, READ DIRECTIONS"
  : PRINT                          : GOTO 3020
3040 GOSUB 10000
3050 A$=INKEY$
3060 A$=INKEY$                     : IF A$="" THEN 3060
3070 RETURN
4000 END
10000 'SUBROUTINE PLOT  * * * *     INPUT: LEGAL K,J
  OUTPUT: GRAPH
10010 ASTEP=2*PI/K                 : BSTEP=ASTEP*(J+1)
10020 S=1/SIN(PI*(J+1)/K)          : TK=K
  : TJ=J+1 '* S=SCALE FACTOR* *
10030 TK=TK-INT(TK/TJ)*TJ          : IF ABS(TK)>HALF THEN TT=TJ
  : TJ=TK                          : TK=TT
  : GOTO 10030 '* TJ=(K,J+1)* *
10040 TJ=INT(TJ+HALF)              : TK=K/TJ
  : IF  TK-INT(TK/2)*2 > HALF       THEN S=S*COS(PI*TJ/(2*K))
10050 UI=K+HALF                    : UL=TK-HALF
10060 PMODE4,1                     : PCLS
  : SCREEN1,1                      : 'INTIALIZE SCREEN * * * * *
10070 ' DRAW !! * * * * * * * *
10080 FOR I=UNIT TO UI             : CH=ZERO
  : CV=ZERO                        : A=(I-1)*ASTEP
10090 FOR L=UNIT TO UL             : CH=CH+COS(A)/S
  : CV=CV+SIN(A)/S                 : LINE -(HW+INT(HH*CH+HALF),
    HH+INT(HH*CV+HALF)),PSET       : A=A+BSTEP
  : NEXT L
10100 LINE -(HW,HH),PSET           : NEXT I
10110 RETURN
10500 ' DELAY  * * * * * * * * * *
10510 FOR I=1 TO 800               : NEXT I
  : RETURN
```

*Program Listing 1*

# IDS MICROPRISM

## $689.88
**UPS Delivered**

- 110 cps bi-directional, logic-seeking
- Single pass correspondance quality print
- 10, 12, & 16 cpi with extended mode (24 x 9 dpi)
- 84 x 84 dpi Dotplot graphics; 1/48" line space

### Okidata Printers

| | |
|---|---|
| MICROLINE 82A | $439.88 |
| ML82A TRACTOR | $59.88 |
| MICROLINE 83A | $694.88 |
| OKIGRAPH ROM | $44.88 |
| MICROLINE 84 Parallel 200 cps | $1044.88 |
| MICROLINE 84 RS-232C 200 cps | $1169.88 |

### Anadex Printers

| | |
|---|---|
| DP-9500A | $1474.88 |
| DP-9510A | $1474.88 |
| DP-9620A | $1569.88 |

### Brother Printers

| | |
|---|---|
| DAISYWRITER 2000 | $1089.88 |
| TRACTOR | $49.88 |
| CABLE | $149.88 |

### IDS Printers

| | |
|---|---|
| PRISM 80 W/4-colors, Sprint Mode, Dot Plot, & Cut Sheet Guide | $1329.88 |
| PRISM 80 w/o color | $1174.88 |
| PRISM 132 W/4-colors, Sprint Mode, Dot Plot, & Cut Sheet Guide | $1699.88 |
| PRISM 132 w/o color | $1254.88 |

### TEC Printers

DMP85 ........ $469.88

The generic version of NEC & Prowriter printers. Features 120 cps, bi-directional, logic-seeking, 1.3K buffer. 5 fonts, 8 sizes on 9x9 matrix, w/proportional print, true descenders, & Greek/Math font. 160 x 144 dots/inch Hi-Res graphics matrix, 1/144" line feed. Friction & tractor standard, rear paper path.

### C. Itoh Printers

| | |
|---|---|
| C. ITOH PROWRITER | $499.88 |
| C.ITOH PROWRITER Parallel & RS-232C | $619.88 |
| C. ITOH PROWRITER 2 | $734.88 |
| C.ITOH PROWRITER 2 Parallel & RS-232C | $799.88 |
| C. ITOH F-10/40 STARWRITER Parallel, 40 cps | $1499.88 |
| C. ITOH F-10/55 PRINTMASTER Parallel, 55 cps | $1799.88 |
| F-10 TRACTOR | $289.88 |

### NEC Printers

| | |
|---|---|
| NEC PC 8023A-C | $509.88 |
| NEC 3530 (35cps) | $1809.88 |
| NEC 7730 (55 cps) | $2579.88 |

### Centronics Printers

| | |
|---|---|
| CENTRONICS 122-1 | $979.88 |
| CENTRONICS 352 | $1614.88 |
| CENTRONICS 353 | $2274.88 |
| CENTRONICS 739-1 | $659.88 |

### Smith-Corona

SMITH CORONA TP-1 .... $599.88
Specify 10 or 12 cpi
Specify Parallel or RS-232C

### Call For Prices

On QUME & DIABLO Daisywheels. QUANTEX, DATASOUTH, DIP, MPI, & other printers available

### Lynx TRS-80 Modem

**$229.88 UPS DELIVERED**

- Model I & III compatible modem system
- Auto dial/auto answer
- Works with ST-80 & any other software
- Works with or without RS-232C

### MicroTerm

An excellent terminal program from Micro-Systems Software. Microterm features simultaneous input while at menu, auto dial support (including pre-programed dialing and transmission without operator intervention), direct file transfer, 34K of capture buffer, and a high operational baud rate (near 9600). Well-suited to both the Lynx & Hayes Smartmodem's auto dial & auto answer features. Specify Model I or III when ordering.

MICROTERM .......... $89.88

### MODEMS

| | |
|---|---|
| HAYES SMARTMODEM | $229.88 |
| HAYES 1200 BAUD SMARTMODEM | $574.88 |
| RS-232C CABLE | $29.88 |
| SIGNALMAN MARK I | $89.88 |
| NOVATION AUTO CAT | $224.88 |
| NOVATION 1200 BAUD AUTO CAT | $569.88 |

### Model 3

Model 3 with 48K RAM and two single sided, double density drives, delivered ready to run. Fully compatible with all hardware/software. RS-232C port optional. Non-Radio Shack warranty, but full 90 days on parts & labor.

MODEL 3 .......... $1799.88

### Disk Drives

40 TRACK SS .......... $239.88

### Dosplus

The finest DOS available. Features a BASIC array sort (multi-key, multi-array), input @ (controlled screen input), random access and ASCII modification, BASIC checks for active "DO", repeat of last DOS command, device routing, single file conversion from Model III TRSDOS, reading 40-track disks in 80 track drives, etc. Specify Model I or III, single or double density, 40 or 80 track disk drive.

DOSPLUS 3,4 .......... $104.88

### Quic-N-Easi

**Quic-N-Easi lets anyone** write custom business application programs without BASIC—and you do it faster. Now you can try **Quic-N-Easi** for 15 days FREE. You're satisfied or your $195 refunded—no questions asked. Non-programmers & experts alike can write sophisticated business programs with **Quic-N-Easi**. Inventory, records, sales tracking, budgeting—and you'll program faster, with greater flexibility, like a pro—or your money back.

| | |
|---|---|
| TRS-80 Model I/III Quic-N-Easi | $195.88 |
| Model II/16 (CP/M required) Quic-N-Easi | $195.88 |

### NEWSCRIPT 7.0

ProSoft's **NEWSCRIPT** supports most printers (including the new Microline 84), and features true right-justified proportional printing, single and double-width type within text, subscripts, superscripts, underlining, boldfaces, multiple character pitch, full-screen editing, global search/replace, customized "form" letters, plain english commands, typeahead/print-ahead buffers for speed, a professional manual, & customer support from the author himself.

| | |
|---|---|
| Newscript | $104.88 |
| Options (below) are not available separately | |
| Mail Labels | $15.00 |
| File Convertor | $15.00 |
| Proportional Driver | $49.88 |
| Electric Webster | $150.00 |

## Orders & Information: CALL (603)-673-8857

**Orders Only: CALL (800)-343-0726**

## No Hidden Charges

No surcharge for credit cards—No charge for UPS shipping—Stock shipments next day. All equipment shipped factory fresh with manufacturer's warranty—We accept CODs. Open PO's not accepted—$50 minimum order—No foreign orders accepted

Prices subject to change—call for quotes

VISA MasterCard

HIGH TECHNOLOGY AT AFFORDABLE PRICES

**THE BOTTOM LINE**

Milford NH 03055-0423

# Bit Smitten—Part V

by Jay Chidsey

**This month Jay Chidsey shows you how to gain access to string data for use in programs which require random selection of words.**

This month I will discuss the array capabilities of TRS-80 Level II Basic with particular attention to gaining random access to string data for use in something like a poetry generating program. Many game programs require similar random selection of words or phrases out of a large number of words contained in data statements.

Arrays are very easy to use, once you understand what they are and how they work.

An array is an ordered series or arrangement of alphanumeric (alphabetic and numeric) items presented in a logically related and useful pattern.

The simplest kind of array is one dimensional; it's a list of items. A one-dimensional array is called a vector. A list can be as long as you want and have need for (and have memory space to accommodate), and it can be made up of either words or numbers.

An array can also be multi-dimensional. If, for example, you were the franchise holder on five McDonald's or Mister Donut or Frostee Freeze places, you might want to keep a record of 12 different categories of expenses for each of the five locations. Such a two-dimensional (or any multi-dimensional) representation is called a matrix or a table.

If you kept such records every day, then the day of the month could be a third dimension, the month (1–12) might be a fourth, and the year (82, 83 and so on) might be a fifth dimension. This would form a five-dimensional matrix in which each expense item entry was uniquely tagged, and recoverable from your data files through appropriate programmed inquiry. Item A(4,2,17,9,2), for example, would be the fourth expense item listed for location two on the 17th day of the ninth month (September) in 1982.

Level II Basic's array capability makes it possible to take information out of dead storage and put it into dynamic and usable memory. The information put into an array might come from input statements contained in your program, but most commonly it will come from data statements. I will deal only with data storage in this month's column. I will show you how to get words out of storage in data lines and into memory for manipulation and use. I'll use a scaled-down version of my poetry generating program (see the listing) as an example.

It's useful to think of the data in your TRS-80 as a huge high stack of information, with the first data entry as the item on the very top of that stack, and the last item in the last data line as the one on the very bottom of the stack. When you read that data into memory, the item at the top of the stack is read first, and assigned to the variable you name following the Read command. In this case, the variables to be subscripted are N, V, and A; N(1), N(2) and so on.

If you want to read down only so far, or to find a particular item and stop there, you can then insert the Restore command into your program, and the next Read will begin at the very top of the stack again. You can't start over part way down the stack, but you can start again at the top and peel off the data you want to discard quite simply.

In this poetry program I want to read three categories of words: nouns, verbs and adjectives. I will then randomly select among those words to construct poems.

**Notes on the Program**

In a program that uses words rather than numbers in its data lines, you must first clear space for string variables. This is done at line 90 with a CLEAR 10000. If no number of bytes are cleared by such a statement, the TRS-80 defaults to an automatic Clear of 50 bytes. If you are hard pressed for memory in a program and are using no string variables, you can free 50 additional bytes of memory by executing a CLEAR 0 at the beginning of your program.

There are four ways to use clear. The CLS (Clear Screen) command in a program and the clear key on your keyboard wipe the monitor screen of all content. Neither affect variables in storage, but both terminate the double-width print mode initiated by CHR$(23). The clear command used in a program by itself, with no number following, wipes out the content of all variables returning them to zero for numeric variables and to null (" ") for all string variables. Clear followed by a number reserves that number of bytes of string space for your program and zeros all variables. If you are using many strings, you should CLEAR n bytes early in a program, before any variables are introduced. Unless you do so, you will face the OS (Out of String Space) error message.

Clearing 10000 bytes will accommodate nearly 250 each of nouns, verbs, and adjectives. Since the program itself occupies little space, you need not be

*The Key Box*

**Cassette Basic**
**Model I or III**
**16K RAM**

overly concerned about reserving a large amount of string space in this program.

Lines 100–150 count the words in each data category, first nouns, then verbs, and then adjectives. Nouns are stored in data lines 2000–2998, verbs in lines 3000–3998, and adjectives in lines 4000–4998. The data entries of End at lines 2999, 3999, and 4999 halt the counting of nouns and switch to the counting of verbs and then to the counting of adjectives. Line 160, by subtracting the Ends from each count of words, places the actual number entered by the user in the variables N, V, and A. Line 170 sets the string variables N$, V$, and A$ to null, or blank. This is good programming practice unless you are hard pressed for memory space.

Line 180 dimensions the arrays we are about to create. The Basic operator is DIM. TRS-80 Level II Basic's default value for any array is 11 (0–10), so the dimension command must be used only if and when, as in this case, a larger number of elements are to be anticipated. How many of each? Line 160 has given us that count. We dimension N$ to N, V$ to V, and A$ to A.

Line 190 restores the data stack for reading again. The first time we went through it, in lines 100–150, we were merely counting the words. Now we must restore the stack so we can actually read it, word by word, into usable memory. Line 210 presents screen material to entertain the user while the data is being read into arrays N$(x), V$(x), and A$(x) by the read function.

Lines 240–280 are the heart of this program. A$ was one word (or sequence of words) when we were counting and replacing previous content with new content at each count, but not now! This is where we spin data into one-dimensional arrays. N$ is as many different words as you have dimensioned for and have in data to put into the prepared slots. N$(1) will be the first noun in your data, N$(2) the second, and so on up to N$(150) or N$(306) or whatever.

You could read these nouns into memory by programming something like READ N$(1): READ N$(2): READ N$(3) and so on. This takes lots of typing and lots of memory space. Lines 240, 260, and 280 spin those same words into memory, using For. . . Next loops, in a few seconds. At each pass through the For. . . Next loop they are incremented by one, going up from one to whatever total value N and V and A hold. Each pass through the loop, therefore, produces a different subscripted variable—N$(1), N$(2), and so on.

Lines 250 and 270 read and eliminate the End entries that separate nouns, verbs and adjectives. Now we have all the data in usable memory. You can check this by inserting a line:

```
285 FOR Q=1 TO N: PRINT N$(Q): NEXT
  : FOR Q=1 TO V: PRINT V$(Q): NEXT
  : FOR Q=1 TO A: PRINT A$(Q): NEXT
```

Your nouns, then your verbs, and then your adjectives will cascade down the monitor screen, and then stream up as the bottom line fills and bumps them upward. Remember to cancel line 285 by typing in 285 and hitting enter before you run the program.

Line 500 first sends us to the random number generator planted at line 5000, and then to a poem format (there are more than a dozen in the actual program) to create a poem. The random number generator makes it likely that each poem will be different. The more words in data storage, the more likely it is that each poem will be different. Line 1999 provides an easy way to punch up another poem; you simply hit enter.

To use this program, enter as many words as you wish beginning on line 2000 for nouns, line 3000 for verbs, and line 4000 for adjectives. Each new line must begin with data and each word must be separated by a comma from the next. If you wish, write your own additional poem formats, using the listing's format in lines 500 and 510 as an example. If you do this, change the GOTO in line 1999 to GOTO 490 and add:

```
490 F = RND(6): ON F GOTO 500, 550, 600, 650
  700, 750
```

For eight formats you would substitute RND(8) and add 800, 850 to line 490. Place a GOTO 1999 at the end of each format. ■

```
10 CLS: PRINT @276, "POETRY GENERATING PROGRAM"
20 PRINT @344, "(MINIMUM VERSION)"
30 PRINT @601, "BY JAY CHIDSEY"
40 PRINT @725, "GREEN SPRINGS OH 44836"
50 PRINT @980, "";: INPUT "PRESS ENTER TO CONTINUE"; YY
90 CLS: CLEAR 10000
100 READ N$: N=N+1
110 IF N$="END" THEN 120 ELSE 100
120 READ V$: V=V+1
130 IF V$="END" THEN 140 ELSE 120
140 READ A$: A=A+1
150 IF A$="END" THEN 160 ELSE 140
160 N=N-1: V=V-1: A=A-1: REM  SETS N,V,A TO ACTUAL COUNT
170 N$="": V$="": A$="": REM  SETS STRINGS TO BLANK
180 DIM N$(N), V$(V), A$(A)
190 RESTORE: REM  DATA NOW READY TO BE REREAD FOR CONTENT
200 : PRINT: PRINT TAB(18) "POETRY GENERATING PROGRAM"
210 PRINT:PRINT:PRINT
    : PRINT"WORDS IN STORAGE:": PRINT: PRINT
    : PRINT"NOUNS............"N
    : PRINT"VERBS............"V
    : PRINT"ADJECTIVES......."A
240 FOR X=1 TO N: READ N$(X): NEXT
250 READ EN$:      REM  PEELS OFF 'END'
260 FOR Y=1 TO V: READ V$(Y): NEXT
270 READ EN$
280 FOR Z=1 TO A: READ A$(Z): NEXT
290 PRINT: PRINT: INPUT "PRESS >ENTER< TO CONTINUE"; YY
500 CLS: GOSUB 5000: PRINT: PRINT
    : PRINT TAB(25) "THE " N$(X): PRINT: PRINT
    : PRINT, "HOW " A$(Z) "LY " V$(Y) "S THE " N$(X) "!"
    : PRINT, "" N$(X) "S SELDOM " V$(Y) "."
510 GOSUB 5000: PRINT
    : PRINT, "MORE "A$(Z) " ARE THE " N$(X) "S,"
    : PRINT, "AS THEY " V$(Y) "."
1999 PRINT: PRINT: PRINT
    : PRINT "PRESS ENTER FOR ANOTHER POEM"
    : INPUT "PRESS BREAK KEY TO EXIT PROGRAM"; YY
    : GOTO 500
2000 DATA STONE, RING, CLOUD, BARN, DOG, CAT, MOUNTAIN, RIVER
2999 DATA END
3000 DATA RUN, JUMP, FLY, SEE, PLAY, THROW, SING, SHOUT, CLIMB
3999 DATA END
4000 DATA BEAUTIFUL, SORROWFUL, GOOD, SCENIC, PURPLE, SWEET
4999 DATA END
5000 RANDOM: X = RND(N): Y = RND(V): Z = RND(A): RETURN
```

*Program Listing*

*Writer and businessman Chidsey uses his Model III in both activities. He can be reached at 205 E. Adams St., Green Springs, OH 44836.*

# QWERTY

# DAISYWHEEL II

At last! QWERTY 3.0 is available for the DAISY WHEEL II! Finally, owners of the Daisy Wheel II can make their printers do what they were **designed** to do. Produce scores of special symbols and accents! Add footnotes and borders to text! Bold print, double underline, even underline spaces! Print superscripts and subscripts, simultaneously! Print a single page of text without resorting to copy-markers! Stop printing to change wheels or insert text!

Once you use DAISY, you'll wonder how you worked without it. We **guarantee** you will agree! If not, return DAISY within fourteen days for a prompt refund. That's how confident we are!

**DAISY FEATURES**

Easy transition between the three print pitches. One command changes pitch and resets formats for optimum appearance in that pitch.

Over 100 new symbols, including foreign language accents, Greek letters, and mathematical symbols, all using the standard Courier wheel. Allows tildes, carats, and overbars to be placed over any character.

Any character can be used as a superscript or subscript, singly or simultaneously. Extra high superscripts for integrals. Fine space controls to make superscripts and subscripts look perfect.

Underlining, double underlining, and bold printing, with our without underlining of spaces.

Special provision for printing ratios of one long expression over another (as in algebra).

Provision for putting FOOTNOTES on a page in such a way that they remain on that page despite later editing.

TABLE and MABLE commands enable exact positioning of the print head anywhere on the line. Invaluable in printing neat data tables, columnar material, etc.

PRETTY command enables one to print repetitions of a chosen character; combined with TABLE or MABLE, it becomes easy to produce attractive borders, even in headers and footers.

PAGE END allows the user to discover where pages will end, when they are printed, without actually printing.

Special commands allow the printing of a single page anywhere in the text, with correct page number, footer, and header, without the use of copy-markers. A special command will reset the default format parameters, to allow use of copymarkers without having to insert a format line to adjust for format changes.

FOLIO format enables one to produce print-outs in two or three columns per page, as for newsletters, indices, etc. FOOTNOTES, footers, headers, and PAGE END will work correctly.

Access to disk directory and ability to kill files from within DAISY, without losing any text.

Special code in the text will stop printing until user restarts it; use it to change print wheels. One can insert up to 60 characters of text at this point, as for form letters.

Special print-outs allow one to keep records of DAISY commands used to create a certain effect.

Flaws in SCRIPTSIT's widow supress are corrected. SCRIPSIT's problem with typing two spaces after a period is fixed. SCRIPSIT's scrolling speed is doubled, with instructions for user modification.

CAUTIOUS BUYERS: We invite you to buy the DAISY manual. This is without a doubt one of the clearest, most well-written manuals ever produced for **any** piece of software. Over 110 pages. **Indexed.** In a custom 3-ring binder. We will credit the full $10.00 price when you decide to buy DAISY.

DAISY REQUIRES: A TRS-80 Model I or Model III with 48K, at least one disk drive, and a copy of **Model I SCRIPSIT.** Please specify which machine and how many drives in your system.

QWERTY 3.0 is also available for the Centronics 737, 739, and Lineprinter IV.

| | |
|---|---|
| **DAISY complete package.........** | **$74.95** |
| **DAISY manual alone..............** | **10.00** |
| **QWERTY 3.0 complete package.....** | **74.95** |
| **QWERTY 3.0 manual alone.........** | **10.00** |

Please add $2.00 for shipping and handling. Add $4.00 for first class postage or UPS. Overseas orders add $4.00 for surface mail, $12.00 for air mail.

**MED SYSTEMS SOFTWARE**

P.O. BOX 3558 CHAPEL HILL, NC 27514

TO ORDER, CALL 1-800-334-5470

✓83

# Up Your Mod I

by Bruce Powel Douglass

**Percom Doubler II**
**Percom Data Corp.**
**11220 Pagemill Road**
**Dallas, TX 75243**
**$169.95**

**LN Doubler 5/8**
**LNW Research Corp.**
**2620 Walnut**
**Tustin, CA 92680**
**$219.95**

**MULTIDOS**
**Cosmopolitan Electronics Corp.**
**Box 234**
**Plymouth, MI 48170**
**$79.95**

**NEWDOS80 Version 2.0**
**Apparat Inc.**
**4401 S. Tamarac Way**
**Denver, CO 80237**
**$149**

**DOSPLUS 3.4**
**Micro-System Software**
**5846 Funston St.**
**Hollywood, FL 33023**
**$149.95**

**LDOS 5.1**
**Logical Systems Inc.**
**11520 North Port Washington Road**
**Mequon, WI 53092**
**$129**

**Percom TFD 100-1**
**Percom Data Corp.**
**11220 Pagemill Rd.**
**Dallas, TX 75243**
**$305**

**PMC SFD-51-A**
**Personal Microcomputers, Inc.**
**475 Ellis St.**
**Mt. View, CA 94043**
**$355**

**Double density disk storage—a close look at all you will ever need to upgrade your Model I.**

Model I users have always been plagued by a problem called data separation, when the disk controller has trouble separating data from other bit streams of information, such as the clock cycle. This problem can become severe particularly when you are formatting or writing to the innermost tracks of a disk. With double density (since the current double-density disk controllers all use a built-in data separator), disk I/O is actually more reliable than single density. My old drives reported fewer disk read/write errors from single-density I/O with the Radio Shack disk controller when I moved up to double density. You can store more information on a double-density disk as well—184,000 bytes on a formatted disk. With a four-drive system, single-density users are limited to a meager 332,000 bytes of on-line storage, whereas double-density users can have up to 700,000 bytes on-line.

How do you get double density? Which disk controller and DOS should you choose? There are a wide variety. I contacted many manufacturers of double-density products, and those that responded are represented here.

Two double-density disk controllers, five DOSes, and four disk drives are reviewed. Although most of this information is applicable to Model I users only, four of the DOSes have Model III versions.

Indispensable in my testing of the disk drives and disk controllers was the Floppy Doctor, a product of the Micro Clinic in Fountain Valley, CA. The utility comes on a self-booting disk and has a memory test and nine disk-drive tests.

### The Disk Controller

The disk controller is the circuit in the expansion interface that lets the computer talk to the disk drive. The Western Digital disk controller that accompanies the Radio Shack Expansion Interface is a single-density disk controller, but it can write several kinds of DAMs (data address marks). To allow compatibility, both the Percom and LNW disk controllers allow your computer to use the original disk controller as well as the double-density disk controller.

You install both in the same manner. First, dismantle the Expansion Interface and remove the disk-controller chip from its socket. Next, plug the old disk controller onto your new disk-controller circuit board (which houses an additional, double-density disk controller) and plug the new circuit board into the disk-controller slot. It will take about 5–15 minutes to install, depending on your familiarity with the parts involved. It requires no soldering, trace cutting, or any modification to existing circuits.

I reviewed the Percom Doubler *II* and the LN Doubler 5/8. The Percom Doubler comes with easy-to-follow in-



Circle 399 on Reader Service card.→


PANDEMONIUM
By Norman J. Wazaney Jr.
A word game for our time.
At last… a computer word game that entertains, challenges, educates! Pandemonium is thoroughly fascinating, stimulating, and highly addictive.
Features include a built-in 6000 word dictionary, scoring display and a player selectable clock.
Play it alone! Play it with your kids! Play it at a party!
The word is out… Pandemonium is in. Available now for only $39.95 at computer stores.
To order by phone, call 800-526-9042 and use your Visa or MasterCard. All shipments made the same day in which orders are received. To order by mail, add $1 for shipping charges and send your check to:
Soft images
DIVISION OF DECISION SYSTEMS INC
200 Route 17, Mahwah, NJ 07430.
TRS-80 MOD I/III 48K TRS-DOS
TRS-80™ Radio Shack/Tandy Corp.
OUR SOFTWARE IS UNPROTECTED PERMITTING CONVENIENT BACK-UP.
PANDEMONIUM

structions and Percom's own DBLDOS; essentially a TRSDOS patched to run double density with a few corrected flaws. It sells for $169.

The LN Doubler 5/8 comes with better instructions and an even easier installation manual. The manual includes the Western Digital technical manual for their disk controllers. The LN Doubler 5/8 also supports eight-inch floppy drives (if you have the necessary speedup enhancement to your Model I), and comes with DOSPLUS, a much better disk operating system. The LN Doubler 5/8 sells for $220.

Are they different in performance? No. I was unable to demonstrate a significant difference in any disk-controller functions between the Percom Doubler II and the LN Doubler 5/8. Reliability testing showed no significant differences in granules locked out or in read/write errors. This was using the brand new drives I received for this review and older Pertec drives. New Maxell disks were used and were formatted to 40 tracks using double density.

### The DOSes

A disk operating system is a program that controls computer operation in a disk-based computer system. The aspects of a DOS that I consider important are: ease of use; speed of operation; accompanying utilities, such as editor/assemblers, file utilities, debuggers, monitors, and languages; readability and completeness of documentation; compatibility with other disk formats; and cost.

The DOSes reviewed here all support double density and are available for the Model III. They are MULTIDOS, NEWDOS80 2.0, DOSPLUS, and LDOS.

### MULTIDOS

MULTIDOS, written by Vernon Hester, is the most inexpensive of these DOSes and has many features its more expensive competitors don't have. It is the only totally compatible DOS around. It will read any of the other disk formats, including DBLDOS (none of the others will). It is also a "no-hang" DOS, meaning that even if you remove the operating system from drive 0 and the DOS tries to access the system utilities on the DOS disk, it won't up and quit, as will the other DOSes. It won't do what you wanted, but it will remain in control of your computer. One useful, although perhaps minor, convenience is that pressing the enter key without input will re-execute the last DOS command. MULTIDOS is a fast DOS as well.

The only major problem with MULTIDOS is that it only checks four times on a read/write. My old Pertecs show many more "Data Record Not Found" errors with MULTIDOS than the other DOSes, which check 10 times before returning an error.

The manual is a terse 66 pages (not including a few pages of errata and additional information). It lacks an index, but covers the features of MULTIDOS adequately, provided that you already have the TRSDOS manual. This complaint is true of NEWDOS80 as well. DOSPLUS and LDOS have manuals that don't require that you have read the TRSDOS manual to understand them.

MULTIDOS has many useful library commands besides those provided in TRSDOS. You can disable the break key from DOS, and build and execute DO files (although you cannot run a Basic program and pass input to the Basic program from the DO file). Some of MULTIDOS's commands include: Forms, which lets you specify lines per page and line length for printed output; Hash, which returns the hash code of a file ID; and KEYBRD, which sets the keyboard parameters. MULTIDOS has a default keyboard driver that provides a repeating, flashing block cursor, but this can be changed with the KEYBRD command. Link and Route control the I/O of data to the video and printer. You can send everything that goes to the printer to the video too (or instead) and vice versa. Topmem allows you to set the top-of-memory pointer from DOS.

MULTIDOS has a number of useful system utilities. Backup lets you make speedy backups on a single drive and alter the number of tracks on it at the same time. Copy allows single-drive copying of single files, including to and from non-MULTIDOS disk formats. Format formats disks in single, double, or P (DBLDOS) density formats.

Other utilities include a disk-based version of Radio Shack's Editor/Assembler, a RAM Scanner to locate a one or two-byte word in memory, a graphics utility to allow direct keyboard entry of graphics characters, a printer spooler, and best of all, MULTIDOS's Versatile File Utility (VFU).

VFU is a utility for frequently needed disk operations, including purging files, printing a disk directory, multiplying file copies, and executing menu-based programs. The Purge option gives a screen of the available files, as do the Copy and Execute commands.

MULTIDOS has the best Basic around, SuperBasic. It also has a programming and debugging Basic called BBasic with Boss built into it. SuperBasic is the smallest Basic available since it uses extensive overlays for various operations—SuperBasic leaves 40,000 bytes free after loading. Basic can be loaded directly, recovered (with BASIC*), and it provides two more options: BASIC! loads Basic and recovers a program left there by a non-MULTIDOS Basic. This means you can recover a Basic program left from running DOSPLUS or NEWDOS80 and recover it by inserting a MULTIDOS disk in drive 0, pressing the reset key, and typing BASIC!. BASIC# recovers a Level II program in memory. The recover utilities allow direct continuation of a program. SuperBasic provides a special form of Level II to provide even more memory when needed.

BBasic incorporates Boss into Basic, and allows single stepping of Basic programs, reviewing the current values of any set of variables at any time, breakpoints, and an improved trace function. Other than this, BBasic is just like SuperBasic, except it takes up more room.

SuperBasic allows you to use single-key abbreviations for some commands. It also has a Pn command to display a requested page of a program. Several additions to Basic are provided with CMD"xxx" commands. CMD "C" compresses your program without eliminating remark statements. CMD"E" displays an error message associated with the last disk error. CMD"K" zeros all elements of an array. CMD"L" deletes an array. CMD"M" moves a program line. CMD"N" duplicates a program line. CMD"O" opens an additional file buffer. CMD"Q" performs a string sort of a single or double-dimension string array, and CMD"V" displays all current scalar variables and their values.

SuperBasic has overlay utilities to renumber parts of your Basic program, search for ASCII strings within program text, change all or part of variable names, change graphics codes (CHR$(X)") into packed strings and it provides a cross-reference utility that shows all line numbers that contain specified variables. SuperBasic uses Name to chain programs without losing variables. I think SuperBasic is the

# DISCOUNT

A LITTLE BIT OF "**GREEN**" GOE$ A LONG WAY!!!

**FREE SHIPPING** on all prepaid cash orders over $50.

## COMPARE OUR PRICES...Call (216) 481-1600

## DISKIT III

### MODEL III DISK UPGRADE

Features:

- Uses Micro Main Frame Finest Disk Controller
- Gold Plated Edge Connectors
- Switching Power Supply
- Supports 5" or 8" Drives
- 40/80 Track Supported
- Single/Dual Head Supported
- Metal Disk Drive Brackets
- All Hardware and Cables for Two Disk Drives
- 1 Hour or Less for Installation
- 100% Compatible
- No Soldering Needed
- 180 Days Warranty on Controller

| | |
|---|---|
| DISKIT III W/O Drives | $279.00 |
| DISKIT III W/ONE Tandon 100-1 40 Track Drive | $479.00 |
| DISKIT III W/TWO Tandon 100-1 40 Track Drive | $699.00 |

DEALER INQUIRIES WELCOME

DISKIT III is a trademark of MDS

### PRINTERS

#### NEW LOWER PRICES

| | |
|---|---|
| Epson MX-80 | $CALL |
| Epson MX-80FT w/Graftrax | $CALL |
| Epson MX-100 w/Graftrax | $CALL |
| IDS Prism 80 | $CALL |
| IDS Prism 132 | $CALL |
| Okidata Microline 80 | $CALL |
| Okidata Microline 82A | $CALL |
| Okidata Microline 83A | $CALL |
| Okidata Microline 84 | $CALL |
| Prowriter 8510 | $CALL |
| Smith Corona TP-1 | $629.00 |

### TRS-80, HEATH, IBM, APPLE, ZENITH DISK DRIVES

#### AT OR BELOW DEALER'S COST!

TRAXX-Model I & III Complete

| | |
|---|---|
| Tandon 40 Track | $279.00 |
| Tandon 80 Track | $CALL |

* * * * * SPECIAL * * * * * *

Tandon 80/80 Track ............... $429.00

* * * * * * * * * * * * * * *

BARE TANDON

| | |
|---|---|
| T100-1 40 Track | $209.00 |
| T100-2 40/40 Track | $CALL |
| T100-3 80 Track | $CALL |

* * * * * SPECIAL * * * * * *

T100-4 80/80 Track ............... $359.00

* * * * * * * * * * * * * * *

TRAXX 8 INCH MODELS

| | |
|---|---|
| Seaman 77 Track s/s | $339.00 |
| NEW TANDON "THIN LINE" 8 INCH | |
| 848-1 Single Side | $399.00 |
| 848-2 Dual Side | $499.00 |
| APPLE Add-On Disk Drive | $359.00 |

### MODEMS

| | |
|---|---|
| UDS 103 LP Direct Connect | $175.00 |
| UDS 103 JLP AUTO ANS | $209.00 |
| Hayes Micromodem II (APPLE) | $299.00 |
| Hayes 100 Model (S-100) | $325.00 |
| Hayes Smart Modem (RS-232) | $249.00 |

### MICROBUFFER

| | |
|---|---|
| MBP-16K Parallel | $149.00 |
| MBS-8K Serial | $149.00 |
| INTERFACE CARDS | |
| 8141 (RS-232) | $75.00 |
| 8150 (2K Buffer RS-232) | $150.00 |
| 8161 (IEEE 488) | $55.00 |
| 8131 (Apple Card) | $85.00 |
| 8230 (Apple Cable) | $25.00 |
| 8220 (TRS-80 Cable) | $25.00 |

### DISKETTES

| | |
|---|---|
| PARAGON MAGNETICS GOLD, Soft-Sectored, Single-Sided, Double-Density, 5¼ inch diskettes with reinforcing HUB-RINGS | $23.95 |
| VERBATIM-Soft-Sectored Diskettes | |
| 5¼" 1S/DDen (MD525-01) | $26.95 |
| 5¼" 2S/DDen (MD550-01) | $39.95 |
| 5¼" 2S/4Den (MD557-01) | $51.50 |
| 8" 1S/DDen (FD34-8000) | $43.95 |
| VERBATIM-Hard-Sectored Diskettes | |
| 5¼" 1S/DDen 10-Sector (MD525-10) | $26.95 |
| 5¼" 2S/DDen 10-Sector (MD550-10) | $39.95 |
| 5¼" 2S/4Den 10-Sector (MD557-10) | $51.50 |

### SUPPLIES

| | |
|---|---|
| HUB RING KIT for 5¼" disks | $10.95 |
| HUB RING KIT for 8" disks | $12.95 |
| REFILLS (50 Hub Rings) | $5.95 |
| CLEANING KIT for 5¼" drives | $24.95 |
| 5¼" Diskette Case | $3.50 |
| 8" Diskette Case | $3.95 |
| 5¼" File Box for 50 diskettes | $24.95 |
| AVERY TABULABLES | |
| 1,000 3½" x 15/16 | $5.49 |
| 3,000 3½ x 15/16 | 10.95 |
| 5,000 3½ x 15/16 | $15.95 |
| FAN FOLD PAPER (Prices F.O.B. S.P.) | |
| 9½ x 11 18 lb. WHITE 3,000 ct | $27.95 |
| 14⅞ x 11 18 lb. WHITE 3,000 ct | $37.95 |

### BOOKS

OTHER MYSTERIES

| | |
|---|---|
| TRS-80 DISK | $19.95 |
| Microsoft Basic Decoded | $24.95 |
| The Custom TRS-80 | $28.95 |
| Basic Faster & Better | $29.95 |
| The Custom Apple | $28.95 |
| 1001 Things To Do With Your Personal Computer | $10.95 |
| Choosing A Word Processor | $15.95 |
| CP/M Primer | $15.95 |
| I Speak Basic TRS-80 | $12.95 |

### TRS-80

| | |
|---|---|
| Model III w/2 Tandon 48K | $1795.00 |
| Model III w/1 Tandon 48K | $1599.00 |

Both units come with 120 day warranty from MDS and use our MEMORY and DISKIT

### LNW

| | |
|---|---|
| LNW-80 Computer | $1595.00 |
| LNW-80-II Computer w/cpm 96K | $1995.00 |
| SYSTEM EXPANSION II | $349.00 |
| LNDOUBLER 5/8 Board | $149.00 |

### SOFTWARE

TRS-80 Model I & III

* * * * * SPECIAL * * * * * *

FREE Floppy Doctor with the purchase of any DISKIT III ............. $30.00 VALUE!

* * * * * * * * * * * * * * *

| | |
|---|---|
| NEWDOS 80 2.0 | $139.95 |
| DOSPLUS 3.4 Model I | $89.00 |
| DOSPLUS 3.3 Model I | $69.95 |
| *NEW* Electric Pencil | $79.95 |
| ELECTRIC WEBSTER | |
| 50,000 Word Spelling Checker | $79.95 |
| Correcting Feature | $59.95 |
| When purchased together | $129.95 |
| UNITERM Terminal | $79.95 |
| UNITERM/80 Terminal | $89.95 |
| ACE MAIL for Hayes Smart Modem | $69.95 |
| LABELMAKER for MX80 | $19.50 |
| AIDS III Data Management | $49.95 |
| Maxi Manager | $89.00 |
| Floppy Doctor I, III | $24.95 |
| Inventory + | $29.95 |
| Cash Register/80 | $29.95 |

### GAMES

TRS-80 • **SAVE** • MOST GAMES AVAILABLE FOR **APPLE**

| | | |
|---|---|---|
| 141-001 | Introductory 3-Pack Rescue, Morloc's and Datestones | DISK .. $49.95 |
| 221-001 | Super Nova | DISK .. $17.95 |
| 221-002 | Galaxy Invasion | DISK .. $17.95 |
| 221-003 | Attack Force | DISK .. $17.95 |
| 221-004 | Cosmic Fighter | DISK .. $17.95 |
| 221-006 | Robot Attack | DISK .. $17.95 |
| 221-008 | Stellar Escort | DISK .. $17.95 |
| 351-001 | Scarfman (Mod. I/III) | DISK .. $17.95 |
| 351-500 | Scarfman (color) | CASS . $17.95 |
| 351-501 | New Bounceoids | DISK .. $19.95 |
| 391-001 | Math-Pak-1 | CASS . $14.95 |
| 391-002 | Math-Pak-2 | CASS . $14.95 |
| 431-001 | Sargon II (Mod. I) | DISK .. $29.95 |
| 431-101 | Sargon II (Mod. III) | DISK .. $34.95 |
| 511-003 | Adventure | DISK .. $29.95 |
| 021-004 | Pigskin | DISK .. $17.95 |
| 060-601 | Adventure #1,2&3 | DISK .. $34.95 |
| 060-604 | Adventure #4,5&6 | DISK .. $34.95 |
| 060-607 | Adventure #7,8&9 | DISK .. $34.95 |
| 060-611 | Adventure #10,11&12 | DISK .. $34.95 |
| 061-066 | The Mean Checker Machine | DISK .. $17.95 |

**MDS MICRO DATA SUPPLIES**

22295 EUCLID AVE.
EUCLID, OHIO 44117
(216) 481-1600

459

WE ACCEPT
- Visa
- MasterCard
- Checks
- Money Order
- C.O.D.

ALL PRICES ARE FOR MAIL ORDER ONLY. Prices, Specifications and Offerings subject to change without notice.

ADD $3.00 FOR SHIPPING & HANDLING. $6.00 Extra for C.O.D. Orders. Ohio Residents add 6.5% Sales Tax best Basic to accompany any DOS available.

**NEWDOS80 2.0**

NEWDOS80 2.0 is a double-density DOS that writes and reads a specific disk format differently from the other DOSes. The familiar gran for granule has been replaced by lumps. What's a lump? NEWDOS80's documentation doesn't go out of its way to tell you. It is definable with the PDRIVE GPL parameter, so that there are between two and eight granules per lump. You can define the granules-per-lump (GPL) parameter as eight. Once you do so, the DOS's features are about the same as in the others.

NEWDOS80 2.0 will read other formats, but they must be configured with the PDRIVE (similar to MULTIDOS's CONFIG) each time a different disk is used, particularly if the disks are in different densities. To get around this problem, The Alternate Source sells a program that gives NEWDOS80 2.0 automatic density recognition. This program is called DDSD and costs $19.95.

One of the examples in the PDRIVE section implied that I could read the directory of a Model III disk. I was unable to read the Model III disk's directory with NEWDOS80 2.0. You can, however, copy from a Model III TRSDOS disk on a Model I with this DOS.

It will take you longer to get started with NEWDOS80 than with the other operating systems presented here. This is partially due to the manual's technical approach, and partially to the approach taken by the DOS in its performance of the various functions. All the other DOSes allow you to back up a disk with the familiar Backup command, but NEWDOS80 requires using the Copy format. Once learned, NEWDOS80's approach is quite useable and powerful.

NEWDOS80 has a large number of library commands. The most notable are:

- Break—enables and disables the break key;
- Chain and DO—shifts to the disk file for command input;
- JKL—sends the current screen contents to the printer;
- HIMEM—sets the top of memory from DOS;
- LC and LCDR—toggles the lowercase driver (Model I only);
- Pause—displays the message and pause until enter (for use in DO files);
- PDRIVE—assigns the default attributes to a physical disk drive;
- Print—outputs a text file to the printer;
- R—repeats the last DOS command; and
- Route—routes one device to another.

NEWDOS80 also sports a miniDOS not found in the other DOSes. The miniDOS allows the execution of DOS library commands except Append, Chain, Copy, or Format. Since it is interrupt driven, it can be used from almost any program that keeps interrupts active and uses the NEWDOS80 keyboard driver.

The utilities provided by NEWDOS80 include the ever-popular Superzap. It also supplies a patched version of Radio Shack's Editor/Assembler and a disassembler. The disassembler writes disassembled code to disk from where it is loaded by the patched assembler. It will also send the disassembly to the printer or screen and display a cross-reference table.

LMOffset is a program used primarily for loading system tapes that overlay the DOS and store them on disk. Chainbld is used for the creation and editing of Chain or DO files. ASpool is a printer spooler.

NEWDOS80 also sports an enhanced Basic. Enhancements include toggling the break key, abbreviated line listing and editing commands similar to MULTIDOS's SuperBasic. Chaining is allowed and the Merge command merges non-ASCII files. REF displays a reference list of variables and line numbers, and can specify different outputs in a number of ways.

CMD"F=ERASE" and CMD"F=KEEP" allow selective clearing of Basic variables, like the MULTIDOS SuperBasic CMD"L", only more powerful. The first format selectively clears a number of variables in a single statement, while the second format clears all except the declared variables. Some other CMD enhancements include clearing all Returns and For...Next loops from the stack, dynamic deletion of text lines, swapping of variable contents, single-stepping Basic, recovering a deleted program (providing it hasn't been overwritten in memory), and sorting arrays. CMD"O" sorts a multidimensioned array.

NEWDOS80 has some extensive disk I/O enhancements to Basic as well. The manual spends 20 pages discussing these disk I/O enhancements in chapter 8 alone; appendices A and B feature 65 more pages on the topic.

NEWDOS80 handles the standard sequential (which they call print/input) and random (which they call field item). It also handles marked item (which has three subtypes) and fixed item (which has two subtypes). Random-access files can be of 1-256 bytes in fixed length. The fixed time and marked item files have a maximum record length of 4,095 bytes.

Sequential files can only be accessed sequentially. That is, if you want to read item number 200, you must wade through the first 199 items. Random-access files can access record number 200 directly and are more useful and powerful.

In fixed and marked item files, Get and Put are used to access data in the I/O buffers. The type and length of each item is determined by the IGEL (item group expression list) of the Get and Put for fixed-item files, but in marked-item files, the item always begins with marker bytes, specifying the format of the item. Although they are initially hard to understand, these new file types are quite useful.

NEWDOS80 is a good DOS. The Basic is not as good as MULTIDOS's, but it operates well. It has a disassembler, Superzap, and other powerful utilities. Apparat supplies zaps free to registered owners and has provided good customer support in the past. Unfortunately, it does not have automatic density recognition.

**DOSPLUS 3.4**

DOSPLUS comes with the LN Doubler 5/8. It has automatic density recognition, although it cannot read DBLDOS format.

The approach to operating system functions took a departure from TRSDOS with DOSPLUS. The DIR command automatically displays the information normally supplied with DIR (A) in TRSDOS, and a new command, CAT, displays the abbreviated directory normally retrieved with DIR in TRSDOS. It would be nice if the DIR command displayed the number of granules left.

DOSPLUS has an automatic repeating, lowercase keyboard driver and DO-file capability. The Format command works quickly and verifies the tracks backwards rather than forwards to save time (since the drive head is already at the highest track, it needn't move back to track 0). DOSPLUS Model I and III double-density versions are directly compatible, meaning that a Model I DOSPLUS disk can be read by a Model III DOSPLUS. This does not mean that a Model I DOSPLUS can read a Model III TRSDOS disk. DOSPLUS does come with a Convert command to transfer files from a Model III disk to a DOSPLUS disk, somewhat like NEWDOS80.

Although DOSPLUS does not have a VFU command similar to MULTIDOS, Picotrin Technology of Lantana, FL markets a DOSPLUS-compatible program that provides cursor-oriented functions.

The DOSPLUS manual is reasonably well done except for the omission of an index. The technical section is particularly good in how it describes the entry and exit points for the various DOS routines and DCB (device control blocks).

The library functions are similar to the DOSes previously described. The DO construct works well and lets you specify in the command the reserved high memory for the DO file.

The DOSPLUS utilities include a printer spooler, single-drive copy, Crunch (a program that compresses Basic programs saved on disk), Diskdump and Diskzap (floppy-disk editors similar to Superzap), Map (shows file locations on a disk), Restore (recovers killed files), Sysgen (creates nonstandard system disks), Tape (similar to NEWDOS80's LMOffset), and Transfer (for multiple file copies from one disk to another).

DOSPLUS enters Basic differently than MULTIDOS or NEWDOS80. The Basic is enhanced as well. TRON allows you to single-step Basic, and Tab tabs on a line printer beyond 64 columns. You can move and copy Basic program lines, and DOSPLUS Basic has a large number of abbreviates.

DOSPLUS also comes with a tiny Basic called TBasic. TBasic occupies slightly more space than MULTIDOS's SuperBasic, but has no file space allocated. TBasic lacks many of the enhancements found in regular DOSPLUS Basic, but is useful because it will still run the full set of Basic commands. TBasic displays abbreviated error messages in a further attempt to free up memory.

### LDOS 5.1

LDOS recently reduced its price from $169 to $129, due to the large volume of sales. Registered LDOS users receive a quarterly magazine called, appropriately enough, *The LDOS Quarterly*, support hard to match.

LDOS comes with a rather imposing 300-plus-page, well-written manual. The LDOS manual is the best of the DOSes reviewed here.

In order to make LDOS run with double density, you must use the PDUBL disk-driver program, which sits at the top of memory. This creates some real hassles with programs in high memory at times. You can use the Sysgen command to save the driver configuration so it automatically loads on boot. Model I users cannot boot up on a double-density LDOS disk—it must be single density, although you may change it to double density after the boot, a minor inconvenience.

LDOS on the Model I will not read a Model III TRSDOS disk, although the library command CONV can be used to transfer files from a Model III TRSDOS disk.

The library commands in LDOS are similar to those available in the other DOSes. Trace supports and displays the PC register in the upper right corner of the display, and Filter is similar to the Route command on other DOSes but better. It establishes a program to filter the I/O path of some device. Several filter programs are supplied with the LDOS disk.

MINIDOS/FLT is a filter similar to NEWDOS80's miniDOS. In it, you can toggle the clock on and off, enter Debug, display free space, kill files, and so on. A filter for the line printer is also provided.

One of the most powerful features of the LDOS operating system is its job control language (JCL). A DO or Chain option is available for all the other DOSes, but a true JCL is lacking. The LDOS JCL is actually a compiled language. It can handle conditional statements and support variables. LDOS allows concatenation of variables (which they refer to as tokens) and logic expressions using NOT, AND, and OR. LDOS JCL allows macros in its programs. This means that you can specify a macro name and some complex function will be performed.

The JCL feature is a powerful one, although it could be made more powerful by adding the ability to direct execution of the JCL program on the basis of the results of some program. An interactive control language (ICL) is available for NEWDOS80 version I and it allows this flexibility.

One of the utilities includes CMDFILE. Similar to NEWDOS80's LMOffset but more powerful, this program allows appending patches to machine code, appending machine-code programs together, saving them to disk or tape, moving them around in memory, and so on. CONV moves files from a Model III TRSDOS disk to an LDOS disk. Since this program can be used by a Model I running double density, this provides the only utility to move programs from a Model III disk to a Model I disk. LCOMM is a communications program that allows your TRS-80 to talk to other TRS-80 computers or to bulletin board systems, such as Forum 80. Repair updates and corrects information on certain types of disks to make them useable to LDOS. This utility is more useful for Model III users than for Model I folks.

LDOS's enhanced Basic, LBasic, has some abbreviations and allows single-stepping of Basic programs. Restore has been updated to allow you to specify a line number to start the Restore at. LBasic allows chaining of Basic programs without loss of variables. LBasic can also sort arrays and provide reference information on Basic programs. LBasic is good, but in my opinion, not as good as the Basics in the other operating systems.

LDOS is a good DOS, but there are a few minor inconveniences. The Basic could be more powerful, and it would be helpful if the top of memory was free. (PDUBL sits at the top of memory if you're using double density.) The manual is quite good, and the Filter and JCL features make this the most powerful microcomputer DOS around. Finally, the LDOS support is unprecedented in quality and ease for the registered user.

### In Summary

As you can see, all these DOSes have their good points. MULTIDOS has its low price tag, versatile file utility, no-hang ability, virtually total compatibility, and the excellent SuperBasic. The manual is a bit terse and MULTIDOS only checks a read/write four times.

NEWDOS80 has good utility programs like LMOffset, a disassembler, and Superzap. The manual, however, is lacking a tutorial and NEWDOS80 needs automatic density recognition. DOSPLUS has Diskdump, Diskzap and automatic density recognition. LDOS has its excellent manual, great customer support, very flexible filtering of I/O, and a powerful JCL. Which you choose depends very much on your applications.

### The Drives

A disk drive must be able to move a read/write head very precisely to any

✓ See List of Advertisers on Page 451

location and rotate the disk within precise limits so that address marks and data can be accurately located. The drive speed is supposed to be 300 rpm. Most drive manufacturers say that a one-percent variation in drive speed is tolerable, and often a precisely adjusted drive will still vary within ± one percent of 300 rpm.

Older minifloppy disk drives had 35 tracks and wrote only in single density. Technology has improved, and now 80 and even 96-track minifloppy drives are available in single or in double density.

I was hoping to find differences in reliability of the brand new drives under severe stress-testing conditions. As long as I kept below the upper limit of the speed of the stepping motor of the drive, I was unable, generally, to get *any* errors in these drives. I could find no clear-cut reliability differences among the various drives, so it was impossible to statistically investigate the presence or absence of an interaction of effects due to the disk controller used. I did, however, discover differences in their operation. The drives I reviewed are: Percom model TFD 100-1, PMC model SFD-51-A, Teac, and Trak model 5410-121.

The Percom is the only one of these drives that is flippy, meaning that you can remove the disk from the drive, flip it over, and reinsert it into the drive and write to the other side of the disk. This effectively doubles the amount of storage you have in your disk library. The Percom drive also has the drive-cable card extending out from the rear of the machine, so it is easily accessible without removing the case, a nice feature. It has a stepping-motor speed that allows you to seek tracks at 12 milliseconds. This is faster than the Teac (at 30 ms) and my Pertecs (at 20 ms), but slower than the PMC and the Trak (both at 5 ms). The list price of the Percom drive is $305.

The PMC drive also has the drive-cable card extending out the rear of the drive. It has a fast-stepping motor with 5-ms track seeking. It allows writing only to one side of the disk.

The Teac drive was the slowest of the bunch, requiring a minimum of 30 ms to seek a track. In order to change the drive cable, you must remove the cover. The card for the drive is inconveniently spaced, meaning the drives must be slightly closer than recommended by Radio Shack when using their four-drive cable. The Teac drive I used was loaned to me by American Business Computers. They give their Teac drives a one-year warranty. All the other disk drives were loaned by the manufacturer.

The Trak drive was another fast drive, requiring only 5 ms to seek a track. The card for the drive cable was even more inaccessible than the Teac drive, and to use it with the four-drive cable, it requires you to run the drive with the cover removed, use the last connector on the cable, or use a disk-drive extender cable, which costs about $10. It came with a single-drive cable already connected to it, so single-drive users won't need to get a cable when purchasing this drive. A nice feature of this drive is the disks are spring loaded. When you wish to remove a disk, you open the drive door and the disk springs out a half inch or so for easy removal. This is also the only drive with a power-on light as well as a drive-busy light.

The disk drives all worked reliably. I prefer the Percom drives because of the easy cable hook-up and relatively fast operation, but primarily because it was flippy. Nonetheless, all the drives worked well and deserve a recommendation. ■

## The Easy MANHATTAN Way To Handle
## EXPENSES, LISTS & MAILINGS

**EXPENSE LEDGER** — Files entries and generates reports on business or personal expenses. Sorts by date, finds and displays on-screen or prints out all expenses, month summary, or selection by payee, purpose or category. 15 categories assigned by user, displayed on screen when entering, editing or deleting. 48K version holds 470 entries, 32K holds 200 — use for year, quarter or month. $35.

**MAIL LIST MANAGER** — Holds 1,000 names and addresses in 48K two-disk version or 400 in one-disk 32K version. Sorts by zip code or name [lightning fast in machine code], prints labels one-up or two-up, selectable by any combination of user-entered Print Key or geographic area. Prints full list in notebook format. Two-disk version $50, one-disk $45.

**DISK LISTMAKER** — User-friendly simple database, holds 1,000 names or items plus 5-digit codes in 48K, 500 in 32K. Displays on-screen and prints out lists and sub-lists. Full editing, sorts by name. Disk version of original cassette program in use by schools, real estate firms, many individuals. $20.

For Models I or III [with DOS conversion]
Write for full catalog of disk & cassette programs
California residents add 6% tax

**MANHATTAN SOFTWARE**
Box 1063, Woodland Hills, CA 91365 ✓259
24-hour Visa/MC Order Line [213] 704-8495

## *MICRO MOONLIGHTER NEWSLETTER*

The ONLY publication devoted exclusively to helping you create, build, and maintain a home-based business using your micro-system. SUBSCRIBE NOW to what may be the most important publication in YOUR FUTURE!

1 Year (12 issues) only $25 U.S., $29 Canada, $35 World Wide

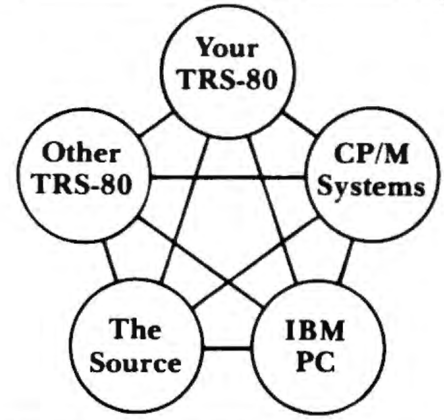
J. Norman Goode, Publisher
***Micro Moonlighter Newsletter***
2115-D Bernard Avenue • Nashville, TN 37212
Visa and Mastercard welcome. Send account number and expiration date. 615-297-5106.

☐ Visa Acct # ____________ Exp ________
☐ M/C Acct # ____________ Exp. ________
Name ______________________________
Address ____________________________
City ______________ St. _____ Zip ________

✓270

# Communicate!

## FILE TRANSFERS & TERMINAL EMULATION




**LYNC is *THE* choice of professionals for communications between computers. They choose LYNC because of its many powerful capabilities, its ease of use, and it works on many different computers. They use LYNC to transfer program or data files between computers, and to take advantage of the many features offered by the Source and other large-data base systems. LYNC's unique closed-loop error detection and correction protocol gives professionals the assurance that their data is getting across without being lost or garbled. Computer communications is the wave of the future, and if you are ready for the future, LYNC is here NOW.**

**E/LYNC: $45**
**Send and receive ANY type of files between your computer and another computer running LYNC or E/LYNC. File transfers are done using the exclusive LYNC closed-loop protocol. E/LYNC also has a Terminal Emulation mode for connecting with large time-sharing database systems or remote bulletin board services. E/LYNC can be upgraded to LYNC for $30.**

**LYNC: $75**
**This version of LYNC is exclusively for owners of the TRS-80 Model I & III. In addition to the features of E/LYNC, LYNC allows the user to fetch files from the remote computer and use multiple file names when sending or fetching. In the Terminal Mode you can send data files to the remote computer and store received data to disk files.**

**LYNC is also available for CP/M systems and IBM PC systems for $125.**

**LICENSING: E/LYNC and LYNC are licensed on a per user basis. If you intend to use E/LYNC or LYNC to communicate with another person, they must also purchase E/LYNC or LYNC. Versions for CP/M are written in Z-80 assembly language. Versions for IBM PC are written in IBM 8088 assembly language, and require PC DOS.**

**AVAILABLE DISK FORMATS: TRS-80 Mod I & III, TRS-80 Mod II with CP/M, 8" standard, IBM PC, Vector Graphic, Cromenco, North Star, Osborne, Xerox 820 (8" & 5"), NLS KayPro, SuperBrain, TeleVideo.**

**TERMS: Money Order, Check, C.O.D., MC or Visa. First Class Shipping Included. UPS Blue add $3. C.O.D. add $5. CA res. add 6%. Overseas Airmail add $6. Manual Only, $10 (Specify Computer).**

**PHONE ORDER DESK**
**TECHNICAL HOT LINE**
**(805) 966-3077**
**10 am-4 pm PST Monday - Friday**

**(805) 966-3077**

**INTERNATIONAL SOFTWARE ALLIANCE**
**1835 Mission Ridge Road Santa Barbara, CA 93103**
✓516

LYNC & E/LYNC are trademarks of Midnight Software. CP/M is a trademark of Digital Research

See List of Advertisers on Page 451

# Excitement and Fantasy

THE PROGRAM STORE

## HELLFIRE WARRIOR

From Epyx

This sequel to TEMPLE OF APSHAI gives you 4 new levels of dunjons with ***60 rooms each!*** You'll find giant insects, skeletons, ghouls, mummies, dragons and bottomless pits. Can you conquer them and rescue the Warrior Maid Brunhild to break the spell that binds her?

**16K Tape or 32K Disk, $39.95**

Expansion Modules for HELLFIRE WARRIOR (You must have HELLFIRE WARRIOR to play):

**THE KEYS OF ACHERON:** Find the 4 jewels with the help of a friendly unicorn.

**DANGER IN DRINDISTI:** Defeat the king's enemies in over 100 caverns and chambers.

**Each Module:**
**16K Tape or**
**32K Disk, $19.95**

## Warriors of RAS

By Masteller from Med Systems

In this adventure/role playing series, you ***create*** the characters, and can save them for future games, or use them in other volumes of the series.

**DUNZHIN, Vol. 1**

You're in a many-level dungeon filled with demons and goblins.

**KAIV, Vol. 2**

Can you find the treasure buried deep within the catacombs of the KAIV?

**THE WYLDE, Vol. 3**

The newest in this fascinating series.

**Each Volume:**
**48K Tape or Disk; $29.95**

## ASYLUM

From Med Systems **

You are sitting alone at 2 AM. Your eyes are bloodshot as you peer into your computer's screen and cry, "I must be CRAZY!" If this has never happened to you, you've never tried ASYLUM. It's Med Systems most ambitious 3-D graphics adventure yet!

ASYLUM places you on a cot in a small (padded?) room. Periodically the janitor lobs a hand grenade through the window. What you do next could mean escape—or disaster.

**16K Tape, $19.95 32K Disk, $22.95**

**ASYLUM II Now Available:**
**16K Tape, $19.95 32K Disk, $22.95**

Also order **DEATHMAZE 5000,**
**Med System's challenging 3-D maze game.**
**16K Tape, $14.95 32K Disk, $19.95**

## TANK ARCADE

From Avalon Hill

Fast-paced 1 or 2 player game with classic arcade action and battlefield strategy. You must predetermine how many hits it will take to wipe out your opponents, and then ***do it!*** The battlefield changes randomly after each game.

**16K Tape, $14.95**

## SHOOTOUT AT THE OK GALAXY

From Avalon Hill

This exciting new game requires fast arcade response and well-thought-out strategy. ***Thirty alien warships*** have entered your Patrol Zone—can you handle your defense? Are your shields up? Have you checked your energy level? Is your azimuth set? OK then... ***Good Luck!***

**16K Tape, $19.95**

## VOYAGER I

From Avalon Hill

You're on board a spaceship infested with killer robots in this graphic science fiction game. You must clear the 4-level 144-location ship of robots and arm it to self-destruct. Can you do it and escape before you, too, are blown up? High-speed graphics are represented in 3-D perspective representing your eye's view; with instant switching to floor plan maps.

**16K Tape, $19.95**

## DNIEPER RIVER LINE

From Avalon Hill

It's 1943; you're the German Commander defending the DNEIPER RIVER LINE against the Soviet units, controlled by the computer. Can you hold your position? 4 levels of difficulty; with over 300 illustrated counters and mounted mapboard.

**48K Disk, $29.95**

**Also available:**

**TANKTICS: Computer/board game of Germans against Russians in WWII.**

**16K Tape including wargame board, $23.95**

## FROGGER

From Cornsoft

One of the most popular arcade games ever... now available for your home computer! Just as in the arcade game, you must dodge heavy traffic and jump the moving logs in the stream to get home. Fantastic graphics! Requires joystick.

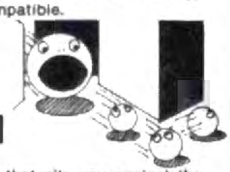
**16K Tape or**
**32K Disk, $34.95**

## BOUNCEOIDS

From Cornsoft

We're not sure what they are, but they're **huge** and dangerous! BOUNCEOIDS bounce off walls, and can kill with one touch. Your mission is to destroy them with your new class 4 annihilator. Then there are the terrible bugs coming out of null space to crunch you to smithereens. You'll need coordination, strategy and courage! Joystick-compatible.

**16K Tape, $15.95**
**16K Disk, $19.95**



## SCARFMAN

From Cornsoft

Action-filled arcade game that pits you against the monsters. Race your Scarfman around a maze, gobbling up scoring dots. You are pursued by five monsters: If you eat a "+" they'll lower their eyes and you can eat them, otherwise they'll eat you!

With exciting graphics and sound, SCARFMAN may be played using the keyboard or Joystick.

WARNING: MAY BE HABIT-FORMING!

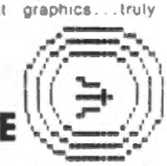
**16K Tape, $15.95**
**16K Disk, $19.95**
***While they last!***

## CRAZY PAINTER

From Cornsoft

Your assignment is to ***paint the maze,*** avoiding everyone who tries to stop you. If anyone walks in the paint, it's ruined, and you must go back and repaint it. With sound, great graphics... truly "a-maze-ing".

**16K Tape, $15.95**
**16K Disk, $19.95**



## SPACE CASTLE

From Cornsoft

Ahead of you lies the menacing castle, floating in space amidst its layers of orbiting shields. At intervals, smart mines spin off the shields and head for your ship. Dodging the mines and destroying the shields isn't your only problem, though: once you penetrate the innermost shield, The evil Yugdab will unleash all his fury in an attack! A fast-paced and challenging arcade game, indeed.

**16K Tape, $15.95 16K Disk, $19.95**

**THE PROGRAM STORE • Dept. 8E212 • Box 9609 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016**

Item Tape/Disk/Book Price Postage $2.00 Name______

Total ______ Address______

City______ State______ Zip______

☐ CHECK ☐ VISA ☐ MASTERCARD Card #______ Exp______

See List of Advertisers on Page 451

LOAD 80

*Make your MX-80 look like a magician.*

# GRAPHTRAX Tricks

*Mark Schneider*
*3109 Karen Avenue*
*Long Beach, CA 90808*

The GRAFTRAX option for my Epson printer has bit-plot graphics; a new italic print font; a backspace feature to help overcome the TRS-80 quirk of including a line feed with every carriage return; and a TRS-80 mode which accepts graphics in the 128–191 ASCII range of the computer without losing any printer features. These features alone were worth the price, but what kind of characters could I create?

As a math teacher, I used my printer to create worksheets and exams. I had to add many symbols by hand after printing, such as greater than or equal to, less than or equal to, not equal to, square root, divided by, degree, integral, delta and many more. I wanted more than math symbols so I created a whole new alphabet. I designed an alphabet for Hebrew, which does not duplicate any English letters.

### Letter Size

The standard character font on the printer is five across by seven down, with an extra two dots for lowercase descenders. However, the printer plays tricks with some letters, printing between dots. I wanted my letters to be 10 dots wide. GRAFTRAX can print the dots twice as close; my characters print normal width with twice as much detail.

The maximum height allowed is eight, so I used all eight dots: six for normal letters and two for letters with descenders. Each dot is assigned a power of two for a value. The bottom dot is two raised to the zero power, or one. The second is worth two, and so on to the top dot (two to the eighth, or 128). By adding the value of those dots required in each column, you arrive at a number between 0 and 255. LPRINTing that code fires the correct pins and prints the column. The calling sequence LPRINT CHR$(27)"L" ; CHR$(n1) ; CHR$(n2) tells the printer that the next n1 + (n2 × 256) characters it receives should be printed as bit image graphics. You may print up to 960 dots on a line.

To print one solid column of dots use LPRINT CHR$(27)"L" ; CHR$(1) ; CHR$(0) ; CHR$(255). To underline an entire line (960

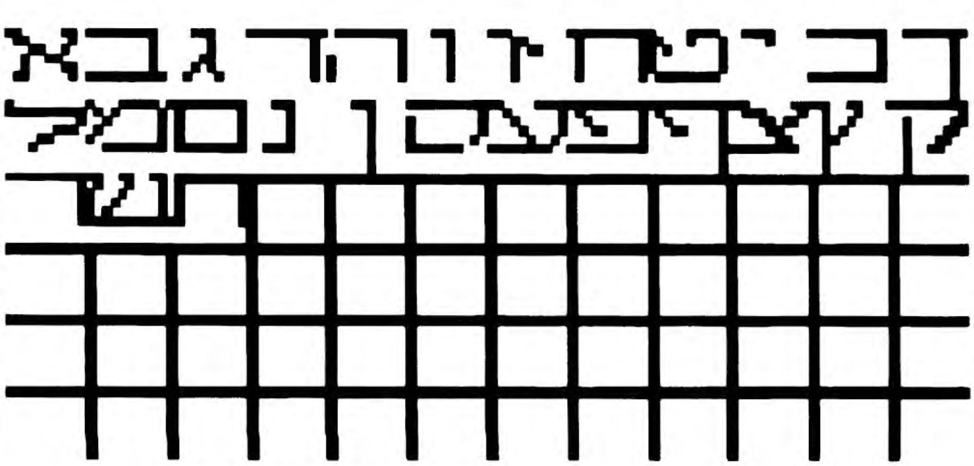


*Fig. 1. Screen Display*

***The Key Box***

**Basic Level II**
**Model I**
**16K RAM**
**Epson Printer**

# THE ULTIMATE IN COLORCOMPUTING

*For the TRS-80 Color Computer and TDP System 100 Personal Computer*

## Super "Color" Writer II

**By Tim Nelson**

### The Rolls Royce of Word Processors

The **Super "Color" Writer** is a FAST, machine code, full featured, character (screen) oriented word processing system for the TRS-80™ Color Computer and ANY printer. The video display is styled after a professional phosphor (green characters on black background) display for hours of use without eye fatigue (optional orange on black). The unique print WINDOW frees you from 32, 51 or 64 character lines FOREVER! This window can be moved anywhere in the text file, up, down, left or right to display the text as it will be printed without wasting paper. You can create or edit **Super "Color" Terminal** files, ASCII files, BASIC programs or Editor/Assembler source listings. It's simple enough for beginners with 4K and . . . for the professional writer with a 32K disk system and a lot to say, there's plenty of room to say it!

| COMPARISON CHART | SUPER COLOR WRITER | | | THE COMPETITION | | |
|---|---|---|---|---|---|---|
| System Size | 4K | 16K | 32K | 4K | 16K | 32K |
| TAPE: Text space | N/A | 7K | 23K | N/A | 2K | 18K |
| ROMPAK: Text space | 2.5K | 16K | 31K | N/A | N/A | N/A |
| DISK: Text space | N/A | 5.5K | 21.5K | N/A | 0.5K | 16.5K |
| Right Justify | | YES | | | NO | |
| Video Window | | YES | | | NO | |
| Edit any ASCII File | | YES | | | NO | |
| Programmable Function | | YES | | | NO | |

The figures speak for themselves and with professional features like PROGRAMMABLE function string commands to perform up to 28 commands automatically. PROGRAMMABLE text file chaining, PROGRAMMABLE column insert & delete, and right hand JUSTIFICATION with punctuation precedence, the choice is clear but there's still more! In their September '82 issue, "80 MICRO" says, "The Color Computer has finally come of age. Nothing illustrates that coming of age better than this offering (SUPER "COLOR" WRITER) by Nelson Software". The **Super "Color" Writer** takes full advantage of the new breed of "smart printers" with Control codes 1-31, 20 Programmable control codes 0-255 for special needs. Works perfectly with all Epson, Radio Shack, Okidata, NEC, IDS, Centronics, Citoh, Smith Corona, Diablo Etc., Matrix, or Letter Quality Printers.

### CHECK THESE FEATURES!!

User friendly • Easy commands • 32K Compatible • Window • Key beep • HELP table • 128 character ASCII & graphics • Mem left and Mem used • Full cursor control • Quick paging • Scrolling • Word wrap around • Tabs • Repeat all functions • Repeat last command • Insert character & line • Delete character, delete to end of line, line to cursor, line & block • Block move, copy & delete •Global Search, Exchange & Delete • Merge or Append files • Imbed Control Codes in text • Underline • Superscripts • Subscripts • Headers, Footers & 2 Auxiliary footnotes on odd, even or all pages definable position • Flush right • Non-breakable space • 4 centering modes: 5, 8.3, 10 & 16.7 (CPI) • Full page & print formatting in text • Single sheet pause • Set Page length • Line length, Line spacing, Margins, Page numbers • Title pages • Printer baud: 110, 300, 600, 1200, 2400 • Linefeeds after CR • Soft & hard formfeed • Works with 8 bit printer fix • and more!

## Super "Color" Writer II Disk

The Disk version of the **Super "Color" Writer** works with the TRS-80C Disk System and has all the features listed above plus many more! Use with up to four Disk Drives. Includes an extended HELP table you can access at any time. Call a directory, print FREE space, Kill disk files and SAVE and LOAD text files you've created all from the **Super "Color" Writer.** Print, merge or append any **Super "Color" Terminal** file, ASCII file, BASIC program or Editor/Assembler source listing stored on the Disk or tape. The **Super "Color" Writer Disk** version has additional formatting and print features for more control over your printer and PROGRAMMABLE chaining of disk files for "hands off" operation. Print an entire BOOK without ever touching a thing! Includes comprehensive 90 plus page Tutorial manual.

**TAPE $49.95 ROMPAK $74.95 DISK $99.95**

**Tutorial only $15.00 (Refundable with purchase)**

**ORDERING INCLUDE $3.00 for shipping in the U.S. & Canada, $6.00 for Foreign orders. C.O.D. add $2.00.**

**NELSON SOFTWARE SYSTEMS**

VISA MasterCard

✓ 128

9072 Lyndale Avenue So. 612/881-2777

A Division of Softlaw Corporation Minneapolis, Minnesota 55420 U. S. A.

## Super "Color" Terminal

**By Dan Nelson**

### The Ultimate in Smart Terminals

The **Super "Color" Terminal** turns the Color Computer into a Super-smart terminal with all the features of VIDEOTEX™ plus much more. COMMUNICATE with Dow Jones & Compuserve and with computers like the TRS-80™ MODEL I, II, III, APPLE etc., via moden or RS-232 direct! Save the data to tape or print it! Reduces ON-LINE cost to a minimum!

### FEATURES

10 buffer size settings from 2-30K • Buffer full indicator • Prints buffer contents • Full 128 ASCII keyboard • Compatible with **Super "Color" Writer** files • UPLOAD & DOWNLOAD ASCII files, Machine Language & Basic programs • Set RS-232 parameters • Duplex: Half/Full • Baud Rate: 110, 300, 600, 1200, 2400, 4800 • Word Lengths 5, 6, 7 or 8 • Parity: Odd, Even or None • Stop Bits: 1-9 • Local linefeeds to screen • Tape save & load for ASCII files, Machine code & Basic programs • Unique clone feature for copying any tape.

## Super "Color" Terminal Disk

The disk version of the **Super "Color" Terminal** works with the TRS-80C Disk system and has all the features listed above plus many more! Use with up to four Disk Drives • Call a directory, print FREE space, kill disk files, save and load text files or BASIC programs • Echo ability in full duplex • Lower case masking • 10 Keystroke Multiplier (MACRO) buffers that can be saved on disk to perform repetitive log-on tasks and send short messages (up to 250 characters each) • Programmable prompt or delay for send next line • Selectable character trapping • Set printer Baud rate to 110, 300, 600, 1200, & 2400 • Operators Manual.

**TAPE $39.95 ROMPAK $49.95 DISK $69.95**

**Operators manual only $10.00 (Refundable with purchase)**

NEW!!

## Super "Color" Mailer

Correspondence-Mailmerge

The **Super "Color" Mailer** is a powerful multi-purpose file merging program that uses files created by the **Super "Color" Writer II.** One of **Super "Color" Mailer's** most popular uses is producing customized form letters — at a fraction of the time and expense of individually typed letters. With **Super "Color" Mailer** you can combine a **Super "Color" Writer II** file containg a form letter with a file containing a list of names and addresses. You can even insert special words and phrases — unique to each addressee — into the body of the letter. Other **Super "Color" Mailer** uses include creating invoices, printing mailing labels, addressing envelopes, and producing "boiler plate" legal documents out of many different paragraphs. Features include: the ability to selectively print mailing lists by any of up to 10 user definable fields • automatically prints current date • address • salutation • closing • P.S. etc. • prints any ASCII file • justification.

**TAPE $39.95 DISK $59.95**

NEW!!

## Super "Color" Disk-ZAP

The Ultimate in Disk Repair Utilities

A must for ALL Color Computer Disk system owners. A high-speed machine code Disk Utility that can copy sectors and tracks • repair directory tracks and smashed disks, etc. **Super "Color" Disk-ZAP** has a special screen display that displays sector, track and memory contents in HEXADECIMAL and ASCII at the same time with double cursors that can be moved in any direction. With **Super "Color" Disk-ZAP** you are able to verify or modify disk sectors at will. You can even type right onto the Disk! You can send sector contents to the printer or any other RS-232 device in either ASCII or HEXADECIMAL listing. Search the entire Diskette for any ASCII or HEXADECIMAL string. Comes complete with comprehensive manual.

**DISK ONLY $69.95**

COMING SOON!!

## Super "Color" Calc

Electronic Spread Sheet

The finest electronic spread sheet and financial modeling program available for the Color Computer — A sophisticated yet easy to use, calculating and planning tool. Project figures into the future to answer the "What if?" questions you face. Create files compatible with the **Super "Color" Writer II.** Combine spread sheet tables with your documents to create ledgers, projections, statistical & financial reports & budgets.

**AVAILABLE AT DEALERS EVERYWHERE. IF NOT, ASK WHY!!**

**TRS-80 is a registered trademark of the Tandy Corp.**

dots) use LPRINT CHR$(27)"L" ; CHR$(192) ; CHR$(3) ; : FOR X = 1 TO 960 : LPRINT CHR$(1); : NEXT X . The Basic interpreter does not correctly LPRINT CHR$s 0, 10, 11 or 12. The manual suggests POKEing these values directly to the printer port (at address 14312), bypassing the Basic interpreter. The first example in this paragraph does not work as written; change it to

*Program Listing 1. Create File*

```
1ØØ DIML(12Ø):CLS:PRINT"
                    MX8Ø Graftrax Worksheet
               from DATAWARE by Mark Schneider

This program will aid you in creating bit-graphic characters
for use with an Epson printer equiped with the  'GRAFTRAX'
option."
11Ø PRINT:PRINT"Would you like instructions printed out on your
printer?
(None will appear on the screen)!

Y or N  ";
12Ø I$=INKEY$:IFI$="N"ORI$="n"THEN22Ø
     ELSEIFI$="y"ORI$="Y"THEN13Ø
     ELSEPRINT@711,"?";:FORCC=1TO5Ø:NEXTCC:PRINT@711," ";:
     FORCC=1TO5Ø:NEXTCC:GOTO12Ø
13Ø LPRINT"     GRAFTRAX Worksheet    from DATAWARE    by Mark S
chneider"
14Ø LPRINT:LPRINT"                      CONTROL KEYS FOR DRAWING GRAPH
ICS

4 arrow keys...................move flashing cursor around box"
15Ø LPRINT"
<enter> and arrow key..........draw a dot and move cursor"
16Ø LPRINT"
<clear> and right arrow........move cursor to upper left of next
 box

<clear> and left arrow.........move cursor to upper left of prev
ious box"
17Ø LPRINT"
<clear> and <enter>............erase borders in this box after a
rrival"
18Ø LPRINT"
<space> and right arrow........leave a dot at the cursor, then m
ove to next box"
19Ø LPRINT"
<space> and <enter>............build file containing bit pattern
s

<space> and <clear>............build alternate file"
2ØØ LPRINT"
<clear> and up arrow...........begin drawing again

<space> and up arrow...........exit program"
21Ø LPRINT:LPRINT:LPRINT"
After all characters have been created:

1...move the cursor to the next box.
2...Press <space> and <enter> to build the file named in line 3Ø
Ø.
3...Press <space> and <clear> to build a file with /ALT extensio
n."
22Ø CLS:PRINT"Do you wish to <B>uild or <E>dit a drawing?

B or E ";
23Ø I$=INKEY$:IFI$="B"ORI$="b"THENMODE$="build"
     ELSEIFI$="E"ORI$="e"THENMODE$="edit"
     ELSEPRINT@135,"?";:FORCC=1TO5Ø:NEXTCC:PRINT@135," ";:
     FORCC=1TO5Ø:NEXTCC:GOTO23Ø
24Ø PRINTCHR$(27):PRINT"Name for file you wish to ";MODE$;":
";:INPUTT$
25Ø PRINT"
Besides ";T$;" you may also build the file ";T$;"/ALT":
FORCC=1TO8ØØ:NEXTCC
26Ø CLS:FORX=1ØTO11ØSTEP1Ø:FORY=ØTO47:SET(X,Y):NEXTY,X
27Ø FORY=8TO4ØSTEP8:FORX=ØTO119:SET(X,Y):NEXTX,Y
28Ø '
     Initialize cursor   (XX,YY) is upper left point of box

29Ø XX=Ø:YY=Ø:X=Ø:Y=Ø:I=Ø:J=Ø
3ØØ IFMODE$="edit"THEN2Ø1Ø
1ØØØ '

      Begin Loop for Keyboard Entry

1Ø1Ø X=XX+I:Y=YY+J                          'increment from upper
left of box
1Ø2Ø SET(X,Y):FORCC=1TO25:NEXT:RESET(X,Y)  'flash cursor
1Ø3Ø P=PEEK(144ØØ)                          'read keyboard
1Ø4Ø IFP=ØTHEN1Ø2Ø                          'no key pressed
1Ø5Ø IFP=1ØTHEN24Ø                          'begin again
1Ø6Ø IFP=8THENIFJ>ØTHENJ=J-1                'move cursor up
1Ø7Ø IFP=16THENIFJ<7THENJ=J+1               'move cursor down
1Ø8Ø IFP=32THENIFI>ØTHENI=I-1               'move cursor left
1Ø9Ø IFP=64THENIFI<9THENI=I+1               'move cursor right
11ØØ IFP=9THENIFJ>ØTHENSET(X,Y):J=J-1       'draw and move up
111Ø IFP=17THENIFJ<7THENSET(X,Y):J=J+1      'draw and move down
112Ø IFP=33THENIFI>ØTHENSET(X,Y):I=I-1      'draw and move left
113Ø IFP=65THENIFI<9THENSET(X,Y):I=I+1      'draw and move right
114Ø IFP=34THEN122Ø                         'move back one box
115Ø IFP=192THENZ=1:GOTO125Ø                'draw and move to next
 box
116Ø IFP=66THENZ=Ø:GOTO125Ø                 'move to next box
117Ø IFP=3THEN129Ø                          'erase borders
118Ø IFP=129THENTT$=T$:GOTO3Ø1Ø             'build file
119Ø IFP=13ØTHENTT$=T$+"/ALT":GOTO3Ø1Ø      'build alternate file
12ØØ IFP=136THENCLS:END                     'exit program
121Ø GOTO1ØØØ                               'return to begin loop
122Ø IFXX>ØTHENXX=XX-1ØELSEXX=11Ø:YY=YY-8  'back one box
```

*Program continues*

*Program continued*

```
123Ø IFYY<ØTHENYY=YY+8                      'check if off screen
124Ø I=Ø:J=Ø:GOTO1ØØØ                       'no increment in box
125Ø IFXX<11ØTHENXX=XX+1ØELSEXX=Ø:YY=YY+8  'next box
126Ø IFYY>47THENYY=YY-8                     'check if off screen
127Ø I=Ø:J=Ø:IFZ=1THENSET(X,Y)              'draw point if Z=1
128Ø GOTO1ØØØ
129Ø FORII=ØTO9:RESET(XX+II,YY):NEXT        'erase border
13ØØ FORJJ=ØTO7:RESET(XX,YY+JJ):NEXT:GOTO1ØØØ
2ØØØ '
     Edit existing file......first build array
                             then print characters

2Ø1Ø OPEN"I",1,T$
2Ø2Ø X=Ø:Y=7                      'first character bottom left
2Ø3Ø INPUT#1,M                    'number of characters
2Ø4Ø DIMX(M,1Ø)                   'store binary codes
2Ø5Ø FORN=1TOM:FORA=1TO1Ø
2Ø6Ø INPUT#1,X(N,A)               'get binary codes
2Ø7Ø NEXTA,N
2Ø8Ø FORN=1TOM:FORNN=1TO1Ø
2Ø9Ø A=X(N,NN)
21ØØ FORV=7TOØSTEP-1              'powers of 2
211Ø U=INT(A/2[V)                 'binary factor?
212Ø A=A-U*2[V                    'remainder
213Ø IFU=1THENSET(X,Y-V)
     ELSERESET(X,Y-V)             'set if a factor
214Ø NEXTV
215Ø X=X+1:IFX=12ØTHENX=Ø:Y=Y+8  'next box
216Ø NEXTNN,N
217Ø CLOSE
218Ø MODE$=""
219Ø GOTO29Ø
3ØØØ '

     Create file...first print number of characters
                   then build binary totals for each
                   character column and add to file

3Ø1Ø OPEN"O",1,TT$:D=Ø
3Ø2Ø PRINT#1,(YY/8)*12+XX/1Ø     '# of characters
3Ø3Ø FORA=7TOYY+7STEP8            'increment rows
3Ø4Ø FORB=ØTO119                  'move along botom of boxes
3Ø5Ø FORC=ØTO7                    'move up each column of charact
ers
3Ø6Ø IFPOINT(B,A-C)THEND=D+2[C   'if point set, increment by tha
t power of 2
3Ø7Ø NEXTC
3Ø8Ø L(B)=D                       'build array for each row
3Ø9Ø D=Ø                          'reset binary counter
31ØØ NEXTB
311Ø FORH=ØTO119
312Ø PRINT#1,L(H);                'add binary bit totals for this
 row to file
313Ø NEXTH,A
314Ø CLOSE
315Ø GOTO1Ø3Ø                     'back to loop
```

```
1ØØ '
     Sample program to read in file of bit column codes
     and store them in array T( character # , column #)

11Ø CLS:INPUT"Name of file containing bit graphics";N$
12Ø CLS
13Ø OPEN"I",1,N$
14Ø INPUT#1,M              'M is # of characters
15Ø DIMT(M,1Ø)             'M characters , 1Ø columns per characte
r
16Ø FORA=1TOM              'increment characters
17Ø FORAA=1TO1Ø            'increment columns
18Ø INPUT#1,T(A,AA)        'read binary bit total for each column
19Ø NEXTAA
2ØØ NEXTA
21Ø CLOSE
3ØØ '
       Print characters by calling a subroutine
     W = 2 prints each column twice for double width

31Ø FORW=1TO2                   'once normal, then double width
32Ø FORA=27TO1STEP-1:GOSUB41Ø:NEXTA:
     LPRINT" is the Hebrew alphabet."
33Ø LPRINT"
     Thank you very much.......";:
     A=5:GOSUB4ØØ:A=2:GOSUB4ØØ:A=25:GOSUB4ØØ:
     LPRINTSTRING$(W,32);:
     A=5:GOSUB4ØØ:A=4:GOSUB4ØØ:A=6:GOSUB4ØØ:A=27:GOSUB4ØØ
34Ø LPRINT"

     Peace be upon you..........";:
     A=15:GOSUB4ØØ:A=11:GOSUB4ØØ:A=1Ø:GOSUB4ØØ:
     A=13:GOSUB4ØØ:A=19:GOSUB4ØØ:LPRINTSTRING$(W,32);:
     A=15:GOSUB4ØØ:A=6:GOSUB4ØØ:A=13:GOSUB4ØØ:
     A=26:GOSUB4ØØ:LPRINT:LPRINT:LPRINT
35Ø NEXTW
36Ø END
4ØØ '
       Subroutine to print a bit-plot graphic character

41Ø LPRINTCHR$(27)"L";     'set printer to bit graphics mode
42Ø POKE14312,W*1Ø         'W columns per character
43Ø LPRINTCHR$(255);       'second argument Ø since fewer than 25
6 columns
44Ø FORAA=1TO1Ø            '1Ø columns per character
45Ø POKE14312,T(A,AA)      'poke binary total to printer port
46Ø IFW=2
     THENPOKE14312,T(A,AA) 'if double width then repeat
47Ø NEXTAA
48Ø RETURN
```

*Program Listing 2. Read File*

LPRINT CHR$(27)"L" ; CHR$(1) ; : POKE 14312,0 : LPRINT CHR$(255). My program POKEs all values to 14312 to avoid these problems.

**Bit-Graphic File**

The first program sketches a giant model of each character desired. Using the arrow keys, space, Enter, and Clear in various combinations, you can move the cursor within each character, draw lines and dots, and move to a new box. Memory location 14400 scans all these keys; by PEEKing this location, the keys, individually or in combination, are sensed.

To fit as many characters on the screen as possible, the border lines are part of the character box. As the cursor moves to each box, erase the border lines by entering and clearing to avoid interference. Also to save space, the instructions are produced on the printer. When forming characters, decide whether to draw a dot where the cursor sits before moving it to the next box. Pressing Clear and right arrow does not draw a dot before moving, while pressing space and right arrow leaves a dot (see lines 1150, 1160, 1270).

When the characters are complete (see Fig. 2), a scanning routine determines the bit-plot codes and prints them out to a disk file. Sometimes you cannot decide on the best form for a letter by looking at a giant model; an option builds two files. The file name asked for at the beginning of the program is used if you press space and Enter. Then modify the drawing and save an alternate version by pressing space and Clear. This file has the same name and /ALT as an extension (lines 1180, 1190). You can edit from one file and save the revised version to the alternate file.

The file building portion of the program (lines 3000–3140) writes the number of characters to the file. Then the program scans the screen one line at a time, and computes the binary bit code for each column (lines 3030–3100). Because the Point command works rather slowly, the disk spins for several minutes if each code is written as it is computed. By storing the codes in an array (line 3080), an entire line of 12 characters is written to disk at once (lines 3110–3130).

**Bit-Plot Graphic Characters**

The second program reads the file created in the first program to build an array containing the 10 column bit codes for each character (lines 160–200). A subroutine at lines 410–470 prints each character. The calling sequence is as I stated at the beginning of this article: control L, and two arguments sent to the printer. The first argument is POKEd, as it may be 10. The second should be a zero, but since you cannot LPRINT CHR$(0), the printer allows substituting CHR$(255) for CHR$(0). You could also POKE this value to the printer port. The variable W (lines 310, 420, 460) allows double width letters by printing each column twice. You could form triple-size letters similarly, although they would look grotesque.

The sample printing routines (lines 310–350) demonstrate ways to call the subroutine. First, every character is printed (here in reverse order, as Hebrew is read from right to left); then selected letters are chosen using the number order in which they were read from the disk (see Fig. 2).■

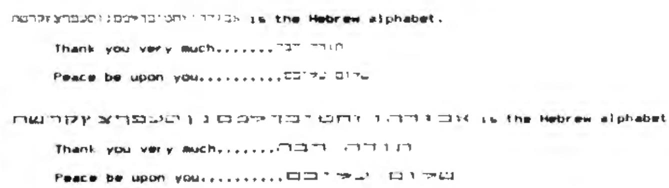



*Fig. 2. Sample Printout*

*Eliminate unnecessary Model II forms messages.*

# Forms Cleanup

*Jim Barbarello*
*RD #1 Box 241 H*
*Tennent Road*
*Englishtown, NJ 07726*

The current Model II Disk Operating System (TRSDOS 2.0a) contains a commonly used library command called Forms. Forms initializes the printer-driver software, and skips a preset number of lines after a specified number of lines have been printed.

> ***The Key Box***
> **Model II Basic**
> **64K RAM**
> **Model II**
> **TRSDOS 2.0a**
> **One Disk Drive**
> **Printer required**

A printer-driver initialization such as Forms should always be executed immediately before printing a report to ensure that the printer starts at the top of Forms. Unfortunately, Forms automatically prints a series of messages on the screen which the operator must answer in a specific manner.

Aside from messing up an otherwise meticulous screen presentation, the Forms messages can confuse the operator. Fortunately, there is a much simpler way of performing the required initialization function while also eliminating Forms' drawbacks.

```
TRS-80 Model II DEBUG Program
2000     C3 7F 2E 5D 44 7A 44 65   43 1D 45 50 43 DB 5E 50   ...+DzDeC.EPC.+X
2010     61 F8 3A D9 4E C8 4F B3   41 2B 4F 3C 50 42 50 CF   a.:.N.O.A+O<PBP.
2020     50 E4 50 C0 61 C3 61 C6   61 33 65 06 65 99 65 A7   P.P.a.a.a3e.e.e.
2030     61 AA 61 AD 61 5D 44 91   44 9D 44 0A 45 E3 5C D0   a.a.a+D.D.D.E.+.
2040     5A E0 5A E5 5A A1 5D EF   5C FF 5C 2E 5D 47 5D 77   Z.Z.Z.+.+.+.-G+w
2050     5D 00 5D C5 4E 44 C6 4F   52 C3 4C 53 D2 41 4E 44   +.+.ND.OR.LS.AND
2060     4F 4D CE 45 58 54 C4 41   54 41 C9 4E 50 55 54 CA   OM.EXT.ATA.NPUT.
2070     49 4D D2 45 41 44 CC 45   54 C7 4F 54 4F D2 55 4E   IM.EAD.ET.OTO.UN
   PC     SP   SZHPNC  AF   BC   DE   HL   IX   IY   AF'  BC'  DE'  HL'
 2000   21FE   000000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
? P
DEBUG is now ON
TRSDOS READY
DEBUG
```

*Figure 1*

```
TRS-80 Model II DEBUG Program
F000     56 23 4E 06 42 3E 11 CF   C9 00 FF 00 FF 00 FF 00   V#N.B>..........
```

*Figure 2*

### FRMS

FRMS is a short machine-language program that allows you to set the number of printed lines on a page and maximum characters per line (the maximum page length is preset at 66). FRMS is simple to use and completely eliminates the annoying and sometimes confusing Forms messages. No operator intervention is required to use FRMS. Since FRMS acts instantaneously, the operator has no indication that a Forms Set is required or being performed.

FRMS initializes the printer-driver parameters indicated above but does not check the printer's status. The line printer need not be on-line when FRMS is executed. Before you can use FRMS, you must perform three one-time tasks:

- Create the machine language program (FRMS).
- Save the created program on disk.
- Incorporate the necessary Basic statements in the target Basic program.

### Creating the Program

We will use Debug (see pages 86–92 of the *TRS-80 Model II Disk Operating System Reference Manual*) to place the program into memory. Begin by starting up the Model II with a System disk. After entering the date (required) and time (optional) you will be in the TRSDOS Ready mode (unless the disk contains an auto startup procedure. If you automatically enter Basic, simply type SYSTEM and press enter and you will return to the TRSDOS Ready mode).

Enter Debug On and then enter Debug. The screen will appear similar to Fig. 1. Press M (to examine and change memory). The question mark prompt on the screen will be replaced with "M A=....". Type F000. The first line on the screen will be similar to Fig. 2. Now press the F1 key. The cursor will move to the first displayed number on the F000 line. Type in the following 18 characters exactly (do *not* press enter or the space bar):

56234E06083E11CFC9

Review the screen. If any

character is incorrect, use the right or left arrow keys as necessary to position the cursor over the incorrect character. Then retype the correct character. When all characters are correct, press the F2 key to enter this program into memory. Now press O (not zero) to turn Debug off. The FRMS program is now stored in memory. (This is a demonstration version which we will change later.)

### Saving the Program

Next we will use the library utility Dump (see page 102 of the TRSDOS manual). Enter DUMP FRMS(START = F000,END = F008). After some time, you will see "TRSDOS Ready." Type DIR and press enter. Your directory listing will now include a new file named FRMS.

### Using FRMS in a Basic Program

Type BASIC and press enter. When you see Ready, type AUTO and press enter and then enter the program shown in Listing 1.

Enter SAVE"FRMSTST". After the test program has been saved and you have turned on your printer and put it on line, run the program. Enter 2 as the number of lines, and 3 as the number of characters per line. Enter 123456789 as the test line. Your printed output will show:

```
123
456





789
```

Now enter 2 as the number of lines and 9 as the number of characters per line. Again, enter 123456789 as the test line. Your printed output will now be:

```
123456789
```

No top of form was performed as before.

Now that we know the FRMS program is working properly, we must modify it. When we performed the first test above, six blank lines were printed between the 456 and 789 because we set our page length at eight lines for demonstration purposes. We will now use Debug to modify FRMS for a 66-line page length.

### Changing the Demo

Enter SYSTEM to return to the TRSDOS Ready mode. Type DEBUG ON and press enter and then type DEBUG and press enter. Type MF004 to go to the memory location of the byte that must be changed. Press F1 to move the cursor to the F004 presentation line. Now type 42 (since 66 decimal equals 42 hex) and then press the F2 key to store the modified information into memory. Press O (not zero) to turn Debug off. Finally, enter DUMP FRMS(START = F000,END = F008) to save the modified FRMS program.

### Using FRMS in Your Basic Programs

You may use the current version of FRMS in any Basic program that does not require the use of any TRSDOS high memory modules or other machine-language programs. If this is a requirement of your Basic program, I will discuss later how to relocate FRMS and reserve a memory area for it.

The FRMS program must be on the same disk as the Basic program which will use it. Use the TRSDOS library command Move (see pages 120-121 of the *DOS Reference Manual*) to move FRMS to the disk which will contain the Basic program.

Next, you must select three variables (one must be an integer variable) that will not be used in the remainder of your Basic program. Our demonstration program used FL, FC and A%, but you can use any variables. As an example, we will specify Z1%, ZL and ZC.

The first required statement for your Basic program is DEFUSR0 = &HF000 (the "zero" in DEFUSR0 is optional). This should appear early in your program. This statement tells the computer that you plan to call a machine-language program using the command X = USR0 (variable), and that this machine-language program will reside in memory starting at location F000 hex (the "zero" in USR0 is also optional). Next, load this program into memory. The Basic program must include SYSTEM "LOAD FRMS". This statement

```
10  DEFUSR = &HF000:ON ERROR GOTO 90
20  SYSTEM"LOAD FRMS"
30  INPUT"Enter Number of Lines per Page...";FL
40  INPUT"Enter Number of Characters/Line...";FC
50  A% = FL*256 + FC
60  X = USR(A%)
70  PRINT:LINE INPUT"Enter Test Line for Printing...";TL$
80  LPRINT TL$:PRINT:PRINT:GOTO 30
90  RESUME 30
100 (press the break key when 100 appears)
```

*Program Listing 1. Demonstration program*

```
                ;---------------------------------------------
                ;       UTILITY PROGRAM # 1
                ;       NAME: FRMS
                ;       PROGRAMMER:     J. HARBARE [illegible]
                ;       LATEST UPDATE: [illegible] 6 NOV 1981
                ;
                ;ABSTRACT: [illegible]
                ; [illegible] register D) and the
                ;[illegible] register C)
                ;[illegible] BASIC [illegible] USR0[illegible]. It then
                ;performs a FORMS SET without the annoying prompts
                ;that must be answered by the operator. Maximum
                ;[illegible] per page [illegible] is preset at 66.
                ;---------------------------------------------
F000                    ORG     0F000H  ;PROGRAM STARTS AT 61440 DECIMAL
F000 56                 LD      D,(HL)  ;# CHAR/LINE[illegible] LOW ORDER BYTE)
F001 23                 INC     HL      ;POINT TO NEXT MEMORY ADDRESS
F002 4E                 LD      C,(HL)  ;# LINES/PAGE[illegible] HIGH ORDER BYTE)
F003 0642               LD      B,66    ;66 LINES PER PAGE
F005 3E11               LD      A,17    ;SVC # 17 (PRINTER INITIALIZATION)
F007 CF                 RST     8       ;EXECUTE SVC # 17
F008 C9                 RET             ;RETURN TO BASIC PROGRAM
F000                    END     0F000H
00000 TOTAL ERRORS
```

*Program Listing 2. Assembly language version of FRMS*

## STOCK MARKET DATA over 20 YEARS on DISK or TAPE

NOW AVAILABLE-
OVER 20 YEARS OF WEEKLY OR DAILY CLOSING PRICE DATA ON DISK OR TAPE FOR THESE PROVEN MARKET INDICATORS

1962 - 1982

- DOW JONES INDUSTRIAL
- DOW JONES TRANSPORTATION
- DOW JONES UTILITY
- STANDARD AND POOR'S 500

ALSO INCLUDES-
PROGRAMS TO MAINTAIN THESE FILES, CREATE FILES FOR YOUR OWN STOCKS, AND SAMPLE PROGRAMS ILLUSTRATING RUNNING AVERAGE AND CYCLIC MARKET ANALYSIS TECHNIQUES.

WEEKLY DATA
- □ 101 MODEL I/III CASSETTE.$34.99
- □ 102 MODEL III DISKETTE...$34.99
- □ 103 HARDCOPY PRINTOUT....$22.99

DAILY DATA
- □ 104 MODEL I/III CASSETTE.$44.99
- □ 105 MODEL III DISKETTE...$44.99
- □ 106 HARDCOPY PRINTOUT....$32.99

FREE SHIPPING WITHIN CONTINENTAL USA
(ADD 10% FOR SHIPPING ELSEWHERE)
24-HOUR TOLL-FREE ORDER LINE
1-800-227-3800 EXT. 942

YOUNG IDEAS, INC.
✓172
555 N. NEW BALLAS • SUITE 310
ST. LOUIS, MO 63141

# This Smart Little Program Lets You Talk Back To Big Computers.


TRS-80™ users now have an easy way to talk with Source,™ Compuserve,™ other computers and a variety of bulletin board systems. COMMWHIZ* is so simple to use, you'll wonder why it hasn't been done before. Yet it offers professional terminal capabilities to Mod I and Mod III users who have 32K with one disk or more. And at $79.95 post-paid, talk is cheap. Specify Mod I or III. VISA, M/C accepted. (VA residents add 4% sales tax.) All popular DOSs/Cassette Vers. Avail./Includes free auto file xfer program. From Dr. Dobbs Journal review: "We reccommend COMMWHIZ . . . both powerful and easy to learn . . . a fine and truly 'smart' terminal program which we will continue to use . . . its creators are responsive, helpful and available." Source - TCV527. Compuserve - 70220,235

Fall Discount Combo: Hayes Smartmodem & COMMWHIZ — $299 plus shipping

## VolksMicro Computer Systems Inc.

✓549

202 Packets Court, Suite C, Williamsburg, Virginia 23185 804 220 0005

TRS-80 and Scripsit are TMs of the Tandy Corp. Electric Pencil is the TM of Michael Shrayer Software Inc.

# "My brand of personal service makes Corvus the best mass-storage system you can buy."

"A system is only as good as the company you buy it from or the quality of its service. When you buy a Corvus system from me, I guarantee that quality. I do it in the only way possible... with my brand of personal service. That's what makes Corvus the best brand on the market. Ask our customers. They know Corvus. And they know my work."

Corvus connects with:†
- Corvus Concept
- Omninet/Constellation
- TRS 80
- Apple
- IBM
- Xerox
- Zenith
- Commodore
- S-100
- LSI-11

ARM disc subsystem also available
Specializing in N.E. Corridor

## LAWRENCE S. EPSTEIN ASSOCIATES

1669 59 Street, Brooklyn, NY 11204

✓392

(212) 236-3173

TELEX 238-790 (NYK)

† Registered Trademark




need only be executed once. Locate it immediately after the DEFUSR command. An early program line such as 20 DEFUSR = &HF000: SYSTEM"LOAD FRMS" would be perfect. Next, create a subroutine to access the FRMS function. It can begin at any line. I suggest the sample routine in Program Listing 3.

Then at any point in the program, you can perform the FRMS function by simply identifying the number of lines per page (ZL), the maximum characters per line (ZC) and performing a GOSUB 60000. An example is shown in Program Listing 4.

To use FRMS you must perform these four steps:

- Include the FRMS program on your Basic program disk.
- Include DEFUSR = &HF000: SYSTEM"LOAD FRMS" as a program line.
- Include the machine language Forms Set subroutine as shown in Listing 3 in your Basic program.
- Specify ZL and ZC. Then GOSUB 60000 (Forms Set subroutine).

If you require a page length other than 66 lines, you can modify FRMS as we did above. For instance, you could change the 42 at memory locations F004–F005 to 3C (3C hex equals 60 decimal. This sets page length to 60 lines). After saving this modified version (call it FRMS60) using the Dump command, use "SYSTEM"LOAD FRMS60"" in your Basic program to load the correct version.

### Relocating FRMS

In the normal Basic mode, memory top is set at EFFF hex. This leaves the memory from F000–F800 free for TRSDOS high memory and other machine-language programs without having to set user memory (F801–FFFF is the screen memory area). However, you may reserve extra memory upon entry into Basic and avoid any chance of memory conflict. To show this, I will relocate FRMS so that it loads into a protected memory area. Follow these steps:

- Start up the computer with a System disk containing FRMS.
- When "TRSDOS Ready" appears, type LOAD FRMS and press enter.
- Enter DUMP FRMSP (START = F000, END = F008, RELO = EFF6).
- Type DIR and press enter. Note you have a new directory entry (FRMSP).
- Enter AUTO BASIC xxxxxxxx/yyy -F:n -M:61429 <E> where xxxxxxxx/yyy is the name of a Basic program you wish to load and run automatically upon startup (this is optional. If not needed, leave one space after Basic); n is the number of disk file buffers you want available (this is also optional).
- Replace DEFUSR = &HF000: SYSTEM "LOAD FRMS" in your Basic program with DEFUSR = &HEFF6: SYSTEM"LOAD FRMSP" (P means protected). ■

*Jim Barbarello is employed by the US Army Communications Electronics Command. His hobbies are music and woodworking.*

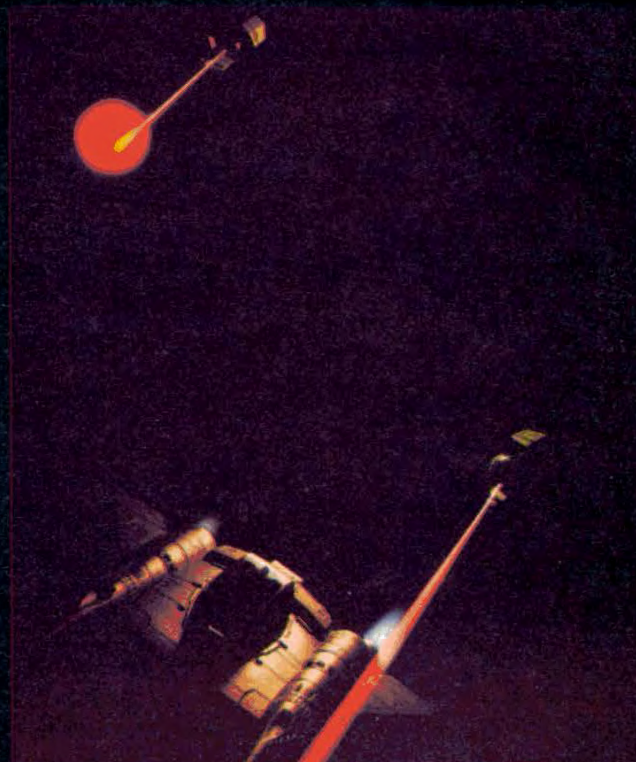
```
60000  REM** MACHINE LANGUAGE FORMS SET SUBROUTINE **
60010  Z1% = ZL*256 + ZC
60020  ZC = USR(Z1%):RETURN
```

*Program Listing 3. Forms Set subroutine*

```
4123  ZL=60:ZC=132:GOSUB 60000:REM** 60 L/P,132 C/L CHOSEN **
4124  GOSUB 5000:REM** PRINT REPORT # 1 **
4125  ZL=60:ZC=80:GOSUB 60000:REM** TOP OF PAGE RESET,80 C/L **
4126  GOSUB 60000:REM** PRINT REPORT # 2 **
etc.
```

*Program Listing 4*


M.CHRUMKA



# Inside Scripsit—Part III

by Craig A. Lindley

**Free your CPU from slow peripherals! Use your knowledge of the features and functions of Scripsit to write a spooler/despooler program.**

In the first two articles in this series I showed how Scripsit was internally constructed and how to modify it to add features and functions to enhance its operation. In this part I will show you how to write a spooler/despooler that frees your computer from printing so it can accomplish other tasks.

Spool stands for simultaneous peripheral output overlap. This technique is used in large computer systems to increase throughput. It frees the CPU from driving a slow peripheral device such as a printer. The data destined for the peripheral device is sent either to memory or a storage medium (spooled) until the peripheral device is ready for it. The data is then transferred to the peripheral device (despooled) as it is required at the rate that the device can accept it. By doing this, the CPU is free to process other higher priority tasks. In the TRS-80, the CPU can run other programs while the despooler is outputting characters to the line printer during the 25-microsecond interrupt time. The despooler is termed a foreground task because it operates during interrupt time. The program that is running (and interrupted by the despooler) is called the background task.

The Scripsit Spooler/Despooler consists of two programs. I didn't want to add any additional features to the Scripsit modification program presented in the last article because it would reduce the text buffer area. In-

*Program Listing 1. Spooler*

```
                ;***************************************
                ;***      PRINT SPOOLER UTILITY      ***
                ;***   SCRIPSIT FILE SPOOLER V1.0    ***
                ;***        NOVEMBER 11, 1981        ***
                ;***               BY                ***
                ;***        CRAIG A. LINDLEY         ***
                ;***************************************
                ;
                ;**********
                ;SYSTEM EQUATES
                ;**********
                ;
0033            CHROUT  EQU     0033H           ;CHAR OUT ROUTINE
4428            CLOSE   EQU     4428H           ;CLOSE FILE ROUTINE
01C9            CLRSCN  EQU     01C9H           ;CLEAR SCREEN ROUTINE
4020            CURSOR  EQU     4020H           ;CURSOR ADDR STORAGE
0060            DELAY   EQU     0060H           ;DELAY ROUTINE
4409            ERROUT  EQU     4409H           ;DOS ERROR OUT ROUTINE
4420            INIT    EQU     4420H           ;INITIALIZE DISK FILE
0049            KEYIN   EQU     0049H           ;KEYIN ROUTINE
442C            KILL    EQU     442CH           ;KILL DISK FILE
4430            LOAD    EQU     4430H           ;LOAD CMD FILE ROUTINE
FCFF            MEMSIZ  EQU     0FCFFH          ;DESPOOLER MEMORY
                                                ;PROTECT LIMIT
6BC8            OPCMDL  EQU     6BC8H           ;SCRIPSIT OUT MSG ROUTINE
402D            OPSYS   EQU     402DH           ;REENTER OP SYSTEM
441C            PARSE   EQU     441CH           ;PARSE FILESPEC ROUTINE
002B            SCAN    EQU     002BH           ;SCAN KEYS ROM ROUTINE
523F            WP      EQU     523FH           ;SCRIPSIT ENTRY POINT
4439            WRITE   EQU     4439H           ;DOS WRITE DISK FILE
                ;
7AA6                    ORG     7AA6H
                ;
```

Listing 1 continues

### The Key Box

**Model I**
**32K RAM**
**Assembly language**
**One disk drive**
**Scripsit, Editor/Assembler and Printer required**

stead I developed a new modification program for Scripsit with only the spool and underline functions added. I do all the text entry and editing with the modification program presented in the second article. This new modification program spools the printed result.

I had to implement the underline function in the spooler modification program so you can spool files that incorporate this feature. If you want to perform word processing while despooling you must also use this new program (without all of the enhancements) because the unmodified Scripsit program and the patch program presented in the last article turn off the interrupts, thereby stopping the despooling process.

**Spooler Operation**

The spooler program is shown in Program Listing 1. Execute the program as a DOS command file by entering SSPOOL. The program's execution begins at the label Start shown on the listing. When executed, the screen clears, the sign-on message is displayed and the file SCRIPSIT/LC is located and loaded. If you use only the uppercase version of the Scripsit program, change the file extension on line 500 of Listing 1 to /UC, for proper operation of this program.

After Scripsit is loaded, it is patched extensively to allow the spool program to function correctly. All three places in the Scripsit program that read the status of the parallel printer are patched with the following code:

```
LD    A,30H  ;Printer ready status
NOP          ;Fill empty byte
```

so the printer always appears ready to the Scripsit program.

All five places in the Scripsit program that output a byte in the A register to the printer are patched with the following code:

```
CALL SPOOL   ;Call Spooler subroutine
```

so the spooler program receives all printed output from the Scripsit program. The End routine is modified so it jumps to the CSPOOL routine, which closes the spool file before you leave Scripsit. After the spool file is closed, you are returned to DOS instead of the reboot in the standard Scripsit program. Four other important patches are applied to the memory resident Scripsit program to make it compatible with the despooler program:



✓See List of Advertisers on Page 451

Listing 1 continued

```
                 ;**********
                 ;SYSTEM STORAGE LOCATIONS
                 ;**********
                 ;
7AA6 CA7A        BUFPTR  DEFW    DISKIO+1        ;PTR TO SPOOL DATA BUFFER
7AA8 01          COUNT   DEFB    1               ;DATA BUFFER CHAR COUNT
                 ;
7AA9 53          DCB     DEFM    'SCRIPSIT/LC'   ;FILE DCB
7AB4 03                  DEFB    3
0014                     DEFS    20
                 ;
00FF             DISKIO  DEFS    255             ;DISK I/O BUFFER
                 ;
7BC8 50          FNAME   DEFM    'PRT1/DAT:0'    ;SPOOL FILE NAME
7BD2 0D                  DEFB    13
7BD3 00                  DEFB    0
                 ;
7BD4 3E          IN      DEFB    3EH             ;STATUS IN PATCH DATA
7BD5 30                  DEFB    30H
7BD6 00                  DEFB    0
                 ;
7BD7 21          MEM     DEFB    21H             ;LOAD MEMORY LIMIT
7BD8 FFFD                DEFW    MEMSIZ+256      ;INSTRUCTIONS
                 ;
7BDA CD          OUT     DEFB    0CDH            ;PRINT OUT PATCH DATA
7BDB 8E7F                DEFW    SPOOL
                 ;
7BDD 00          UNDFLG  DEFB    0               ;UNDERLINE FLAG
7BDE 0E          UNDOFF  DEFB    14              ;TURN UNDERLINE OFF
7BDF 0F          UNDON   DEFB    15              ;TURN UNDERLINE ON
                 ;
7BE0 10          EMSG2   DEFB    16              ;MSG LENGTH
7BE1 44                  DEFM    'DISK I/O ERROR !'
                 ;
7F62                     ORG     7F62H
                 ;
                 ;**********
                 ;CLOSE SPOOL FILE ROUTINE
                 ;**********
                 ;
7F62 3AA87A      CSPOOL  LD      A,(COUNT)       ;GET DATA COUNT
7F65 B7                  OR      A               ;IS IT ZERO ?
7F66 2811                JR      Z,CFILE         ;IF YES, CLOSE FILE
7F68 2F                  CPL                     ;FIND REMAINING AREA
7F69 3C                  INC     A               ;TRUE COUNT
7F6A 47                  LD      B,A             ;INTO "B"
7F6B 2AA67A              LD      HL,(BUFPTR)     ;GET BUFFER DATA PTR
7F6E 3600        FILL    LD      (HL),0          ;STORE INTO BUFFER
7F70 23                  INC     HL
7F71 10FB                DJNZ    FILL            ;FILL REST OF BUFFER
                 ;
                 ;WRITE LAST DISK RECORD
                 ;
7F73 11A97A              LD      DE,DCB          ;PT AT DCB
7F76 CD3944              CALL    WRITE
                 ;
                 ;NOW CLOSE THE FILE
                 ;
7F79 CDC901      CFILE   CALL    CLRSCN          ;CLEAR SCREEN
7F7C 11A97A              LD      DE,DCB          ;FILE DCB
7F7F CD2844              CALL    CLOSE
7F82 C32D40              JP      OPSYS           ;BACK TO DOS
                 ;
                 ;**********
                 ;KEYMOD ROUTINE - CAUSES SCRIPSIT TO POLL THE KEYBOARD
                 ;IN THE NORMAL MANNER PERIODICALLY FOR THE DESPOOLER.
                 ;**********
                 ;
7F85 CD2B00      KEYMOD  CALL    SCAN            ;POLL KEYS
7F88 110004              LD      DE,400H         ;RESTORE OVERLAID
7F8B C3EB5F              JP      5FEBH           ;CODE
                 ;
                 ;**********
                 ;SPOOLER ROUTINE
                 ;**********
                 ;
7F8E E5          SPOOL   PUSH    HL              ;SAVE REGISTERS
7F8F D5                  PUSH    DE
7F90 F5                  PUSH    AF              ;SAVE CHAR
7F91 2AA67A              LD      HL,(BUFPTR)     ;LOAD PTR TO BUFFER
7F94 77                  LD      (HL),A          ;STORE PRINT CHAR
7F95 23                  INC     HL              ;BUMP PTR
7F96 22A67A              LD      (BUFPTR),HL     ;STORE UPDATED VALUE
7F99 3AA87A              LD      A,(COUNT)       ;GET BUF CHAR COUNT
7F9C FEFF                CP      255             ;(CP -1)   BUF FULL ?
7F9E 2808                JR      Z,TODISK        ;IF FULL THEN
7FA0 3C                  INC     A               ;BUMP COUNTER
7FA1 32A87A              LD      (COUNT),A       ;STORE UPDATED COUNT
7FA4 F1          GONE    POP     AF              ;RESTORE CHAR
7FA5 D1                  POP     DE              ;RESTORE REGISTERS
7FA6 E1                  POP     HL
7FA7 C9                  RET
                 ;
                 ;IF DISK BUFFER IS FULL - WRITE TO DISK
                 ;
7FA8 11A97A      TODISK  LD      DE,DCB          ;FILE DCB
7FAB CD3944              CALL    WRITE           ;SCRIPSIT WRITE TO DISK
7FAE 2812                JR      Z,TOD1          ;IF WRITE OK THEN
7FB0 21E07B              LD      HL,EMSG2        ;DISK ERROR MSG
7FB3 CDC86B              CALL    OPCMDL          ;OUT MSG ON CMD LINE
7FB6 010000              LD      BC,0
7FB9 CD6000              CALL    DELAY           ;WAIT FOR VIEW
7FBC 31FC41              LD      SP,41FCH        ;STACK PTR
```

Listing 1 continues

- The disable-interrupt instruction located at 52C0H is overlayed with a NOP instruction to allow interrupts to be serviced during Scripsit's operation.
- A call is made to the ROM key scan routine at 2BH to couple the Scripsit program to the despooler program as required.
- The Debug vector at 4315H is cleared to prevent Debug from being entered when the break key is pressed.
- Scripsit's memory-size routine is patched to protect the despooler in high memory.

These four patches allow this Scripsit patch program to operate during the despooling process.

After all these patches are made, you will be asked the following three questions:

1) CREATE SPOOL FILE ON DRIVE (0*-3): ?
2) NUMBER OF COPIES (1*-9): ?
3) ARE THESE CORRECT (Y*/N): ?

The default answers are shown marked with the asterisk and are chosen by pressing enter. If you want to create the spool file on drive 0 and want just one copy of the spool file, press enter three times.

The answer to the first question determines on which drive the spool file will be stored. Spool files can be rather large so place them on a disk with a lot of free space. The default file name is: PRT1/DAT. This file will be created on the specified disk drive unless a file by that name already exists. If a file does exit, the next higher version-numbered file will be created, for example: PRT2/DAT. This version-number advancing process continues until the program determines that a new file has been created. This process prevents the spooler from overwriting an existing spool file. In any case, the program displays the filespec of the created spool file.

Your answer to the next question determines how many copies of the complete spool file will be printed by the despooler. The default is one copy, but you can generate up to nine copies. The despooler has the logic needed to print the specified number of copies requested. The final question is a verification of the previously input data. If you answer with anything besides N, you will be immediately transferred to a fully initialized and operational Scripsit program. If you answer N, you will be asked the questions again.

While in Scripsit you can use all of the commands available in the unmodified program with the addition of the underline feature. When you try to print a document, the printer will remain silent and the disk drives will run. Use the normal Load command to load the Scripsit files you want to spool and then use the Print command to spool them. You can spool as many documents as the disk space allows; you are not limited to one Scripsit document per spool file. For example, if you spool two different documents and had requested two copies of the spool file earlier, you will receive four documents. When you are finished spooling your documents, type End as you normally would and you will be returned to the operating system as usual.

**Despool Operation**

Enter despool to execute the despooler. You will be asked, "FILE: ?" which is your cue to input the filespec of the spool file you wish to despool. Press enter to use the default filespec: PRT1/DAT. The despooler program will search the active disk drives in your system for this file. The first file found with this name will be despooled. To specify a different filespec, just type it in and press enter. If the file you specified is found, it will be despooled. If not, you will be told so. After the file is found, the program shows you how many copies of the spool file you requested when it was spooled and then returns to the operating system. Then the printer will start and you will be free to do other tasks. You can run any programs that you wish, including Basic, as long as the interrupts are not disabled. If the interrupts are disabled and then turned back on again, you will not lose any printed data, but the printer will be silent for the duration that the interrupts were off.

Following are several points to remember about the despooler program's operation.

Do not reboot your computer while the despooler is printing. If you do, you will have to restart despooling of the spool file.

Do not try to print anything in the conventional manner while the despooler is operational. This will result in a mix of the despooled and the real-time print data. You can, however, spool additional Scripsit files while the despooler is operational.

During the time the despooler reads the disk, the keyboard will become momentarily disabled. Care must be taken to prevent characters you type from being lost during these times.

While the despooler is active, an as-

*Listing 1 continued*

```
7FBF C34752           JP      5247H           ;START OVER
                ;
7FC2 21C97A     TOD1  LD      HL,DISKIO       ;I/O BUFFER
7FC5 22A67A           LD      (BUFPTR),HL     ;INITIALIZE PTR AGAIN
7FC8 AF               XOR     A               ;CLEAR "A"
7FC9 32A87A           LD      (COUNT),A       ;COUNT=0
7FCC 18D6             JR      GONE            ;END
                ;
                ;**********
                ;UNDERLINE ON/OFF ROUTINE
                ;**********
                ;
7FCE E5         UNDRLN PUSH   HL              ;SAVE REG
7FCF FE40             CP      40H             ;@ ?
7FD1 2015             JR      NZ,UN2          ;IF NOT THEN
7FD3 21DD7B           LD      HL,UNDFLG       ;GET UNDERLINE FLAG
7FD6 CB46             BIT     0,(HL)          ;IS FUNCTION ON ?
7FD8 2808             JR      Z,UN1           ;IF NOT THEN
7FDA CB86             RES     0,(HL)          ;IF ON - TURN OFF
7FDC 21DE7B           LD      HL,UNDOFF       ;PT AT UNDERLINE OFF CODE
7FDF 7E               LD      A,(HL)          ;GET FUNCTION CHAR
7FE0 1806             JR      UN2
7FE2 CBC6       UN1   SET     0,(HL)          ;IF OFF - TURN ON
7FE4 21DF7B           LD      HL,UNDON        ;PT AT UNDERLINE ON CODE
7FE7 7E               LD      A,(HL)          ;GET FUNCTION CHAR
7FE8 CD8E7F     UN2   CALL    SPOOL           ;OUTPUT NORMAL CHAR TO
                                              ;THE SPOOL FILE
7FEB E1               POP     HL              ;RESTORE REG
7FEC C3745F           JP      5F74H           ;BACK TO SCRIPSIT
                ;
                ;**********
                ;THIS ROUTINE CHECKS EACH VIDEO OUTPUT CHAR TO SEE IF
                ;IT IS AN @ CHAR. IF IT IS THE VIDEO IS SENT AN
                ;UNDERSCORE CHAR INSTEAD.
                ;**********
                ;
7FEF FE40       VUNDRL CP     40H             ;@ CHAR ?
7FF1 2004             JR      NZ,VU1          ;IF NOT THEN
7FF3 2B               DEC     HL              ;BACK UP CURSOR
7FF4 365F             LD      (HL),5FH        ;REPLACE CHAR
7FF6 23               INC     HL
7FF7 FD3406     VU1   INC     (IY+6)
7FFA C3B25F           JP      5FB2H
                ;
                ;**********
                ;SYSTEM MESSAGES
                ;**********
                ;
7FFD 593C       MSG1  DEFW    3C59H
7FFF 54               DEFM    'TRS-80  MODEL I'
800E 03               DEFB    3
800F 953C       MSG2  DEFW    3C95H
8011 53               DEFM    'SCRIPSIT  FILE  SPOOLER'
8028 0D               DEFB    13
8029 03               DEFB    3
802A DE3C             DEFW    3CDEH
802C 56               DEFM    'V1.0'
8030 03               DEFB    3
8031 4D3F             DEFW    3F4DH
8033 43               DEFM    'COPYRIGHT (C) 1981 BY CRAIG A. LINDLEY'
8059 03               DEFB    3
805A 873F             DEFW    3F87H
805C 2A               DEFM    '** UNAUTHORIZED DUPLICATION IS '
807B 50               DEFM    'PROHIBITED BY LAW **'
808F 00               DEFB    0
                ;
8090 0D         MSG3  DEFB    13
8091 43               DEFM    'CREATE SPOOL FILE ON DRIVE (0*-3): '
80B4 00               DEFB    0
                ;
80B5 0D         MSG4  DEFB    13
80B6 43               DEFM    'CREATING FILE: '
80C5 00               DEFB    0
                ;
80C6 0D         MSG5  DEFB    13
80C7 4E               DEFM    'NUMBER OF COPIES (1*-9): '
80E0 00               DEFB    0
                ;
80E1 0D         MSG6  DEFB    13
80E2 41               DEFM    'ARE THESE CORRECT (Y*/N): '
80FC 00               DEFB    0
                ;
80FD 2A         MSG7  DEFM    '** SCRIPSIT SPOOLER PROGRAM **  '
811D 42               DEFM    'BY CRAIG A. LINDLEY'
                ;
                ;**********
                ;SYSTEM ERROR MESSAGES
                ;**********
                ;
8130 0D         EMSG1 DEFB    13
8131 2A               DEFM    '** SCRIPSIT PROGRAM NOT FOUND **'
8151 0D               DEFB    13
8152 00               DEFB    2
                ;
                ;**********
                ;PROGRAM SUBROUTINES
                ;**********
                ;
                ;INSTRUCTION MOVE ROUTINE
                ;
8153 E5         INSMV PUSH    HL              ;SAVE INSTRUCTION PTR
8154 010300           LD      BC,3            ;BYTES TO MOVE
8157 EDB0             LDIR                    ;MOVE IT
```

*Listing 1 continues*

terisk will blink in the upper right corner of the display at the end of every print line to indicate proper operation. At any time you can abort the despool process by holding down the left and right arrow keys simultaneously. The despooler will disconnect itself from the operating system and display an "abort" message. If the despooler runs to completion, a "print complete" message will appear on the display.

**Despooling Theory**

Study Program Listing 2. When executed, this program also begins execution at the label Start on the listing. The despooler first protects itself from Basic and DOS by modifying the DOS top of memory pointer at 4049H. This memory pointer is saved so the full memory area can be restored when the despool process is complete. The program then requests a filespec from the operator. A default is available. When a valid filespec is input, the file is opened and the first record is read into the disk I/O buffer located at FF00H. The first byte of the first buffer contains the number-of-copies byte which is then transferred into memory for later use. By setting the NRN (next record number) to 0000H, the next read will again read the first record of the file. The despooler then patches itself into the operating system interrupt task table and the keyboard scan routine. The address of the normal keyboard driver is saved for restoration later. It is used to modify the despool code so that calls can be made to it.

The interrupt task table in the operating system (located in SYS0/SYS at addresses 4500–4517H) is a list of routine addresses that are executed at interrupt time. Normally, this list is filled with the default address of 45A3H which tells the DOS interrupt handler not to do anything at the interrupt time. When you execute the DOS Debug, Trace or Clock functions, the operating system then places the addresses of these routines into the task list so they are executed each interrupt time. I placed a pointer to my own interrupt task (HANDLR) into the task table (at an ordinarily empty location) so it gets executed each interrupt time also. By doing this, no other interrupt tasks are disturbed so Debug, Clock and Trace can run along with the despooler if desired.

HANDLR checks the line-printer status to see if it is ready for another character and checks if another character is ready to be output to the printer. It also performs numerous housekeep-

*Listing 1 continued*

```
8159 E1                      POP     HL
815A C9                      RET
                     ;
                     ;SINGLE LINE MESSAGE OUTPUT ROUTINE
                     ;
815B 7E              LOUT    LD      A,(HL)          ;GET CHAR
815C B7                      OR      A               ;IS IT 0 ?
815D C8                      RET     Z               ;IF YES THEN
815E FE03                    CP      3               ;OTHER TERMINATOR
8160 C8                      RET     Z               ;IF YES THEN
8161 CD3300                  CALL    CHROUT          ;DISPLAY IT
8164 23                      INC     HL              ;NEXT CHAR
8165 18F4                    JR      LOUT            ;UNTIL FINISHED
                     ;
                     ;MULTIPLE LINE MESSAGE OUTPUT ROUTINE
                     ;
8167 D5              MSGOUT  PUSH    DE              ;SAVE REGISTERS
8168 C5                      PUSH    BC
8169 3E0F                    LD      A,15            ;CURSOR OFF CODE
816B CD3300                  CALL    CHROUT          ;OUTPUT IT
816E 112040          MSG01   LD      DE,CURSOR       ;CURR CURSOR STORAGE
8171 010200                  LD      BC,2            ;MOVE ADDRESS
8174 EDB0                    LDIR                    ;INTO CURSOR STORAGE
8176 CD5B81                  CALL    LOUT            ;DISPLAY MSG LINE
                                                     ;"A" REG HAS TERM. CHAR
8179 23                      INC     HL              ;PT AT NEXT SCREEN ADDR.
817A B7                      OR      A               ;END OF COMPLETE MSG ?
817B 20F1                    JR      NZ,MSG01        ;LOOP UNTIL FINISHED
817D 3E0E                    LD      A,14            ;CURSOR ON CHAR
817F CD3300                  CALL    CHROUT          ;DO IT
8182 C1                      POP     BC              ;RESTORE REGISTERS
8183 D1                      POP     DE
8184 C9                      RET
                     ;
                     ;**********
                     ;START OF THE MAIN PROGRAM
                     ;*********
                     ;
8185 CDC901          START   CALL    CLRSCN          ;CLEAR DISPLAY
8188 AF                      XOR     A               ;CLEAR "A"
8189 321543                  LD      (4315H),A       ;DEBUG OFF
818C 21FD7F                  LD      HL,MSG1         ;SIGN ON MSG
818F CD6781                  CALL    MSGOUT          ;DISPLAY IT
                     ;
8192 11A97A                  LD      DE,DCB          ;PT AT SCRIPSIT DCB
8195 CD3044                  CALL    LOAD            ;LOAD FILE
8198 280C                    JR      Z,CONT          ;IF NOT AVALIABLE THEN
819A CDC901                  CALL    CLRSCN          ;CLEAR DISPLAY
819D 213081                  LD      HL,EMSG1        ;NOT FOUND MSG
81A0 CD5B81                  CALL    LOUT            ;DISPLAY IT
81A3 C32D40                  JP      OPSYS
                     ;
81A6 CD2844          CONT    CALL    CLOSE           ;CLOSE FILE
                     ;
                     ;OVERLAY SCRIPSIT SIGN ON MSG WITH A NEW ONE
                     ;
81A9 21FD80                  LD      HL,MSG7         ;PT AT NEW SIGN ON MSG
81AC 11F757                  LD      DE,57F7H
81AF 013300                  LD      BC,51
81B2 EDB0                    LDIR                    ;MOVE INTO PLACE
                     ;
                     ;**********
                     ;PATCH SCRIPSIT
                     ;**********
                     ;
                     ;CHANGE LOCATION OF TXTBUF TO MAKE ROOM FOR SPOOLER
                     ;
81B4 21FD7F                  LD      HL,MSG1         ;START OF NEW TXTBUF
81B7 227752                  LD      (5277H),HL
81BA 221654                  LD      (5416H),HL
81BD 223255                  LD      (5532H),HL
81C0 229359                  LD      (5993H),HL
81C3 22CB59                  LD      (59CBH),HL
81C6 22B65D                  LD      (5DB6H),HL
81C9 221663                  LD      (6316H),HL
81CC 225263                  LD      (6352H),HL
81CF 22DF66                  LD      (66DFH),HL
81D2 22B469                  LD      (69B4H),HL
81D5 22256E                  LD      (6E25H),HL
81D8 221A74                  LD      (741AH),HL
81DB 222874                  LD      (7428H),HL
                     ;
                     ;INSERT LONG JUMP CODE FOR PATCH INSERTION INTO SCRIPSIT
                     ;
81DE 3EC3                    LD      A,0C3H          ;LONG JMP CODE
81E0 32E85F                  LD      (5FE8H),A       ;PATCH KEYBOARD ROUTINE
81E3 329E7A                  LD      (7A9EH),A       ;UNDERLINE ROUTINE
81E6 32AF5F                  LD      (5FAFH),A       ;VIDEO UNDERSCORE
                     ;
                     ;STORE PATCH ROUTINE ADDRESSES IN SCRIPSIT
                     ;
81E9 21857F                  LD      HL,KEYMOD
81EC 22E95F                  LD      (5FE9H),HL
81EF 21CE7F                  LD      HL,UNDRLN
81F2 229F7A                  LD      (7A9FH),HL
81F5 21EF7F                  LD      HL,VUNDRL
81F8 22B05F                  LD      (5FB0H),HL
                     ;
81FB AF                      XOR     A               ;CLEAR "A"
81FC 32C052                  LD      (52C0H),A       ;NOP DI INSTRUCTION
                     ;
                     ;PATCH PRINTER I/O TO SPOOLER
                     ;
```

*Listing 1 continues*

*Listing 1 continued*

```
81FF 21D47B          LD      HL,IN            ;PT AT STATUS IN INST.
8202 11635F          LD      DE,5F63H         ;PATCH POINT #1
8205 CD5381          CALL    INSMV
8208 113F66          LD      DE,663FH         ;PATCH POINT #2
820B CD5381          CALL    INSMV
820E 115066          LD      DE,6650H         ;PATCH POINT #3
8211 CD5381          CALL    INSMV
8214 21DA7B          LD      HL,OUT           ;PT AT PRT OUT INST.
8217 114452          LD      DE,5244H         ;PATCH POINT #4
821A CD5381          CALL    INSMV
821D 115E66          LD      DE,665EH         ;PATCH POINT #5
8220 CD5381          CALL    INSMV
8223 112267          LD      DE,6722H         ;PATCH POINT #6
8226 CD5381          CALL    INSMV
8229 11977A          LD      DE,7A97H         ;PATCH POINT #7
822C CD5381          CALL    INSMV
                ;
                ;PATCH SCRIPSIT MEMORY SIZE ROUTINE
                ;
822F 21D77B          LD      HL,MEM           ;PT AT MEM LIMIT INST.
8232 116052          LD      DE,5260H         ;PATCH POINT #8
8235 CD5381          CALL    INSMV
                ;
                ;PATCH SCRIPSIT END ROUTINE
                ;
8238 21627F          LD      HL,CSPOOL        ;CLOSE SPOOL FILE
823B 229565          LD      (6595H),HL       ;STORE IN "END" ROUTINE
                ;
                ;***********
                ;ON WITH THE SHOW
                ;***********
                ;
823E CDC901     INPARM CALL  CLRSCN           ;CLEAR SCREEN
8241 211180          LD      HL,MSG2+2        ;PROGRAM NAME
8244 CD5B81          CALL    LOUT
8247 219080          LD      HL,MSG3          ;DRIVE # MSG
824A CD5B81          CALL    LOUT
824D CD4900          CALL    KEYIN            ;GET RESPONSE
8250 FE0D            CP      13               ;DEFAULT ?
8252 280E            JR      Z,IN1            ;IF YES THEN
8254 FE30            CP      '0'              ;CHECK RANGE
8256 38E6            JR      C,INPARM
8258 FE34            CP      '4'
825A 30E2            JR      NC,INPARM        ;IF IN RANGE CONTINUE
825C CD3300          CALL    CHROUT           ;DISPLAY SELECTION
825F 32D17B          LD      (FNAME+9),A      ;STORE IN FILENAME
8262 21C87B     IN1  LD      HL,FNAME         ;PT AT FILENAME
8265 11A97A          LD      DE,DCB           ;DCB STORAGE
8268 CD1C44          CALL    PARSE            ;MOVE INTO PLACE
826B 21C97A          LD      HL,DISKIO        ;DISK I/O BUFFER
826E 0600            LD      B,0              ;LRL=256
8270 CD2044          CALL    INIT             ;INITIALIZE FILE
8273 C28182          JP      NZ,ERROR         ;IF PROBLEM THEN
8276 381E            JR      C,NEW            ;IF NEW FILE THEN
8278 CD2844          CALL    CLOSE            ;ELSE CLOSE OLD FILE
827B 21CB7B          LD      HL,FNAME+3       ;PT AT FILE VER #
827E 34              INC     (HL)             ;UPDATE IT
827F 18E1            JR      IN1              ;TRY AGAIN FOR NEW FILE
                ;
                ;REPORT DISK I/O ERROR
                ;
8281 F5         ERROR PUSH   AF               ;SAVE ERROR CODE
8282 3E0D            LD      A,13             ;CR TO DISPLAY
8284 CD3300          CALL    CHROUT
8287 F1              POP     AF               ;RESTORE ERROR CODE
8288 F6C0            OR      0C0H             ;SHORT ERROR MSG
828A CD0944          CALL    ERROUT
828D 11A97A     DONE LD      DE,DCB           ;PT AT DISK DCB
8290 CD2844          CALL    CLOSE            ;CLOSE FILE
8293 C32D40          JP      OPSYS            ;BACK TO OS
                ;
                ;JUMP HERE WHEN NEW FILE IS CREATED
                ;
8296 21B580     NEW  LD      HL,MSG4          ;FILE CREATE MSG
8299 CD5B81          CALL    LOUT
829C 21C87B          LD      HL,FNAME         ;PT AT FILE NAME
829F CD5B81          CALL    LOUT             ;DISPLAY IT
82A2 21C680     NEW1 LD      HL,MSG5          ;PT AT # OF COPYS MSG
82A5 CD5B81          CALL    LOUT
82A8 CD4900          CALL    KEYIN            ;INPUT 0-9
82AB CD3300          CALL    CHROUT           ;DISPLAY ANSWER
82AE FE0D            CP      13               ;DEFAULT ?
82B0 280A            JR      Z,NEW2           ;IF YES THEN
82B2 D630            SUB     30H              ;CHK LOWER BOUND
82B4 38EC            JR      C,NEW1           ;IF < 30H THEN ASK AGAIN
82B6 FE0A            CP      0AH              ;CHK UPPER BOUND
82B8 30E8            JR      NC,NEW1          ;IF > 39H THEN ASK AGAIN
82BA 1802            JR      NEW3
82BC 3E01       NEW2 LD      A,1              ;DEFALUT = 1 COPY
82BE 32C97A     NEW3 LD      (DISKIO),A       ;STORE # IN
                                              ;DISK I/O BUF
82C1 21E180          LD      HL,MSG6          ;CORRECT MSG
82C4 CD5B81          CALL    LOUT
82C7 CD4900          CALL    KEYIN            ;GET RESPONSE
82CA CBAF            RES     5,A              ;CONV TO UC
82CC FE4E            CP      'N'              ;ANSWER NO ?
82CE C23F52          JP      NZ,WP            ;IF NOT EXECUTE SCRIPSIT
                ;
82D1 11A97A          LD      DE,DCB           ;IF INCORRECT KILL FILE
82D4 CD2C44          CALL    KILL
82D7 C33E82          JP      INPARM           ;GO ASK AGAIN
                ;
                ;
8185                 END     START
00000 TOTAL ERRORS
```

ing tasks including incrementing and decrementing pointers and counters. It also polls a flag (Flag) to determine if the byte to be output is the number-of-copies byte and if so, skips it.

The other interface for the despooler is the keyboard driver. Every time the keyboard is scanned for an input character, all other despooler housekeeping tasks are performed. During these times the spool file is read into the disk I/O buffer if more characters are needed for the printer. The abort function is also checked during each of the keyboard scan times.

### Getting the Programs to Work

Type the source code provided into an editor/assembler. This makes it easy to relocate the despool program for your correct memory size. The despooler should reside at the top of your memory, with the ORGs for the various memory sizes shown in Listing 2. The spooler program must be assembled at the origin location shown in Listing 1 regardless of memory size. I used the filenames SSPOOL/CMD for the spooler program and DESPOOL/CMD for the despooler program. I called it SSPOOL, for Scripsit Spooler, to distinguish it from NSPOOL, which is a spooler I have also written to use with Basic and all other print sources that use the printer device control block for channeling of printer I/O. I will present this program in a future article. The despooler presented in this article works with both spoolers. ■

*Craig Lindley can be reached at P.O. Box 704, Colorado Springs, CO 80901.*

*Program Listing 2. Despooler*

```
                 ;***************************************
                 ;***     PRINT SPOOLER UTILITY       ***
                 ;***        DESPOOLER V1.0           ***
                 ;***       NOVEMBER 11, 1981         ***
                 ;***              BY                 ***
                 ;***       CRAIG A. LINDLEY          ***
                 ;***************************************
                 ;
                 ;**********
                 ;SYSTEM EQUATES
                 ;**********
                 ;
3840             ARROWS  EQU     3840H         ;CURSOR ARROWS KEY ADDR
022C             BLINK   EQU     022CH         ;BLINK "*" ROUTINE
0033             CHROUT  EQU     .0033H        ;CHAR OUT ROUTINE
4020             CURSOR  EQU     4020H         ;CURSOR STORAGE
0060             DELAY   EQU     0060H         ;ROM DELAY ROUTINE
4510             INT25   EQU     4510H         ;INTERRUPT TABLE ADDR
4016             KEYDCB  EQU     4016H         ;KEYBOARD DCB
05D9             LINEIN  EQU     05D9H         ;ROM LINE INPUT ROUTINE
44CF             LOUT    EQU     44CFH         ;DOS LINE OUT ROUTINE
4424             OPEN    EQU     4424H         ;OPEN DISK FILE
402D             OPSYS   EQU     402DH         ;REENTER DOS
441C             PARSE   EQU     441CH         ;PARSE FILESPEC ROUTINE
4436             READ    EQU     4436H         ;READ DISK FILE
4049             TOPMEM  EQU     4049H         ;DOS TOP OF MEM PTR
                 ;
FD20                     ORG     0FD20H
                 ;
                 ;**********
                 ;SYSTEM STORAGE LOCATIONS
                 ;**********
                 ;
0020             DCB     DEFS    32            ;FILESPEC STORAGE
FD28             EOF     EQU     DCB+8         ;EOF STORAGE IN DCB
FD2C             ERN     EQU     DCB+12        ;END REC. NUMBER IN DCB
FD2A             NRN     EQU     DCB+10        ;NEXT REC. NUMBER IN DCB
FD40 00FF        BUFPTR  DEFW    DISKIO        ;POINTER TO DATA BUFFER
FD42 0000        CHRCNT  DEFW    0             ;BUFFER CHAR COUNT
0001             COPYS   DEFS    1             ;# OF COPIES STORAGE
0001             EOFCNT  DEFS    1             ;EOF SECTOR BYTE COUNT
FD46 00          FLAG    DEFB    0             ;1ST CHAR OF FILE FLAG
                 ;
FD47 50          FNAME   DEFM    'PRT1/DAT'    ;DEFAULT FILESPEC
FD4F 0D                  DEFB    13
FD50 00                  DEFB    0
                 ;
0002             KEYDVR  DEFS    2             ;KEYBOARD DVR STORAGE
0002             MEMSIZ  DEFS    2             ;TRUE MEM SIZE STORAGE
0002             SAVTSK  DEFS    2             ;INTERRUPT TASK STORAGE
FD57 0000        RECNUM  DEFW    0             ;CURRENT SECTOR RECORD
                 ;
                 ;**********
                 ;SYSTEM MESSAGES
                 ;**********
                 ;
FD59 46          MSG1    DEFM    'FILE: '
FD5F 03                  DEFB    3
                 ;
FD60 46          MSG2    DEFM    'FILE NOT FOUND !'
FD70 0D                  DEFB    13
                 ;
FD71 46          MSG3    DEFM    'FILE COPIES: '
FD7E 03                  DEFB    3
                 ;
FD7F 41          MSG4    DEFM    'ABORTED'
FD86 0D                  DEFB    13
                 ;
FD87 50          MSG5    DEFM    'PRINT COMPLETE'
FD95 0D                  DEFB    13
                 ;
                 ;**********
                 ;START OF THE MAIN PROGRAM
                 ;**********
                 ;
FD96 2A4940      START   LD      HL,(TOPMEM)   ;GET TRUE TOP OF MEM
FD99 2253FD              LD      (MEMSIZ),HL   ;STORE FOR LATER
FD9C 2116FD              LD      HL,DCB-10     ;MEMORY PROTECT LIMIT
FD9F 224940              LD      (TOPMEM),HL   ;DOS TOP OF MEM PTR
                 ;
FDA2 2120FD      ASK     LD      HL,DCB        ;PT AT INPUT BUFFER
FDA5 E5                  PUSH    HL            ;SAVE ADDRESS
FDA6 0620                LD      B,32          ;STORE 32
FDA8 3E20                LD      A,20H         ;SPACE CODES FOR DCB
FDAA 77          FILL    LD      (HL),A        ;STORE SPACE
FDAB 23                  INC     HL
FDAC 10FC                DJNZ    FILL          ;DO ALL 32
                 ;
FDAE 2159FD              LD      HL,MSG1       ;SIGN ON MSG
FDB1 CDCF44              CALL    LOUT          ;DISPLAY MSG
FDB4 E1                  POP     HL            ;RESTORE BUF ADDRESS
```

*Listing 2 continues*

*Listing 2 continued*

```
FDB5 0620     01010          LD      B,32          ;32 CHAR MAX FILESPEC
FDB7 CDD905   01020          CALL    LINEIN        ;GET FILESPEC
FDBA 78       01030          LD      A,B           ;GET INPUT LENGTH
FDBB B7       01040          OR      A             ;USER FILESPEC ?
FDBC 200C     01050          JR      NZ,DOPEN      ;IF YES THEN
              01060 ;
              01070 ;USE DEFAULT FILESPEC
              01080 ;
FDBE 1147FD   01090          LD      DE,FNAME      ;PT AT DEFAULT SPEC
FDC1 EB       01100          EX      DE,HL
FDC2 E5       01110          PUSH    HL            ;PT AT FNAME
FDC3 CD1C44   01120          CALL    PARSE         ;MOVE SPEC INTO DCB
FDC6 E1       01130          POP     HL
FDC7 CDCF44   01140          CALL    LOUT          ;DISPLAY DEFAULT
              01150 ;
              01160 ;**********
              01170 ;OPEN THE SPOOL FILE
              01180 ;**********
              01190 ;
FDCA 2100FF   01200 DOPEN    LD      HL,DISKIO     ;DISK I/O BUFFER
FDCD 1120FD   01210          LD      DE,DCB        ;PT AT DCB
FDD0 0600     01220          LD      B,0           ;LRL=0
FDD2 CD2444   01230          CALL    OPEN          ;OPEN SPOOL FILE
FDD5 280F     01240          JR      Z,CONT        ;IF ALLRIGHT THEN
FDD7 2160FD   01250          LD      HL,MSG2       ;DISK ERROR MSG
FDDA CDCF44   01260          CALL    LOUT          ;DISPLAY IT
FDDD 2A53FD   01270          LD      HL,(MEMSIZ)   ;GET FULL MEMORY SIZE
FDE0 224940   01280          LD      (TOPMEM),HL   ;STORE FOR DOS
FDE3 C32D40   01290 DOS      JP      OPSYS         ;BACK TO DOS
              01300 ;
              01310 ;**********
              01320 ;LINK UP INTERRUPT HANDLER
              01330 ;**********
              01340 ;
FDE6 2171FD   01350 CONT     LD      HL,MSG3       ;COPIES MSG
FDE9 CDCF44   01360          CALL    LOUT
FDEC 1120FD   01370          LD      DE,DCB        ;POINT AT DCB
FDEF CD3644   01380          CALL    READ          ;READ 1ST RECORD FOR
FDF2 3A00FF   01390          LD      A,(DISKIO)    ;FOR COPY COUNT
FDF5 3244FD   01400          LD      (COPYS),A     ;STORE # IN MEMORY
FDF8 C630     01410          ADD     A,30H         ;CONVERT TO ASCII
FDFA CD3300   01420          CALL    CHROUT        ;DISPLAY
              01430                                ;AND DISPLAY
FDFD 210000   01440          LD      HL,0          ;REWIND FILE DCB
FE00 222AFD   01450          LD      (NRN),HL      ;STORE IN DCB
              01460 ;
FE03 F3       01470          DI                    ;DISABLE INTERRUPTS
FE04 2A1045   01480          LD      HL,(INT25)    ;GET TABLE DATA
FE07 2255FD   01490          LD      (SAVTSK),HL   ;AND SAVE
FE0A 2128FE   01500          LD      HL,PRTINT     ;POINTER TO PRT ROUTINE
FE0D 221045   01510          LD      (INT25),HL    ;INTO INTERRUPT TABLE
FE10 3A28FD   01520          LD      A,(EOF)       ;EOF BYTE COUNT FROM DCB
FE13 3245FD   01530          LD      (EOFCNT),A    ;STORE IN MEMORY
FE16 2A1640   01540          LD      HL,(KEYDCB)   ;GET KEY DVR ADDR
FE19 2251FD   01550          LD      (KEYDVR),HL   ;STORE IN MEMORY
FE1C 228BFE   01560          LD      (KEY+1),HL    ;STORE FOR JUMP
FE1F 2163FE   01570          LD      HL,NEWDVR     ;KEY SCAN ROUTINE
FE22 221640   01580          LD      (KEYDCB),HL   ;INTO DCB
FE25 FB       01590          EI                    ;INTERRUPTS ON
FE26 18BB     01600          JR      DOS           ;BACK TO DOS WITH
              01610                                ;INTERRUPT TASK ENABLED
              01620 ;
              01630 ;**********
              01640 ;INTERRUPT HANDLER ROUTINE
              01650 ;**********
              01660 ;
FE28 2AFE     01670 PRTINT   DEFW    HANDLR        ;POINTER TO INTERRUPT
              01680                                ;HANDLER
FE2A F5       01690 HANDLR   PUSH    AF            ;SAVE REGS
FE2B E5       01700          PUSH    HL
FE2C DDE5     01710          PUSH    IX
FE2E DD2146FD 01720          LD      IX,FLAG       ;PT AT 1ST CHAR FLAG
FE32 3AE837   01730          LD      A,(37E8H)     ;READ PRT STATUS
FE35 CB7F     01740          BIT     7,A           ;RDY ?
FE37 2025     01750          JR      NZ,DONE       ;IF NOT THEN
FE39 2A42FD   01760          LD      HL,(CHRCNT)   ;GET CHAR COUNT
FE3C 7D       01770          LD      A,L           ;TEST FOR ZERO
FE3D B4       01780          OR      H
FE3E 281E     01790          JR      Z,DONE        ;IF ZERO THEN
FE40 2B       01800          DEC     HL            ;DEC CHAR COUNT
FE41 2242FD   01810          LD      (CHRCNT),HL   ;STORE NEW COUNT
FE44 2A40FD   01820          LD      HL,(BUFPTR)   ;GET CHAR ADDRESS
FE47 7E       01830          LD      A,(HL)        ;GET CHAR
FE48 DDCB0046 01840          BIT     0,(IX)        ;# OF COPYS BYTE ?
FE4C 2803     01850          JR      Z,SKIP        ;IF YES THEN
FE4E 32E837   01860          LD      (37E8H),A     ;OUTPUT TO PRINTER
FE51 DDCB00C6 01870 SKIP     SET     0,(IX)        ;SET FLAG
FE55 FE0D     01880          CP      13            ;DATA = CR ?
FE57 CC2C02   01890          CALL    Z,BLINK       ;BLINK "*"
FE5A 23       01900          INC     HL            ;BUMP PTR
FE5B 2240FD   01910          LD      (BUFPTR),HL   ;STORE NEW PTR
FE5E DDE1     01920 DONE     POP     IX            ;RESTORE REGS
FE60 E1       01930          POP     HL
FE61 F1       01940          POP     AF
FE62 C9       01950          RET
              01960 ;
              01970 ;**********
              01980 ;LINK TO KEYBOARD SCAN
              01990 ;**********
              02000 ;
FE63 3A4038   02010 NEWDVR   LD      A,(ARROWS)    ;ADDRESS OF <- AND ->
FE66 FE60     02020          CP      60H           ;MANUAL ABORT ?
FE68 2019     02030          JR      NZ,NEW1       ;IF NOT THEN
```

*Listing 2 continues*

*Listing 2 continued*

```
FE6A 21C03F           LD     HL,3FC0H       ;POS CUROSR ON LAST
FE6D 222040           LD     (CURSOR),HL    ;LINE
FE70 217FFD           LD     HL,MSG4        ;ABORT MSG
FE73 CDCF44           CALL   LOUT           ;DISPLAY MSG
FE76 CDB4FE           CALL   DISCON         ;DISCONNECT FROM OS
FE79 C5               PUSH   BC             ;SAVE REG
FE7A 010000           LD     BC,0           ;DELAY COUNT
FE7D CD6000           CALL   DELAY
FE80 C1               POP    BC
FE81 AF               XOR    A              ;"A" = 0
FE82 C9               RET
                ;
                ;**********
                ;JUMP HERE IF NOT MANUAL ABORT
                ;**********
                ;
FE83 2A42FD     NEW1  LD     HL,(CHRCNT)    ;GET CHAR COUNT
FE86 7D               LD     A,L            ;TEST FOR ZERO
FE87 B4               OR     H
FE88 2803             JR     Z,RDISK        ;IF ZERO - READ DISK
FE8A C30000     KEY   JP     $-$            ;TO NORMAL KEY ROUTINE
                ;
                ;**********
                ;READ DISK RECORD
                ;**********
                ;
FE8D 1120FD     RDISK LD     DE,DCB         ;PT AT DCB
FE90 CD3644           CALL   READ           ;READ RECORD
FE93 2840             JR     Z,ROK          ;IF ALLRIGHT THEN
                ;
                ;**********
                ;CHECK FOR MULTIPLE COPIES
                ;**********
                ;
FE95 FE1C             CP     1CH            ;EOF ERROR CODE ?
FE97 2017             JR     NZ,ABORT       ;IF NOT - REAL PROBLEM
FE99 2144FD           LD     HL,COPYS       ;PT AT # OF COPIES
FE9C 35               DEC    (HL)           ;-1
FE9D 2811             JR     Z,ABORT        ;IF ALL DONE THEN
FE9F 210000           LD     HL,0           ;PREPARE TO REWIND
FEA2 222AFD           LD     (NRN),HL       ;FILE TO BEGINNING
FEA5 2257FD           LD     (RECNUM),HL    ;CURRENT SEC RECORD = 0
FEA8 3A28FD           LD     A,(EOF)        ;GET EOF COUNT
FEAB 3245FD           LD     (EOFCNT),A
FEAE 18DA             JR     KEY
                ;
                ;**********
                ;IF ERROR OR FINISHED - DISCONNECT EVERYTHING
                ;**********
                ;
FEB0 218AFE     ABORT LD     HL,KEY         ;CONT ADDR ON STACK
FEB3 E5               PUSH   HL
FEB4 F3         DISCON DI                   ;DISABLE INTERRUPTS
FEB5 2A55FD           LD     HL,(SAVTSK)    ;GET INTERRUPT TASK
FEB8 221045           LD     (INT25),HL     ;BACK INTO OS
FEBB 2A51FD           LD     HL,(KEYDVR)    ;NORM KEY DRIVER
FEBE 221640           LD     (KEYDCB),HL    ;INTO DCB
FEC1 2A53FD           LD     HL,(MEMSIZ)    ;GET TRUE MEM SIZE
FEC4 224940           LD     (TOPMEM),HL    ;STORE FOR DOS
FEC7 21C03F           LD     HL,3FC0H       ;POS CURSOR ON LAST
FECA 222040           LD     (CURSOR),HL    ;LINE
FECD 2187FD           LD     HL,MSG5        ;COMPLETE MSG
FED0 CDCF44           CALL   LOUT           ;DISPLAY MSG
FED3 FB               EI                    ;INTERRUPTS ON
FED4 C9               RET
                ;
                ;**********
                ;READ OK SET UP CHRCNT AND RETURN
                ;**********
                ;
FED5 2100FF     ROK   LD     HL,DISKIO      ;PT AT DISK BUFFER
FED8 2240FD           LD     (BUFPTR),HL    ;RESET PTR TO BUFFER
FEDB 2A57FD           LD     HL,(RECNUM)    ;CURRENT REC #
FEDE ED5B2CFD         LD     DE,(ERN)       ;MAX RECORD #
FEE2 DF               RST    18H            ;ARE THEY EQUAL ?
FEE3 200B             JR     NZ,MORE        ;IF NOT EOF THEN
                ;
                ;**********
                ;IF SECTOR IS ZERO USE EOF COUNT NOT 256
                ;**********
                ;
FEE5 3A45FD           LD     A,(EOFCNT)     ;GET EOF BYTE COUNT
FEE8 6F               LD     L,A            ;INTO HL
FEE9 2600             LD     H,0
FEEB 2242FD           LD     (CHRCNT),HL    ;STORE FOR CHRCNT
FEEE 189A             JR     KEY
                ;
                ;**********
                ;NORMAL 256 BYTE SECTOR COUNT
                ;**********
                ;
FEF0 23         MORE  INC    HL             ;INC SECTOR COUNT
FEF1 2257FD           LD     (RECNUM),HL    ;STORE UPDATED VALUE
FEF4 210001           LD     HL,256         ;SECTOR DATA CHAR COUNT
FEF7 2242FD           LD     (CHRCNT),HL    ;STORE FOR CHRCNT
FEFA 188E             JR     KEY            ;EXIT
FF00                  ORG    0FF00H
                ;
FF00            DISKIO EQU   $
                ;
FD96                  END    START
00000 TOTAL ERRORS
```

# ESTATE PLANNING MODEL
# SAVE TIME AND MONEY

by using your TRS-80* Model II, 64 K computer for professional estate analysis.

- ☆ **Marital Deduction Formula Analysis Reports** analyze six formula bequests for both husband and wife for both orders of death.
- ☆ **Estate Tax Return Report** computerizes estate tax planning for decedents' estates.
- ☆ **Client-Ready Reports** printed.
- ☆ **State** (Ohio) **and Federal Taxes** computed.
- ☆ **Free Updates** to reflect legislative changes for one year.
- ☆ **Sold and maintained by author:** an attorney and estate planning professional.
- ☆ **ALREADY REVISED** to comply with 1981 TAX REDUCTION ACT.

**YES,** please send me more information on the **Estate Planning Model.**

Name ______________________

Name of Firm ______________________

Street ______________________

City ____________ State ______ Zip ______

Mail to: Beard Software Development Co.
59 Skyline Avenue
Canfield, Ohio 44406

*TRS-80 is a trademark of Tandy Radio Shack.

✓134

# DC/CMD
## DISK CATALOGUE

*Finally*, after more than nine months of development, a universal diskette management system. A software package designed to keep track of all of your diskettes and programs.

## $39.95

*Contains...*

- ★ Quick sort routine for sorting by program name, or program extension, or disk number, or a combination of inter-woven sorts.
- ★ 255 Disk/510 side storage capacity per data file.
- ★ Virtually unlimited printing and data access parameters.
- ★ Pays for itself the first week.

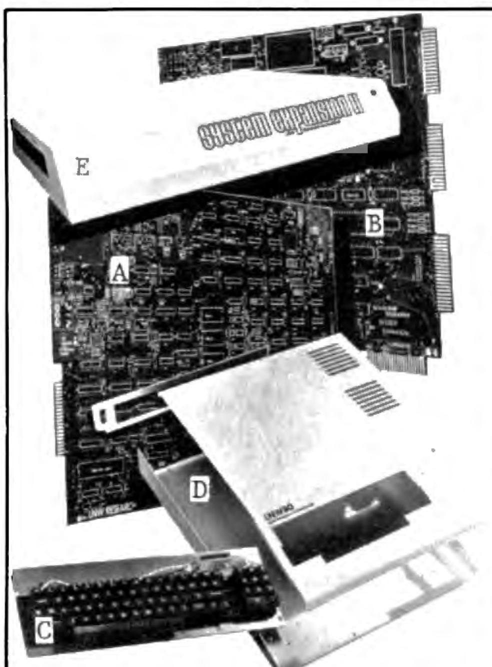
**(TRS MODELS I & III Only)**

**Rainbow** TM ✓265

P & P CORPORATION
PO Box 362 • HADDONFIELD, N J 08033

CALL TOLL-FREE **800-257-6170**
*in New Jersey* **609-428-3900**

# COMPUTER KITS – FROM $69.95




**LNW SEMI-KITS** can save you hundreds of dollars. By obtaining your own parts at the lowest possible cost and assembling the **LNW SEMI-KITS**, you can have the most highly acclaimed microcomputer in the industry – the **LNW80**. The **LNW SEMI-KITS** are affordable modules. You can start with a modest cassette system and expand to a full 4Mhz TRS-80 compatible system with 5 or 8 inch double density disks and color at any time.

**A. LNW80 CPU** - Made of high quality FR4 glass epoxy double sided circuit material, with plated-through holes and gold edge connector. It is fully solder-masked and silk screened. Here are just some of the outstanding features you will have when your **LNW80** CPU board is fully assembled: • 16K RAM • Color and black and white video • 480 x 192 high resolution graphics • 64 and 80 column video • 4 Mhz Z80A CPU • Upper and lower case display • 500 and 1000 baud cassette I/O -**$89.95**

**B. SYSTEM EXPANSION** - Expand the **LNW80** computer board, **TRS-80** and **PMC-80** computer with the following features: • 32K memory • Serial RS232C and 20Ma port • Real time clock • Parallel printer port • 5 inch single density disk controller • Expansion bus (screen printer port) • Onboard power supply • Solder-masked and silk screened legend - **$69.95** (tin plated contacts) - **$84.95** (gold plated contacts)

**C. KEYBOARD** - 74 key expanded professional keyboard - includes 12 key numeric keypad. Fully assembled and tested. - **$99.95**

**D. COMPUTER CASE** - This stylish instrument-quality solid steel case and hardware kit gives your **LNW80** that professional factory-built appearance. - **$84.95** Add $12.00 for shipping.

**E. SYSTEM EXPANSION CASE** - This stylish instrument-quality solid steel case and hardware kit gives your **SYSTEM EXPANSION** interface that professional factory-built appearance. - **$59.95** Add $10.00 for shipping.

**F. LNW80** CPU - HARD TO FIND PARTS KIT - **$82.00**

**G. LNW80 VIDEO** - HARD TO FIND PARTS KIT - **$31.00**

**H. SYSTEM EXPANSION** - HARD TO FIND PARTS KIT - **$27.50**

**I. LEVEL II ROM** set (6 chip set) - **$120.00**

**VISA** and **MasterCard** accepted. Add $3.00 for shipping plus $1.00 for each additional item. All shipments via UPS surface. Add $2.00 for U.S. Mail. Shipments outside continental U.S. funds must be U.S. dollars. Sufficient shipping costs must be included with payment.

**ORDERS & INFORMATION - (714) 544-5744**
**SERVICE - (714) 641-8850**

# LNW Research Corp.

2620 WALNUT Tustin, CA. 92680

✓33

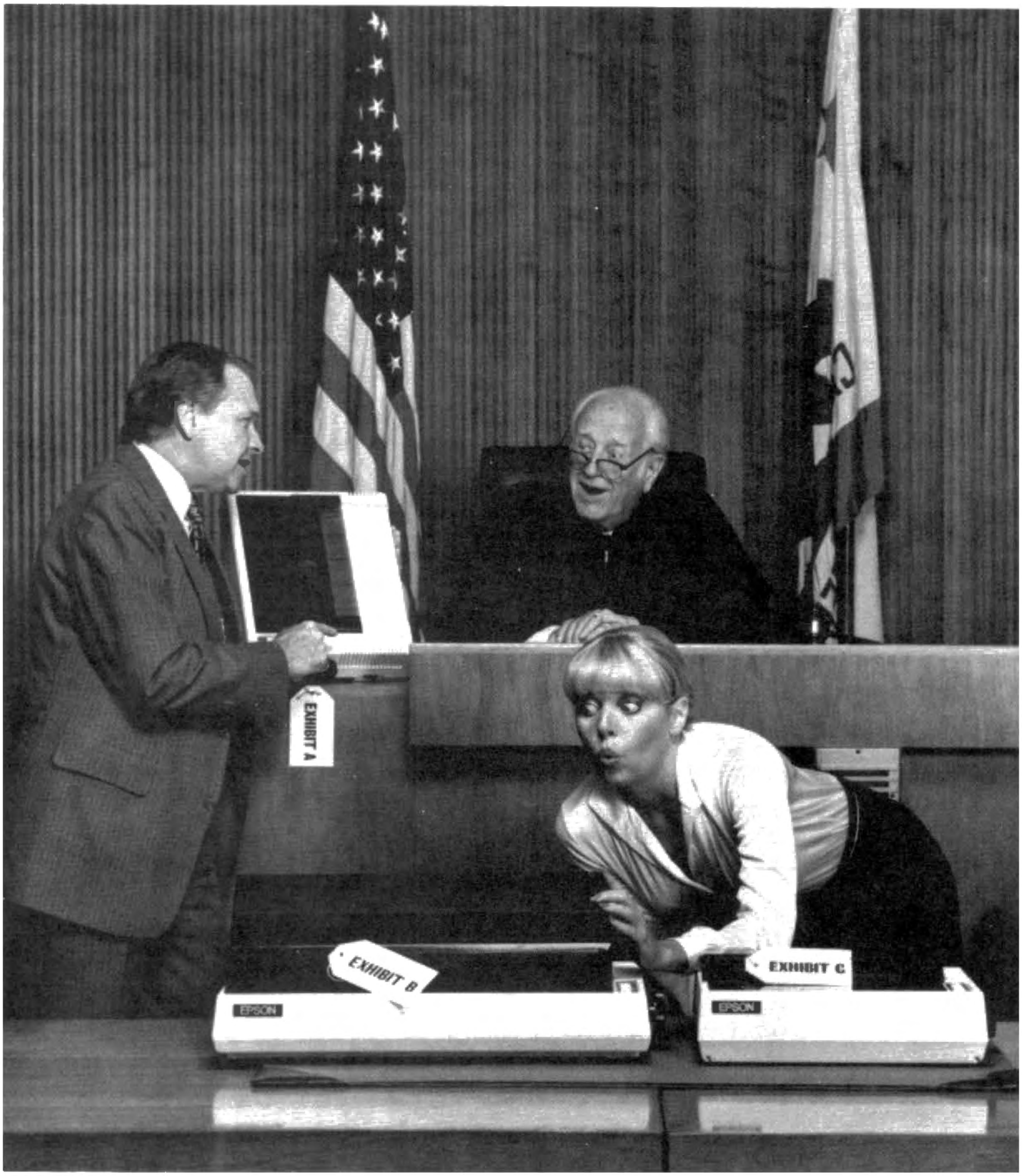


EXHIBIT A
EXHIBIT B
EXHIBIT C
EPSON
EPSON

✓See List of Advertisers on Page 451

LOAD 80

# A Matter of Compatibility

by Art Huston

**Make your program submissions to *80 Micro* compatible with other sytems with this short routine. Your readers will appreciate it.**

Imagine the frustration of a first-time reader who types in a long program only to come up with an undefined user function or Level III error. Maybe the author states, "I have a 16K Model I, but this program can easily be changed to work on a 48K Model III with disks." But he doesn't explain further, because he doesn't have the machine to test the change.

### The Problem

This occurs most often when the program is written in Basic with machine-language routines read from data statements and POKEd into RAM. All too often the author only makes provisions for the program run under his system.

For example, let's say an author has a 16K, Level II Model I and he POKEs a routine into high RAM. For a 68-byte routine, he would probably choose to POKE the routine in starting at 32700 and ending at 32767:

```
10 FOR X=32700 TO 32767:READ Y:POKE X,Y:NEXT
```

Then he would tell the Basic interpreter what address to branch to when encountering a USR statement. He would do this by POKEing the address into 16526 and 16527:

```
20 X=32700:X2=X/256:X1=X-X2*256:POKE16526,X1:POKE16527,X2
```

Later in the program he would have a line that would call the machine-language routine:

```
100 X=USR(0)
```

If a relative newcomer to computers tries to use this program on a 48K Model III under Disk Basic, he will meet up with only frustration and probably give up.

When he runs the program, it will quickly crash with a UL or syntax error. A listing of the program shows that some of the lines are pure gibberish, whereas before the run they were perfectly normal. What happened?

Our newcomer is unaware that Disk Basic takes up space in memory. A program that loads in at 17129 under Level II (17385 for Model III) loads in at around 28000 under Disk Basic. When it POKEs a routine in at 32000, it overwrites itself. The result is a scrambled listing.

The new reader perseveres. He summons a friend and between them they manage to move the POKE routine into high memory for 48K. Line 10 now reads:

```
10 FOR X=65468 TO 65535:READ Y:POKE X,Y:NEXT
```

When the program is run again, the screen says, "Overflow error in 10." The user begins to pull his hair out.

His friend says, "When you POKE above 32767, you have to subtract 65536 from the location you're POKEing." After much trial and error they come up with:

```
10 FORX=65468 TO 65535:READ Y:POKE X-65536,Y:NEXT
```

This seems to work just fine, and hope begins to dawn for our computer neophyte. Then, suddenly, the computer reports an undefined user function in 100. Faced with this ultimate insult, our user gives up.

How is our would-be computerist to know that a DEFUSR is all that's needed? He only needs to tell the Basic interpreter where the routine is located

```
10 REM ** UNIPOKE **
20 INPUT"Number of bytes to POKE";NB
30 Z2=PEEK(16561)+256*PEEK(16562)'Z2=last byte Basic will use
40 Z1=Z2-NB+1'Z1=beginning POKE location
50 Z=Z1-1:Z4=INT(Z/256):Z3=Z-Z4*256:POKE16561,Z3:POKE16562,Z4:
   'last byte used by Basic is one below start of routine
60 FOR Z=Z1 TO Z2:READ ZZ:POKE Z+(Z>32767)*65536,ZZ:NEXT
   'POKE in the routine
70 IFPEEK(16396)=201THEN90  'Test for Cassette Basic
80 DEFUSR=Z1+(Z1>32767)*65536:GOTO100  'Entry to USR for disk
90 Z4=INT(Z1/256):Z3=Z1-256*Z4:POKE16526,Z3:POKE16527,Z4
     'Entry to User for cassette
100 CLEAR50:
    'Clear string space. ROM routine corrects amt. of MEM
     and string storage to agree with 16561,16562
110 '**** Here Begins the Rest of the program ****
```

*Program Listing 1*

# If you're tired of playing games with your TRS-80...

## Turn it into a high power computer!

**Now you can give full CP/M capabilitites to your TRS-80.**

Announcing the new Compactor family from Hurricane Laboratories. We know limitations are a reality but we put on our thinking caps and did something about it. The new Compactor II (112K) gives CP/M 2.2 capabilities to the Model III without altering the TRSDOS and TRS-Basic environment. It's fast, it's easy to install and needs no extra hardware. The Compact IV enhances the video display to 80 x 24. In short, Compactors make your Model III an intelligent and powerful computer. So why not quit playing games and put that TRS-80 to work! It's as simple as saying, "...Compactors by Hurricane Labs."

**30-day money-back guarantee on Compactor II if not totally satisfied.**

**Write to us or ask your TRS-80 dealer for all the facts about the new Compactor Family from Hurricane Labs.**

185

5149 MOORPARK AVENUE, SUITE 105 **HURRICANE** Laboratories, Inc ™ SAN JOSE, CALIFORNIA • (408) 257-8676

See List of Advertisers on Page 451

and the program will run correctly:

```
20 DEFUSR = 65468 – 65536
```

Let's be kind to this individual and try to ease his way into programming. After all, POKEs, PEEKs, and DEFUSRs are advanced stuff that he won't use for years. The key to helping him is to make the programs as compatible as possible.

```
10 REM ** UNIPOKE compressed version**
20 INPUT"Number of bytes to POKE";NB
30 Z2=PEEK(16561)+256*PEEK(16562):Z1=Z2-NB+1:Z=Z1-1:Z4=INT(Z/256
):Z3=Z-Z4*256:POKE16561,Z3:POKE16562,Z4:FORZ=Z1TOZ2:READZZ:POKEZ
+(Z>32767)*65536,ZZ:NEXT
40 IFPEEK(16396)=201THENZ4=INT(Z1/256):Z3=Z1-256*Z4:POKE16526,Z3
:POKE16526,Z4ELSEDEFUSR=Z1+(Z1>32767)*65536
50 CLEAR50
```

*Program Listing 2*

```
10 REM ** UNIDEMO (Demonstrates how UNIPOKE works) **
20 CLS:CLEAR50:PRINT"MEM="MEM,"String Space starts at"PEEK(16544
)+256*PEEK(16545)+1,"
Last Byte avail to Basic="PEEK(16561)+256*PEEK(16562)
30 INPUT"Number of bytes to POKE";NB
40 Z2=PEEK(16561)+256*PEEK(16562)'Z2=last byte Basic will use
50 Z1=Z2-NB+1'Z1=beginning POKE location
60 Z=Z1-1:Z4=INT(Z/256):Z3=Z-Z4*256:POKE16561,Z3:POKE16562,Z4:
    'last byte used by Basic is one below start of routine
80 IFPEEK(16396)=201THEN100  'Test for Cassette Basic
90 DEFUSR=Z1+(Z1>32767)*65536:GOTO110  'Entry to USR for disk
100 Z4=INT(Z1/256):Z3=Z1-256*Z4:POKE16526,Z3:POKE16527,Z4
'Entry to User for cassette
110 CLEAR50:
    'Clear string space. ROM routine corrects amt. of MEM
     and string storage to agree with 16561,16562
120 PRINT"MEM="MEM,"String space starts at"PEEK(16544)+256*PEEK(
16545)+1"
Last Byte avail to Basic="PEEK(16561)+256*PEEK(16562)"
String spc moved lower to make room":PRINT@896,"Press ANY key"
130 IFINKEY$=""THEN130ELSERUN
```

*Program Listing 3*

**A Routine for All Systems**

A routine that would work with 16K, 32K, or 48K, disk or cassette, Model I or Model III would end the beginner's frustration. Such a routine would check for the top of memory and POKE the routine in. Then it would check for a disk or cassette system and do a DEFUSR or POKE the correct address in. Last, it would reset memory size to protect the machine-language routine.

To do these tasks we need the following information:

- Locations 16561 and 16562 point at the highest memory location usable by Basic on both the Model I and Model III. Changing these locations is equivalent to setting the memory size.
- IF PEEK(16396) = 201 then cassette Basic is in operation, else we are in Disk Basic.
- Cassette Basic on the Model I and III POKEs the starting address of the machine-language routine into 16526 and 16527.
- Disk Basic on the Model I and III uses a DEFUSR statement to determine the start of the machine-language routine.

Program Listing 1, UNIPOKE, is a routine to do all the above. Program Listing 2 is a compressed version, and Listing 3 demonstrates how it works. If you use UNIPOKE to POKE in your machine language, there should be no problems with DEFUSRs or moving POKEs higher or lower.

In Listing 1, line 20 is a dummy line that you should take out of your programs. It simply asks for the number of bytes you wish to POKE into memory. In your programs you should set NB equal to the number of bytes in your machine-language routine.

Line 30 sets Z2 equal to the highest location usable by Basic. Line 40 sets Z1 equal to the address at which to begin POKEing. Z2 will be the last location.

In line 50, Z is set equal to one byte below the routine. Then the value of Z is POKEd into 16561 and 16562. These locations tell Basic the highest location it can use and protect the machine-language routine from being overwritten.

Line 60 actually POKEs the routine in. The statement POKE Z + (Z>32767) *65536,ZZ looks complicated, but isn't. The phrase in the parentheses, Z>32767, is checked to see if it is true or false. If it is false, it has a value of 0 and the entire statement reduces to POKE Z,ZZ. If it is true, it has a value of -1 and the statement reduces to POKE Z–65536,ZZ. This routine is used to handle POKEs above 32767.

Line 70 checks PEEK(16396) to see if it is operating in cassette Basic. If so, the program jumps to 90.

Line 80 tells Disk Basic where to find the machine-language routine. Line 90 does the same for cassette Basic.

Line 100 clears string space. It must be a CLEARnnn statement, not just Clear. This allows the Basic interpreter to check the contents of 16561 and 16562 and adjust other pointers accordingly. It also frees up all variable space, so that the variables in UNIPOKE will not take up any room.

**Restrictions and Alternatives**

This routine respects the top-of-memory setting when you enter Basic—it will not overwrite anything already in high memory (like a keyboard debounce routine). This means, however, that each time the routine is executed it will set the top of memory lower and lower, eventually resulting in an out-of-memory error. If you use Run to restart a program, be sure it is Run nnnn, where nnnn is the first line number after this routine.

Using this routine will ensure that program users will not have to relocate machine language, change DEFUSRs to POKEs, or POKEs to DEFUSRs. It will make your program user-oriented in the best sense of the word.

This routine will not make all programs compatible with all systems. It will not make a program with disk input/output work correctly on a cassette system. If a Model I machine-language routine calls locations in ROM, it may crash on the Model III. Changes like these will always require you to tinker.

Programs that POKE the machine language into strings or into remark statements will only need lines 70–90 of UNIPOKE. These lines simply do a DEFUSR or POKEs, depending on whether the system is cassette or disk. In line 60, set Z1 equal to the location of the machine-language routine.

If you don't include this routine at the beginning of your program, include the changes for other systems somewhere in your article. ■

*Art Huston is a technical editor for 80 Micro.*

**NEW** for the Color Computer TRS-80

## 'COCOCASSETTE' SUBSCRIPTION SOFTWARE

*TRS-80 IS A TRADEMARK OF TANDY COMPANY

**ENJOY A MONTHLY COLLECTION OF 8-10 PROGRAMS!**

Including games, education, home finance and more on cassette for **as low as $5.00 a month!** Add some action and imagination to your Color Computer... Best of all, we do the work!

**PRICES**

| | |
|---|---|
| 1 YR (12 ISSUES) | $55.00 |
| 6 MO (6 ISSUES) | $30.00 |
| SINGLE COPIES | $ 6.00 |

--MICHIGAN RESIDENTS ADD 4% TO ORDER
OVERSEAS ADD $10.00 TO SUBSCRIPTION AND $1.00 TO SINGLE COPIES

616 396-7577

PROGRAMS ARE FOR EXTENDED BASIC MODEL ONLY. ISSUES ARE SENT FIRST CLASS

✓408

T&D SUBSCRIPTION SOFTWARE

master charge — SEND CHECK OR MONEY ORDER TO — VISA

**T & D SOFTWARE P.O. BOX 256-C • HOLLAND, MICH 49423**

---

| | |
|---|---|
| **EPSON** MX-100 III, MX 80 - FT III, MX 80 - III | **CALL FOR PRICE** |
| **Okidata** 80 Printer 80 CPS, List $449 | **$330.00** |
| **Okidata** 82A Printer 120 CPS, List $649 | **$440.00** |
| **Okidata** 83A Printer 120 CPS, List $1,119 | **$675.00** |
| **Okidata** 84 Printer (parallel) 132 column, 200 CPS | **$980.00** |
| **Okidata** 84 Printer (serial) 132 column, 200 CPS | **$1050.00** |
| **Okidata** 2350 Printer (parallel) 132 column, 350 CPS two color ribbon, List $2,800 | **$1860.00** |
| **Smith Corona** TP-1 Daisy-wheel printer, parallel or serial, 10 or 12 pitch, List $895 | **$615.00** |
| **BMC** 12AU 12 inch 15 MHZ, green on black monitor | **$95.00** |
| **BMC** 12EU 12 inch 20 MHZ, green monitor with non-glare screen | **$129.00** |
| **DC Hayes** Smartmodem (300 & 1200 BAUD) List $699 | **$505.00** |
| **DC Hayes** Smartmodem, List $279 | **$219.00** |
| **Diskette** storage case for 5¼ inch diskettes (holds 100 diskettes) | **$18.00** |
| **Practical** Peripherals Epson Buffer (parallel 16K) for Apple II or Radio Shack, List $159 | **$125.00** |
| **Percom** Doubler II | **$149.00** |
| **Practical** Peripherals Micro Buffer Inline 32K (specify serial or parallel) | **$244.00** |
| **Tandon** 40 Track Drive complete w/case & power supply | **$255.00** |

**CASH ONLY PRICE INCLUDES money order, certified company & personal check & COD. Personal and company checks take two weeks to clear.**

**Shipping & Handling Charge on light items $2.50. Prices subject to change without notice. NJ residents add 5% sales tax.**

**HOURS: 10 AM to 5 PM Weekdays. After 5 PM and Weekends Questionable.**

D&D SALES INC

P.O. BOX 1449 • BLOOMFIELD • NJ • 07003

**Send for COMPLETE CIRCULAR** ✓55 **201-751-8444**

---

# COPY MODEL III, SCRIPSIT/VISICALC®

# Only $20

Works with all model III Visicalc/Scripsit and allows unlimited backups of master disks. Works even if you're already backed up twice. Simple, 17 line program.

**And look at these other outstanding values!**

### APPARATS NEWDOS/80®, 2.0 $129.95

Convert Model I Scripsit and Visicalc to Model III, plus: Renumber program lines - move program lines - move blocks of program lines - duplicate program lines - selective variable clearing - program single stepping - memory sort multi-dimensioned arrays - swap variables - read and write to model I disks - reference keywords - spool printing - change disk speeds - disable BREAK and CLEAR - Much, much more!

**FREE "Layman's Guide to Newdos 80"** included with this purchase.

Use VISA - Mastercard - Money Orders - Allow 2 weeks for personal checks. Kansas residents add 3.5% sales tax.

### HAYES SMARTMODEM®.. $249.95

Talk with other computers over regular telephone lines. SMARTMODEM can auto dial - auto answer - repeat - allow for audio monitoring - programmable - is FCC approved direct-connect - full or half duplex capability, 0-300 baud - RS-232C interface - seven status LED's.

### VERBATIM® DATALIFE DISKETTES

5" diskettes come in a reusable plastic storage box.

**CALL TOLL FREE! 1 - 800 - 835-0071**

**IN KANSAS: 316 - 665-3611**

### ST80 III®..$129.95

Smart terminal communications package. Allows file uploading/downloading. Emulates many different terminals. Total operator control. (Written by Lance Micklus.)

### FREE Catalog price list

of all Sales Data Direct-Order merchandise available upon request. Super prices... Rapid delivery... Quality products.

SALES DATA, INC. ✓142

526 E. 4th Hutchinson, Kansas 67501
316/665-3611

**Dealer Inquiries Invited.**

Phone or write Sales Data, Inc.

# Tracking Lobster

by William F. Kaczor

**Your TRS-80 can't bait a hook, but it can help you trap a lobster using maps and graphs.**

Searching for lobster-pot buoys in ideal conditions is hard enough—forget it in bad weather. Not only that, but trying to remember where the pots are, when they were last hauled, what the catch was, and how good the catch was is no easy task when you have almost 100 pots spread over four square miles of ocean.

### A Better Way

Now my micro catches the lobsters. Well, not physically, but it does make lobster catching much easier.

Prior to this series of programs, my friend had been relying on his memory to keep track of how well each area was doing and where his pots were. (In "lobsterese," a pot is a trap set on the ocean bottom. This pot, or a string of pots tied together, is tied to a floating buoy marking its location.)

Also left to his memory was how good or bad the catch was in each area, the total caught for the day, and when the area was last worked. I was sure the computer's memory could do a much better job.

### Laying the Ground Rules

We decided the micro should generate a map of each area where he works his pots (six areas). Each map should show all navigational buoys and markers, compass headings, and, of course, where the pots are. The maps will also show the types of pots (single wire, double wire, wood) and the serial number of each pot.

The program should provide a tabulated printout grouped together by each of the six areas, which lists the following:

- It shows all the pots in that area, showing the code for each pot (explained later), the amount of "shorts" (lobsters too small) in each pot, the amount of "keepers" (legal-sized lobsters) in each pot, the total for the area, the percentage of keepers versus total pots worked for each area. Some of this data will be used for migration tracking and prediction of the most active areas.
- This printout also shows a total of keepers and shorts for all six areas combined, date worked, and a year-to-date total of keepers and shorts for all areas.
- If there is no data available for a given area, the printout should indicate it.

The program should also produce a bar-graph printout showing keepers and shorts per area for each date worked.

### Developing the Program

The program runs slowly, not because of Basic, but because my Centronics 737 printer is slow. Although it is not a graphics printer, all the necessary map features are recognizable. To circumvent its slowness, I set the program up so that once the week's data is entered, the whole process runs without operator intervention until complete.

The final outcome is nine separate programs producing eight printouts. Each program runs the next in proper order. All that is required of the operator is to enter the initial data, set the printer to the top of the page, and start the program.

### Computer-generated Maps

We used a pantograph to scale the actual marine maps to fit on single-sheet, tractor-feed printer paper. I set up a grid of 64 by 80 columns to aid in the programming design of each map. To ease the weekly entering of data coordinates for each pot, we divided each map into 20 sections labeled A–T (the printer outlines each sector with dots and labels each one) and allocated 20 possible pot positions in each section. That's 400 possible pot locations per map, six maps total.

Also included was the date, map and island titles, and to make things pretty, the name of the business (I changed the name in the example for privacy). See Figs. 1–6 for a look at each map. We included all necessary navigational beacons and markers along with arrow compass heading pointing north.

### The Pots

Each pot has a serial number. There are three different types of pots: single wire, double wire, and wood (coded SW, DW, and W respectively). We attach a location code to each pot showing what sector and position within the sector each pot is in. A pot code might read C10DW36, meaning sector C, pot

*The Key Box*

**Model I or III**
**48K RAM**
**Disk Basic**
**One Disk Drive**
**Printer**

# Expensive Expansion


*EXPENSIVE* – The *LNW System Expansion II* and built-in comes with a full 32K of 200ns RAM, RS232c 20 MA current loop serial interface. That's for starters. Next, consider our heavy gauge steel case, power indicator lamp, gold-plated connectors, FR-2 glass epoxy circuit board with solder mask and silk screen legends. Then there is the parallel printer port, screen printer port, real time clock, and extra heavy duty onboard power supply with over current protection, over voltage protection and thermal shutdown. If that's not enough then there is the floppy disk controller, guaranteed operation at a 4MHz CPU speed and our 6 month warranty. Every one of these features is *STANDARD*. This is true system expansion. You get every 'expensive' feature without spending more.

*CHEAP* – Our price is $399.95. Any way you compare, features or price, LNW's *System Expansion II* is the clear winner. The LNW System has been field tested for over two years with thousands of users. It works with any DOS, is 100% TRS-80 Model I compatible and it works 'right out of the box'. If there is any doubt in your mind as to whether you should buy ours or the 'other guys', just ask an LNW owner!

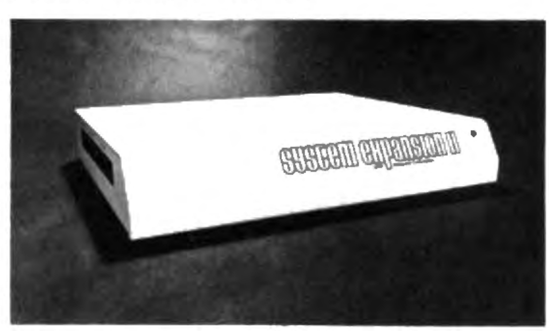
*WE ARE #1* – Number one in price, features, reliability, performance and delivery. LNW is committed to 'expensive' features and quality at reasonable prices. LNW is committed to support, thorough documentation, and reliability.

This has made us the number one manufacturer of system expansion units and accessories for the Model I computer.

*EXPANSION OPTION* – 8-inch drive capability is as easy as plugging in the *LNDoubler 5/8* option*. Now you can have any combination of single- or double-density, single- or double-sided, 8"* and/or 5" disks on-line! 8-inch disk storage increased to 591,360 bytes – 77-track single-sided, double-density or 1,182,720 bytes – 77-track double-density, double-sided.

The *LNDoubler's* unique 5/8 switch allows you to boot from 5- or 8-inch system disks and it's accessible from outside the interface. The $219.95 *LNDoubler 5/8* comes with a double-density disk operating system (DOS+ 3.3.9), *complete* with BASIC and utility programs... ready to run your software.

Each of your present 40-track, single-sided 5-inch drives will store up to 184,320 bytes (formatted storage) – that's an 80% increase in storage capacity for only half the cost of just one disk drive. With three 8-inch double-density, double-sided drives your Model I will have 3.75 Megabytes of online storage – that's more storage than a Model II or Model III!




## LNW Research Corp.

2620 WALNUT Tustin, CA. 92680
(714) 641-8850 (714) 544-5744

✓32

*8" drive operation requires special cable, 8" double density requires 3.55MHz CPU speed up modification or LNW 80 4MHz computer position 10, double wire, serial number 36. When printed on the map, it's shown with an asterisk, *C10DW36.

As each map is printed, the program checks the data file for that map, and by the code assigned to each pot, prints the pot in its proper place on the map.

## Entering Data

I wrote a file editor to modify data files. A typical session at each map area includes loading last week's data, entering the current date, telling the editor to set the keepers/shorts file to zero (previous data had already been added to the year-to-date file at the end of the last run).

Now, with a new file date, we delete or move any pots necessary, then enter the new date, keepers/shorts for each pot.

When writing the editor, I used INKEY$ as much as possible—the fewer the keystrokes the better. A listing to screen, showing the pots in each area and pot codes, aids in the efficient update of each file.

After entering each map's data, the file is saved to disk for later use. When all six map files have been updated, we start the first print routine and then go about our business. The routine will now run itself: printing out each map while accessing the proper data file putting the pots on each map in the proper location; printing out a copy of all the pot information on a tabulation sheet showing each pot code, shorts, keepers, and percentage caught versus traps worked, totals for each area, and year-to-date totals; printing out a bar graph of the shorts versus keepers for each map. This allows us, at a quick glance, to see how each area is doing.

## Success

Now it's easy to locate traps. We can study the maps, data printouts, and bar graph to see what areas are producing and which are not. Then decisions can be made for moving pots into other areas.

We hope to see migration trends from the gathered data or at least try to predict which areas to move the pots to and increase the catch.

My friend says life is much easier now. He spends much less time hunting for pots. On bad days when he can't go out, he studies his data.■

*Program Listing 1*

```
3 CLEAR3200:ON ERROR GOTO 3550:DEFSTRV,H,M,D,A:DEFINTU,B,O,X:DIMZ$(20,20):M="#":A="I":V=":":H=".":CLS:PRINT@465,"WORKING FILENAME....";CHR$(34);"BAKERS";CHR$(34):GOSUB3500:GOTO8
5 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;TAB(21)D3;TAB(30)D4;TAB(39)V;TAB(41)D5;TAB(50)D6;TAB(59)V;TAB(61)D7;STRING$(70-PEEK(16539),32);D8:RETURN
7 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:RETURN
8 GOSUB7;LPRINTSTRING$(80,H);GOSUB2000:S=1:L=1
10 LPRINT"(A)";TAB(3)"I";TAB(19)V;"(B)";TAB(39)V;"(C)";TAB(59)V;"(D)":U1=0:GOSUB1000
20 LPRINTTAB(2)"IIII";TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
30 LPRINTTAB(4)"I";TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U5=0:U6=0:GOSUB1000
40 LPRINTTAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;TAB(43)"* FL 4 SEC";TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
50 LPRINTTAB(5)"I";TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;TAB(44)"9 BELL";TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U5=1:U6=1:GOSUB1000
60 D1=" ":GOSUB5:GOSUB1000
70 LPRINTTAB(6)A;TAB(10);D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:GOSUB1000
80 GOSUB5:GOSUB1000
90 GOSUB5:GOSUB1000
100 GOSUB5:GOSUB1000
110 GOSUB5:GOSUB1000
120 LPRINTSTRING$(80,H):S=2:L=1
130 LPRINT"(E)";TAB(19)V;"(F)";TAB(39)V;"(G)";TAB(59)V;"(H)":U5=0:U6=0:GOSUB1000
140 GOSUB3000:LPRINTTAB(39)V;TAB(43)M;M;M;M;M;"  ";M;M;M;M;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
150 GOSUB3000:LPRINTTAB(39)V;TAB(42)M;TAB(48)M;M;TAB(54)M;M;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
160 GOSUB3000:LPRINTTAB(39)V;TAB(42)M;TAB(56)M;M;" ";V;" "D7;STRING$(70-PEEK(16539),32);D8:U1=0:GOSUB1000
170 LPRINT"* RN2";TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;TAB(42)M;TAB(53)"*";TAB(58)M;V;" "D7;STRING$(70-PEEK(16539),32);D8:U7=0:U1=1:GOSUB1000
180 GOSUB3000:LPRINTTAB(39)V;TAB(41)M;TAB(46)"LIGHTHOUSE";TAB(59)M;M;M;M;M;STRING$(70-PEEK(16539),32);D8:GOSUB1000
190 GOSUB3000:LPRINTTAB(39)V;TAB(41)M;STRING$(64-PEEK(16539),32);M;M;"    ";D8:GOSUB1000
200 GOSUB3000:LPRINTTAB(39)V;M;STRING$(66-PEEK(16539),32);M;" ";D8:GOSUB1000
210 GOSUB3000:LPRINTTAB(39)V;M;STRING$(67-PEEK(16539),32);M;"  ";D8:GOSUB1000
220 GOSUB3000:LPRINTTAB(39)V;" ";M;STRING$(66-PEEK(16539),32);M;"    ";D8:GOSUB1000
230 GOSUB3000:LPRINTTAB(39)V;" ";M;STRING$(67-PEEK(16539),32);M;"  ";D8
240 LPRINTSTRING$(39,H);M;M;M;TAB(48)"BAKERS ISLAND";STRING$(68-PEEK(16539),32);M;STRING$(11,H):GOSUB2000:S=3:L=1
250 LPRINT"(I)";TAB(19)V;"(J)";TAB(37)M;M;STRING$(69-PEEK(16539),32);M;" (L)":U8=0:U5=0:U6=0:U7=0:GOSUB1000
260 GOSUB3000:LPRINTTAB(39)M;M;M;TAB(50)DATE$;STRING$(69-PEEK(16539),32);M;M:U2=0:GOSUB1000
270 LPRINT" "D1;TAB(12)"* C3";TAB(19)V;" "D3;TAB(30);D4;TAB(40)M;STRING$(68-PEEK(16539),32);M;M:U8=1:U2=1:GOSUB1000
280 GOSUB3000:LPRINTTAB(39)M;STRING$(67-PEEK(16539),32);M;M;" "D8:GOSUB1000
290 GOSUB3000:LPRINTTAB(38)M;STRING$(65-PEEK(16539),32);M;M;" ";D8:GOSUB1000
300 GOSUB3000:LPRINTTAB(38)M;TAB(62)M;M;M;M;STRING$(70-PEEK(16539),32);D8:GOSUB1000
310 GOSUB3000:LPRINTTAB(38)M;TAB(58)M;M;M;M;M;STRING$(70-PEEK(16539),32);D8:U7=1:GOSUB1000
320 GOSUB3000:LPRINTTAB(38)M;TAB(57)M;M;V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
330 GOSUB3000:LPRINTTAB(38)M;TAB(56)M;"  ";V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
340 GOSUB3000:LPRINTTAB(37)M;TAB(55)M;"   ";V;" ";D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
350 GOSUB3000:LPRINTTAB(38)M;M;TAB(54)M;" (K)";V;" ";D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
360 LPRINTSTRING$(40,H);M;TAB(54)M;STRING$(25,H):GOSUB2000:U5=0:U6=0:S=4:L=1
370 LPRINT"(M)";TAB(19)V;"(N)";TAB(39)V;" ";M;TAB(47)M;"  ";M;M;M;M;" ";"(O)";TAB(59)V;"(P)":GOSUB1000
380 GOSUB3000:LPRINTTAB(39)V;" ";M;" ";M;M;" ";M;" ";M;M;TAB(59)V;" ";D7;STRING$(70-PEEK(16539),32);D8:U6=1:GOSUB1000
390 GOSUB3000:LPRINTTAB(39)V;" ";M;M;"  ";M;M;TAB(50)D6;TAB(59)V;" ";D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:GOSUB1000
400 GOSUB5:GOSUB1000
410 GOSUB5:GOSUB1000
420 GOSUB5:GOSUB1000
430 GOSUB5:GOSUB1000
440 GOSUB5:U1=0:U2=0:GOSUB1000
450 LPRINTTAB(5)M;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32)D8:GOSUB1000
460 LPRINT"* C5";M;" ";M;" POPE HEAD";TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" ";D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
470 LPRINT"<---";" ";M;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000
480 LPRINTSTRING$(80,H):S=5:L=1
490 LPRINT"(Q)";TAB(19)V;"(R)";TAB(39)V;"(S)";TAB(59)V;"(T)":GOSUB1000
500 GOSUB5:U3=0:GOSUB1000
510 LPRINT" "D1;TAB(10)D2;TAB(18)M;M;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
520 LPRINT" "D1;TAB(10)D2;TAB(17)M;"  ";M;" NORTH";TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U4=0:GOSUB1000
530 LPRINT" "D1;TAB(10)D2;TAB(17)M;"  ";M;" GOOSEBERRY";TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
540 LPRINT" "D1;TAB(10)D2;TAB(18)M;"  ";M;" ISLAND";TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:GOSUB1000
550 LPRINT" "D1;TAB(10)D2;TAB(18)M;M;M;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
560 GOSUB5:GOSUB1000
570 GOSUB5:GOSUB1000
580 GOSUB5:GOSUB1000
590 GOSUB5:GOSUB1000
600 LPRINTSTRING$(80,H):GOSUB7:GOTO 4000
1000 REM ASSIGN D1>D8 HERE.....
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=Z$(S+O1,B1):E$(2)=Z$(S+O1,B2):E$(3)=Z$(S+O2,B1):E$(4)=Z$(S+O2,B2):E$(5)=Z$(S+O3,B1):E$(6)=Z$(S+O3,B2):E$(7)=Z$(S+O4,B1):E$(8)=Z$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X))-2):GOTO 1090
1088      T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090      C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 LPRINT" "D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;:RETURN
3500 OPEN"I",1,"BAKERS/DAT":INPUT#1,DATE$
3510 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
3520 CLOSE:RETURN
3550 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"CAT"
```

## Program Listing 2

```
1 ONERRORGOTO3050:CLEAR3200:DEFSTRV,H,M,D,A:DEFINTU,B,O,X:DIMZ$(
20,20):M="0":A="I":V=":":H=".":CLS:PRINT@465,"WORKING FILENAME..
..";CHR$(34);"CAT";CHR$(34):GOSUB3000:GOTO9
2 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;:RETURN
3 LPRINTTAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59
)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN
4 LPRINTTAB(41)D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539
),32);D8:RETURN
5 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "
D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN

6 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;:RETURN
7 LPRINTTAB(39)V;" "D5;TAB(50)D6;TAB(59)V;:RETURN
8 GOSUB6:GOSUB7:RETURN
9 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:LPRINTSTRING$(80,H):GOSUB2000
:U2=0:GOSUB1000
10 LPRINT"(A)"TAB(8)A;TAB(19)V;"(B)";TAB(39)V;"(C)";TAB(59)V;"(D
)":S=1:L=1:GOSUB1000:GOTO20
11 LPRINTTAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:
RETURN
20 LPRINTTAB(1)D1;TAB(8)A;A;:GOSUB3:GOSUB1000
30 LPRINTTAB(1)D1;TAB(9)A;:GOSUB3:U7=0:U8=0:GOSUB1000
40 LPRINTTAB(1)D1;TAB(19)V;" ";D3;TAB(30)D4;TAB(39)V;" "D5;TAB(5
0)D6;TAB(59)V;STRING$(67-PEEK(16539),32);"*C5":U7=1:U8=1:GOSUB10
00
50 LPRINTTAB(1)D1;TAB(10)A;:GOSUB3:GOSUB1000
60 LPRINTTAB(1)D1;:GOSUB3:GOSUB1000
70 LPRINTTAB(1)D1;TAB(11)A;:GOSUB3:GOSUB1000
80 FORC=1TO4:GOSUB5:GOSUB1000:NEXTC:LPRINTSTRING$(80,H):LPRINT"(
E)";TAB(19)V;"(F)";TAB(39)V;"(G)";TAB(59)V;"(H)":S=2:L=1:GOSUB20
00:GOSUB1000
140 GOSUB5:U4=0:GOSUB1000
150 GOSUB2:LPRINTTAB(21)D3;TAB(28)M;M;" ";M;M;M;M;M;TAB(39)V;:GO
SUB4:GOSUB1000
160 GOSUB2:LPRINTTAB(27)M;TAB(30)M;TAB(36)M;TAB(39)V;:GOSUB4:U5=
0:GOSUB1000
170 GOSUB2:LPRINTTAB(23)M;M;M;M;TAB(35)M;TAB(39)V;M;M;TAB(50)D6;
TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
180 GOSUB2:LPRINTTAB(22)M;TAB(36)M;TAB(39)V;M;" ";M;M;:GOSUB11:G
OSUB1000
190 GOSUB2:LPRINTTAB(23)M;TAB(36)M;M;TAB(39)V;TAB(41)M;M;M;:GOSU
B11:U5=1:GOSUB1000
200 GOSUB2:LPRINTTAB(24)M;TAB(38)M;M;:GOSUB4:GOSUB1000
210 GOSUB2:LPRINTTAB(25)M;TAB(27)"CAT ISLAND";TAB(39)M;:GOSUB4:G
OSUB1000
220 GOSUB2:LPRINTTAB(24)M;TAB(40)M;TAB(42)D5;:GOSUB11:U5=0:GOSUB
1000
230 GOSUB2:LPRINTTAB(25)M;M;TAB(30)DATE$;TAB(41)M;M;M;M;:GOSUB11
:GOSUB1000
240 LPRINTSTRING$(27,H);M;M;M;M;TAB(44)M;M;STRING$(33,H)
250 LPRINT"(I)";TAB(19)V;"(J)";TAB(31)M;M;TAB(46)M;" (K)";TAB(59
)V;"(L)":GOSUB2000:U4=0:U5=0:S=3:L=1:GOSUB1000
260 GOSUB2:LPRINTTAB(21)D3;TAB(33)M;TAB(47)M;:GOSUB11:U2=0:GOSUB
1000
270 LPRINTTAB(1)D1;TAB(15)M;M;TAB(19)V;" "D3;TAB(34)M;M;M;M;M;TA
B(48)M;:GOSUB11:U4=1:GOSUB1000
280 LPRINTTAB(1)D1;TAB(15)M;M;TAB(19)V;" "D3;TAB(30)D4;TAB(39)M;
TAB(48)M;:GOSUB11:U2=1:GOSUB1000
290 GOSUB2:LPRINTTAB(21)D3;TAB(30)D4;TAB(40)M;TAB(48)M;M;TAB(51)
D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U1=0:U7=0:GOSUB1
000
300 LPRINT" RN2";TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(38)M;M;M
;TAB(50)M;" ";D6;TAB(58)M;M;M;STRING$(70-PEEK(16539),32);D8:U6=0
:U2=0:GOSUB1000
310 LPRINT"<---";TAB(12)M;M;TAB(19)V;" "D3;TAB(30)D4;TAB(38)M;"
";M;TAB(51)M;TAB(57)M;TAB(61)M;TAB(63)" ";M;M;M;M;"  "D8:U1=1:GO
SUB1000
320 LPRINTTAB(1)D1;TAB(11)M;" ";M;TAB(19)V;" "D3;TAB(30)D4;TAB(3
9)V;" "M;TAB(52)M;M;M;M;M;TAB(62)M;" ";M;"  ";M;M;" ";D8:GOSUB10
00
330 LPRINTTAB(1)D1;TAB(10)M;TAB(13)M;TAB(19)V;" "D3;TAB(30)D4;TA
B(39)V;TAB(42)M;TAB(63)M;" ";M;M;"   ";D8:U8=0:GOSUB1000
340 LPRINTTAB(1)D1;TAB(11)M;M;" ";M;TAB(19)V;" "D3;TAB(30)D4;TAB
(39)V;TAB(43)M;TAB(47)"CAT ISLAND";STRING$(64-PEEK(16539),32);M;
STRING$(69-PEEK(16539),32);M;M:GOSUB1000
350 LPRINTTAB(1)D1;TAB(13)M;M;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;
TAB(43)M;TAB(63)" ";M:GOSUB2000
360 LPRINTSTRING$(44,H);M;TAB(49)DATE$;TAB(63)M;STRING$(14,H)
370 LPRINT"(M)";TAB(19)V;"(N)";TAB(30)M;M;TAB(39)V;"(O)";TAB(45)
M;M;TAB(63)M;" (P)":U4=0:U5=0:U6=0:U7=0:S=4:L=1:GOSUB1000
380 GOSUB2:LPRINTTAB(21)D3;TAB(30)M;M;TAB(39)V;TAB(47)M;TAB(55)M
;TAB(63)M;"  ";M;M;"  ";D8:U4=1:GOSUB1000
390 GOSUB6:LPRINTTAB(39)V;TAB(45)M;M;" ";M;M;TAB(54)M;" ";M;TAB(
63)" ";M;M;"  ";M;" ";D8:GOSUB1000
400 GOSUB6:LPRINTTAB(39)V;TAB(45)M;M;TAB(50)M;M;TAB(54)M;TAB(57)
M;STRING$(68-PEEK(16539),32);M;" ";D8:GOSUB1000
410 GOSUB6:LPRINTTAB(39)V;TAB(52)M;M;TAB(58)M;M;M;M;STRING$(67-P
EEK(16539),32);M;"  ";D8:U5=1:U6=1:GOSUB1000
420 GOSUB8:LPRINTTAB(61)M;STRING$(66-PEEK(16539),32);M;M;M;" ";D
8:GOSUB1000
430 GOSUB8:LPRINTM;STRING$(66-PEEK(16539),32);M;"   ";D8:GOSUB10
00
440 GOSUB8:LPRINTM;STRING$(65-PEEK(16539),32);M;"    ";D8:GOSUB1
000
450 GOSUB8:LPRINTTAB(61)M;M;M;M;" ";M;M;"  ";D8:GOSUB1000
460 GOSUB8:LPRINTSTRING$(66-PEEK(16539),32);M;M;"  ";D8:GOSUB100
0
470 GOSUB8:LPRINTSTRING$(70-PEEK(16539),32);D8
480 LPRINTSTRING$(80,H):LPRINT"(Q)";TAB(19)V;"(R)";TAB(39)V;"(S)
";TAB(59)V;TAB(63)M;M;M;"  (T)":GOSUB2000:U7=0:U8=0:S=5:L=1:GOSU
B1000
500 GOSUB8:LPRINTTAB(62)M;"   ";M;M;" ";D8:GOSUB1000
510 GOSUB8:LPRINTTAB(62)M;STRING$(68-PEEK(16539),32);M;M;M;" ";D
8:U8=0:GOSUB1000
520 GOSUB8:LPRINTTAB(62)M;M;M;STRING$(71-PEEK(16539),32);M:GOSUB
1000
530 GOSUB8:LPRINTTAB(63)" ";M;STRING$(71-PEEK(16539),32);M:U8=1:
GOSUB1000
540 GOSUB8:LPRINTSTRING$(65-PEEK(16539),32)M;"    ";M;" ";D8:U8=
0:GOSUB1000
550 GOSUB8:LPRINTSTRING$(66-PEEK(16539),32)M;"    ";M:U8=1:GOSUB
1000
560 GOSUB8:LPRINTSTRING$(66-PEEK(16539),32)M;"   ";M;" ";D8:U7=1
:GOSUB1000
570 GOSUB8:LPRINTTAB(61)D7;STRING$(67-PEEK(16539),32)M;M;M;"  ";
D8:GOSUB2000:GOSUB1000
580 GOSUB5:GOSUB1000
590 GOSUB5:GOSUB1000
600 LPRINTSTRING$(80,H):LPRINTTAB(19)V;TAB(39)V;TAB(59)V:GOTO400
0
1000 REM LOAD D1>D8 HERE
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3
:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8
:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=Z$(S+O1,B1):E$(2)=Z$(S+O1,B2):E$(3)=Z$(S+O2,B1):E$(4)
=Z$(S+O2,B2):E$(5)=Z$(S+O3,B1):E$(6)=Z$(S+O3,B2):E$(7)=Z$(S+O4,B
1):E$(8)=Z$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X
))-2):GOTO 1090
1088       T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090       C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$
:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$
(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 OPEN"I",1,"CAT/DAT":INPUT#1,DATE$
3010 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
3020 CLOSE:RETURN
3050 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"EAGLE"
```

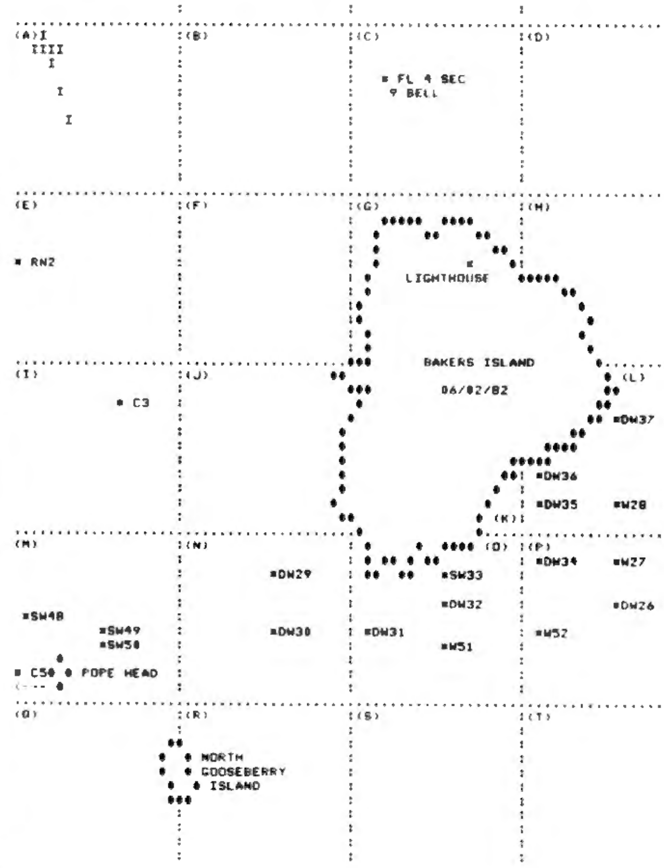



*Figure 1*

## Program Listing 3

```
1 CLEAR3200:ON ERROR GOTO 3050:DEFSTRV,H,M,D,A:DEFINTU,B,O,X:DIM
Z$(20,20):M="0":A="I":V=":":H=".":CLS:PRINT@465,"WORKING FILENAM
E....";CHR$(34);"EAGLE";CHR$(34):GOSUB3000:GOTO8
2 LPRINTTAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59
)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN
3 LPRINTTAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(
16539),32);D8:RETURN
4 LPRINTTAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN
5 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "
D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN

7 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;:RETURN
8 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:LPRINTSTRING$(80,H):GOSUB2000
```

*Listing 3 continues*

*Listing 3 continued*

```
:S=1:L=1
10 LPRINT"(A)";TAB(19)V;"(B)";TAB(39)V;"(C)";TAB(59)V;"(D)";U1=0
:U2=0:GOSUB1000
20 LPRINT"* R";TAB(8)A;:GOSUB2:GOSUB1000
30 LPRINT" N6";TAB(8)A;A;:GOSUB2:GOSUB1000
40 LPRINT"<---";TAB(9)A;:GOSUB2:GOSUB1000
50 GOSUB2:GOSUB1000
60 LPRINTTAB(10)A;:GOSUB2:GOSUB1000
70 GOSUB2:U1=1:U2=1:GOSUB1000
80 FORC=1TO4:GOSUB5:GOSUB1000:NEXTC:S=2:L=1:LPRINTSTRING$(62,H);
M;M;M;M;M;M;STRING$(12,H)
130 LPRINT"(E)";TAB(19)V;"(F)";TAB(39)V;"(G)";TAB(50)M;M;TAB(59)
V;" ";M;STRING$(68-PEEK(16539),32);M;M;M;" (H)":U5=0:U6=0:U7=0:U
8=0:GOSUB1000
140 GOSUB7:LPRINTTAB(21)D3;TAB(30)D4;TAB(39)V;TAB(44)M;M;M;M;M;M
;TAB(52)M;TAB(59)M;M;STRING$(70-PEEK(16539),32)M;M:GOSUB1000
150 GOSUB7:LPRINTTAB(21)D3;TAB(30)D4;TAB(39)V;M;M;M;M;TAB(52)M;T
AB(57)M;M;STRING$(71-PEEK(16539),32);M;M:GOSUB1000
160 GOSUB7:LPRINTTAB(21)D3;TAB(30)D4;TAB(37)M;M;M;TAB(53)M;M;M;M
;STRING$(72-PEEK(16539),32)M:U4=0:GOSUB1000
170 GOSUB7:LPRINTTAB(21)D3;TAB(35)M;M;STRING$(71-PEEK(16539),32)
;M:GOSUB1000
180 GOSUB7:LPRINTTAB(21)D3;TAB(33)M;M;STRING$(69-PEEK(16539),32)
;M;M:U8=1:GOSUB1000
190 GOSUB7:LPRINTTAB(21)D3;TAB(31)M;M;STRING$(67-PEEK(16539),32)
;M;M;" "D8:GOSUB1000
200 GOSUB7:LPRINTTAB(21)D3;TAB(30)M;STRING$(65-PEEK(16539),32)M;
M;"    ";D8:GOSUB1000
210 GOSUB7:LPRINTTAB(21)D3;TAB(28)M;M;TAB(63)M;M;STRING$(70-PEEK
(16539),32);D8:U3=0:GOSUB1000
220 GOSUB7:LPRINTTAB(27)M;TAB(61)M;M;STRING$(70-PEEK(16539),32);
D8:GOSUB1000
230 GOSUB7:LPRINTTAB(25)M;M;TAB(59)M;M;STRING$(70-PEEK(16539),32
);D8:GOSUB1000
240 LPRINTSTRING$(23,H);M;M;TAB(58)M;STRING$(20,H):S=3:L=1
250 LPRINT"(I)";TAB(19)V;" ";M;M;TAB(57)M;" ";V;"(L)":GOSUB2000:
U3=0:U4=0:U5=0:U6=0:GOSUB1000
260 LPRINTTAB(1)D1;TAB(10)D2;TAB(18)M;M;M;TAB(56)M;:GOSUB4:GOSUB
1000
270 LPRINTTAB(1)D1;TAB(10)D2;TAB(16)M;M;TAB(55)M;"(K)";V;" "D7;S
TRING$(70-PEEK(16539),32);D8:U2=0:GOSUB1000
280 LPRINTTAB(1)D1;TAB(13)M;M;M;TAB(54)M;:GOSUB4:GOSUB1000
290 LPRINTTAB(1)D1;TAB(10)M;M;M;TAB(53)M;:GOSUB4:GOSUB1000
300 LPRINTTAB(1)D1;TAB(8)M;M;TAB(52)M;:GOSUB4:U1=0:GOSUB1000
310 LPRINTTAB(5)M;M;M;TAB(22)"EAGLE ISLAND";TAB(50)M;M;:GOSUB4:G
OSUB1000
320 LPRINTTAB(4)M;TAB(49)M;:GOSUB4:U6=1:GOSUB1000
330 LPRINTTAB(3)M;TAB(22)DATE$;TAB(48)M;TAB(50)D6;:GOSUB4:GOSUB1
000
340 LPRINTTAB(4)M;TAB(46)M;M;TAB(50)D6;:GOSUB4:GOSUB1000
350 LPRINTTAB(4)M;TAB(44)M;M;TAB(50)D6;:GOSUB4:GOSUB1000
360 LPRINTH;H;H;M;TAB(42)M;M;STRING$(35,H):S=4:L=1
370 LPRINT"(M)";M;TAB(39)M;M;M;" (O)";TAB(59)V;"(P)":GOSUB1000
380 LPRINTTAB(3)M;TAB(37)M;M;V;TAB(50)D6;:GOSUB4:U5=1:GOSUB1000
390 LPRINTTAB(3)M;TAB(35)M;M;:GOSUB3:GOSUB1000
400 LPRINTTAB(4)M;TAB(33)M;M;"(N)";:GOSUB3:GOSUB1000
410 LPRINTTAB(4)M;M;M;TAB(32)M;:GOSUB3:GOSUB1000
420 LPRINTTAB(7)M;M;M;TAB(31)M;:GOSUB3:GOSUB1000
430 LPRINTTAB(10)M;TAB(30)M;:GOSUB3:U4=1:GOSUB1000
440 LPRINTTAB(10)M;TAB(29)M;TAB(31)D4;:GOSUB3:GOSUB1000
450 LPRINTTAB(10)M;TAB(28)M;TAB(30)D4;:GOSUB3:GOSUB1000
460 LPRINTTAB(11)M;TAB(27)M;TAB(30)D4;:GOSUB3:GOSUB1000
470 LPRINTTAB(11)M;TAB(26)M;TAB(30)D4;:GOSUB3:GOSUB1000
480 LPRINTSTRING$(11,H);M;TAB(25)M;STRING$(53,H):S=5:L=1
490 LPRINT"(Q)";TAB(11)M;TAB(21)M;M;M;M;" (R)";TAB(39)V;"(S)";TA
B(59)V;"(T)":GOSUB2000:U2=0:U3=0:GOSUB1000
500 LPRINTTAB(1)D1;TAB(11)M;M;;M;M;TAB(18)M;M;M;TAB(30)D4;:GOSUB
3:U3=1:GOSUB1000
510 LPRINTTAB(1)D1;TAB(14)M;M;M;M;:GOSUB2:GOSUB1000
520 LPRINTTAB(1)D1;:GOSUB2:GOSUB2000:GOSUB1000
530 FORC=1TO7:GOSUB5:GOSUB1000:NEXTC:LPRINTSTRING$(80,H)
540 LPRINT"*FL R4 SEC";TAB(19)V;TAB(39)V;TAB(59)V:GOTO4000
1000 REM ASSIGN D1>D8 HERE
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3
:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8
:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=Z$(S+O1,B1):E$(2)=Z$(S+O1,B2):E$(3)=Z$(S+O2,B1):E$(4)
=Z$(S+O2,B2):E$(5)=Z$(S+O3,B1):E$(6)=Z$(S+O3,B2):E$(7)=Z$(S+O4,B
1):E$(8)=Z$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X
))-2):GOTO 1090
1088      T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090      C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$
:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$
(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 OPEN"I",1,"EAGLE/DAT":INPUT#1,DATE$
3010 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
3020 CLOSE:RETURN
3050 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"MARBLE"
```

*Program Listing 4*

```
3 CLEAR 3200:CLS:PRINT@465,"WORKING FILENAME....";CHR$(34);"MARB
LE";CHR$(34)
4 ON ERROR GOTO 3050
5 DEFSTR V,H,M,D:M="*":V=":":H=".":DEFINTU,B,O,X:DIMZ$(20,20):GO
SUB3000
8 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:LPRINTSTRING$(80,H)
```

*Listing 4 continues*

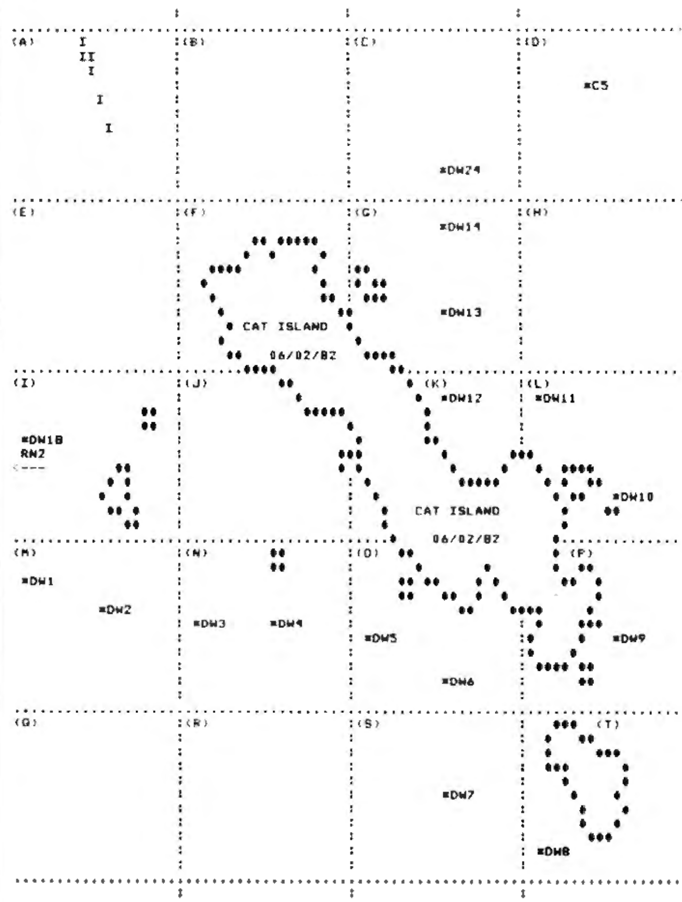



*Figure 2*

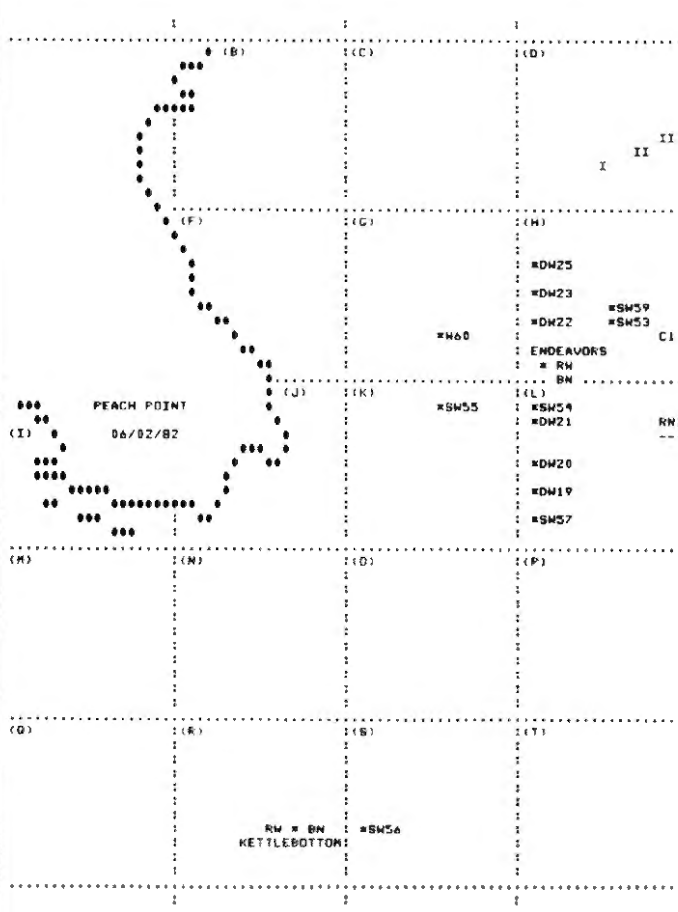



*Figure 3*

*Listing 4 continued*

```
10 LPRINT"(A)";TAB(15)"* C1";V;"(B)";TAB(39)V;"(C)";TAB(59)V;"(D
)":GOSUB2000:S=1:L=1:U2=0:GOSUB1000
20 LPRINT" "D1;TAB(12)"II";TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "
D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1
000
30 LPRINT" "D1;TAB(11)"I I I";TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;
" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOS
UB1000
40 LPRINT" "D1;TAB(10)"I  I  I";TAB(19)V;" "D3;TAB(30)D4;TAB(39)
V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:G
OSUB1000
50 LPRINT" "D1;TAB(14)"I";TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D
5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB10
00
60 LPRINT" "D1;TAB(15)"I";TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D
5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB10
00
70 LPRINT" "D1;TAB(16)"I";TAB(19);V;" "D3;TAB(30)D4;TAB(39)V;" "
D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1
000
80 LPRINT" "D1;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6
;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
90 LPRINT" "D1;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6
;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
100 LPRINT" "D1;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D
6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
110 LPRINT" "D1;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D
6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
120 LPRINT STRING$(80,H):U2=1:S=2:L=1
130 LPRINT"(E)";TAB(19)V;"(F)";TAB(39)V;"(G)";TAB(59)V;"(H)";D(4
):GOSUB1000
140 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5
;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB100
0
150 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5
;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB100
0
160 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(37)M;M;M;"
 "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U4=0
:U5=0:GOSUB1000
170 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(36)M;TAB(40)M;TAB(46
)M;M;M;M;" "D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSU
B1000
180 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(35)M;TAB(41)M;M;M;M;
M;TAB(50)M;" "D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GO
SUB1000
190 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(34)M;TAB(51)M;M;M;TA
B(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
200 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(32)M;M;TAB(54)M;TAB(
59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
210 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(31)M;TAB(55)M;TAB(59
)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
220 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)M;TA"(56)M;TAB(59
)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
230 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)M;TAB(36)"MARBLEH
EAD ROCK";TAB(57)M;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:
GOSUB1000
240 LPRINTSTRING$(30,H);M;TAB(50)M;" ";STRING$(20,H):S=3:L=1
250 LPRINT"(I)";TAB(19)V;"(J)";TAB(31)M;TAB(39)DATE$;TAB(59)M;TA
B(61)"(L)":GOSUB2000:U4=0:U5=0:U6=0:GOSUB1000
260 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(32)M;TAB(59)M;TAB(63
)" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:U4=0:U5=0:U6=0:GO
SUB1000
270 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(33)M;TAB(60)M;TAB(63
)" ";D7;STRING$(70-PEEK(16539),32);D8:U4=1:GOSUB1000
280 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(29)D4;TAB(34)M;TAB(6
1)M;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
290 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(29)D4;TAB(34)M;TAB(6
2)M;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
300 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(29)D4;TAB(34)M;TAB(6
2)M;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:U3=0:U4=0:GOSUB10
00
310 LPRINT" "D1;TAB(9)D2;TAB(17)M;M;M;TAB(35)M;TAB(62)M;" "D7;ST
RING$(70-PEEK(16539),32);D8:GOSUB2000:U3=0:U4=0:U5=0:U6=0:GOSUB1
000
320 LPRINT" "D1;TAB(9)D2;TAB(16)M;TAB(20)M;M;M;M;TAB(29)M;M;M;M;
TAB(35)M;TAB(61)M;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:GOS
UB1000
330 LPRINT" "D1;TAB(9)D2;TAB(17)M;TAB(24)M;M;M;M;M;TAB(33)M;M;TA
B(56)M;M;M;TAB(61)M;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:G
OSUB1000
340 LPRINT" "D1;TAB(9)D2;TAB(17)M;TAB(41)M;M;TAB(55)M;TAB(59)M;M
;TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
350 LPRINT" "D1;TAB(9)D2;TAB(16)M;TAB(40)M;TAB(43)M;M;TAB(53)M;M
;TAB(59)V,TAB(63)" "D7;STRING$(70-PEEK(16539),32);D8
360 LPRINTSTRING$(16,H);M;TAB(40)M;H;H;H;H;M;TAB(51)M;M;STRING$(
27,H):S=4:L=1
370 LPRINT"(M)";TAB(17)M;TAB(40)M;"(O)";TAB(46)M;TAB(50)M;TAB(59
)V;"(P)":GOSUB1000
380 LPRINT" "D1;TAB(9)D2;TAB(18)M;TAB(38)M;M;TAB(47)M;M;M;TAB(59
)V;" "D7;STRING$(70-PEEK(16539),32);D8:U6=1:GOSUB1000
390 LPRINT" "D1;TAB(9)D2;TAB(19)M;M;TAB(37)M;TAB(39)V;" ";M;M;TA
B(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
400 LPRINT" "D1;TAB(9)D2;TAB(19)V;TAB(21)M;M;TAB(36)M;TAB(39)V;M
;TAB(43)M;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8
:GOSUB1000
410 LPRINT" "D1;TAB(9)D2;TAB(19)V;TAB(23)M;M;TAB(35)M;TAB(39)V;"
 ";M;M;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GO
SUB1000
420 LPRINT" "D1;TAB(9)D2;TAB(19)V;"(N)";TAB(25)M;M;M;TAB(34)M;TA
B(39)V;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U3
=1:GOSUB1000
430 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(28)M;TAB(33)M;TAB(39
)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:
GOSUB1000
440 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(29)M;M;M;M;TAB(39)V;
" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOS
UB2000:U4=0:GOSUB1000
450 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(39)V;" "D5;TAB(50)D6
;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
460 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5
;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB100
0
470 LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5
;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8
480 LPRINTSTRING$(80,H):S=5:L=1
```

*Listing 4 continues*

*Listing 4 continued*

```
490 LPRINT"(Q)";TAB(19)V;"(R)";TAB(39)V;"(S)";TAB(59)V;"(T)":GOS
UB1000
500 FORC=1TO10:LPRINT" "D1;TAB(9)D2;TAB(19)V;" "D3;TAB(30)D4;TAB
(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);
D8:GOSUB1000:NEXTC
510 LPRINTSTRING$(80,H)
520 LPRINTTAB(19)V;TAB(39)V;TAB(59)V
530 GOTO 4000
1000 REM ASSIGN D1>D8 HERE
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3
:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8
:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=Z$(S+O1,B1):E$(2)=Z$(S+O1,B2):E$(3)=Z$(S+O2,B1):E$(4)
=Z$(S+O2,B2):E$(5)=Z$(S+O3,B1):E$(6)=Z$(S+O3,B2):E$(7)=Z$(S+O4,B
1):E$(8)=Z$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X
))-2):GOTO 1090
1088       T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090       C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$
:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$
(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 OPEN"I",1,"MARBLE/DAT":INPUT#1,DATE$
3010 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
3020 CLOSE:RETURN
3050 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"MISERY"
```

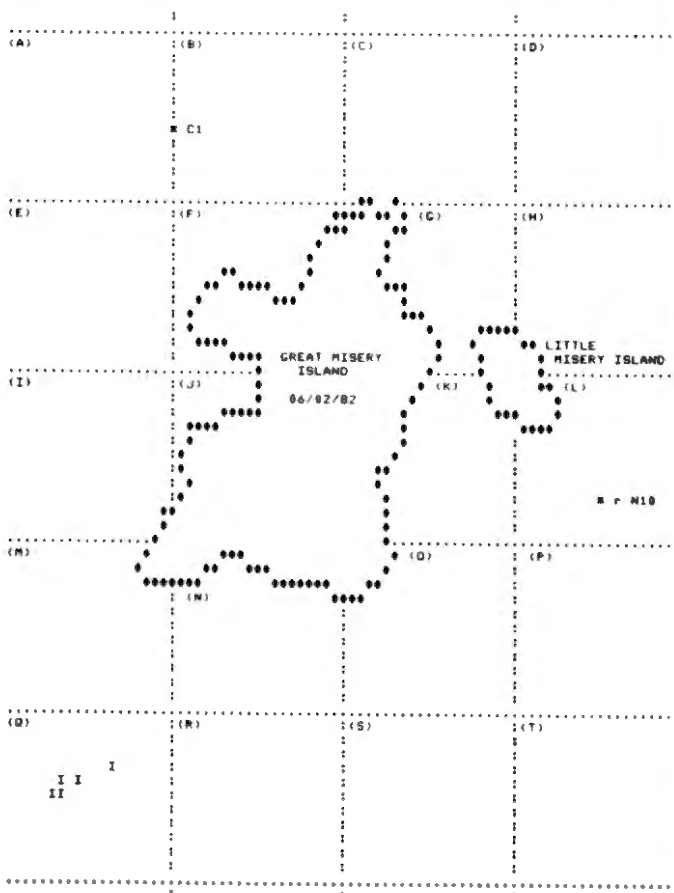



*Figure 4*

*Program Listing 5*

```
3 CLEAR3200:ON ERROR GOTO 3550:DEFSTRV,H,M,D,A:DEFINTU,B,O,X:DIM
Z$(20,20):M="$":A="I":V=":":H=".":CLS:PRINT@465,"WORKING FILENAM
E....";CHR$(34);"MISERY";CHR$(34):GOSUB3500:GOTO8
5 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;TAB(21)D3;TAB(30)D4;TAB(39)V
;TAB(41)D5;TAB(50)D6;TAB(59)V;TAB(61)D7;STRING$(70-PEEK(16539),3
2);D8:RETURN
7 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:RETURN
8 GOSUB7:LPRINTSTRING$(80,H):GOSUB2000:S=1:L=1
10 LPRINT"(A)",TAB(19)V;"(B)";TAB(39)V;"(C)";TAB(59)V;"(D)":GOSU
B1000
```

*Listing 5 continues*



See List of Advertisers on Page 451

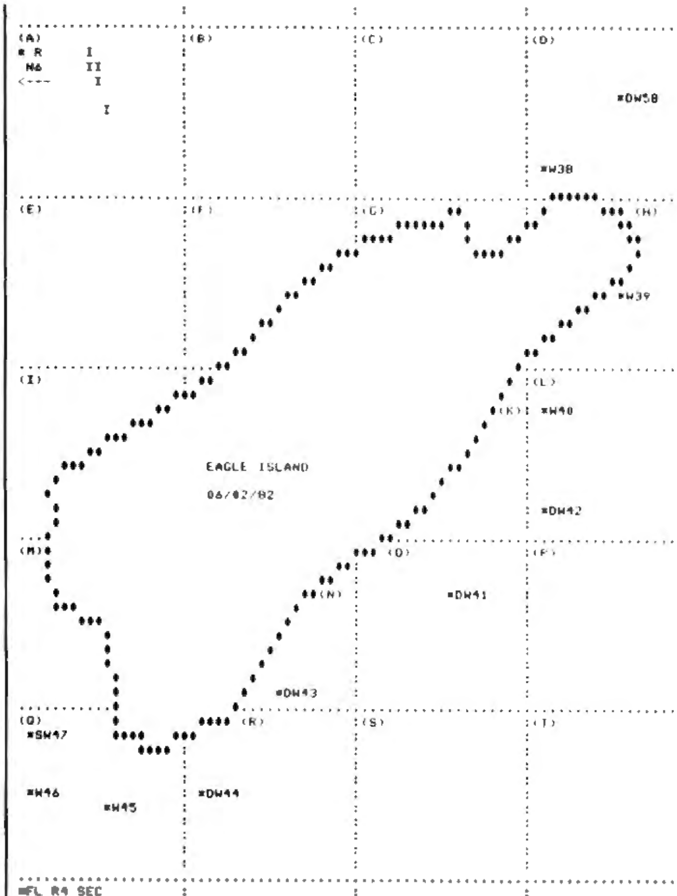



*Figure 5*

*Listing 5 continued*

```
15 GOSUB5:GOSUB1000
20 GOSUB5:GOSUB1000
30 GOSUB5:GOSUB1000
40 GOSUB5:GOSUB1000
50 GOSUB5:U3=0:GOSUB1000
60 LPRINT" "D1;TAB(10)D2;TAB(19)"* C1";TAB(30)D4;TAB(39)V;" "D5;
TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U3=1:GOSU
B1000
70 GOSUB5:GOSUB1000
80 GOSUB5:GOSUB1000
90 GOSUB5:GOSUB1000
100 GOSUB5
110 LPRINTSTRING$(41,H);M;M;"  ";M;STRING$(34,H):U5=0:S=2:L=1
120 LPRINT"(E)";TAB(19)V;"(F)";TAB(38)M;M;M;M;" ";M;M;" ";M;" (G
)";TAB(59)V;"(H)":GOSUB1000
130 GOSUB3000:LPRINT" "D3;TAB(30)D4;TAB(37)M;M;M;TAB(45)M;M;TAB(
50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U4=0:U5=0:GOS
UB1000
140 GOSUB3000:LPRINT" "D3;TAB(36)M;TAB(44)M;TAB(50)D6;TAB(59)V;"
 "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
150 GOSUB3000:LPRINT" "D3;TAB(35)M;TAB(44)M;TAB(50)D6;TAB(59)V;"
 "D7;STRING$(70-PEEK(16539),32);D8:U3=0:GOSUB1000
160 GOSUB3000:LPRINTTAB(25)M;M;TAB(35)M;TAB(43)M;TAB(50)D6;TAB(5
9)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
170 GOSUB3000:LPRINTTAB(23)M;M;"  ";M;M;M;M;"   ";M;TAB(44)M;M;M
;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB100
0
180 GOSUB3000:LPRINTTAB(22)M;TAB(31)M;M;M;TAB(46)M;TAB(50)D6;TAB
(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:U6=0:GOSUB1000
190 GOSUB3000:LPRINTTAB(21)M;TAB(46)M;M;M;TAB(59)V;" "D7;STRING$
(70-PEEK(16539),32);D8:U7=0:GOSUB1000
200 GOSUB3000:LPRINTTAB(21)M;TAB(49)M;TAB(55)M;M;M;M;M;STRING$(7
0-PEEK(16539),32);D8:U8=0:GOSUB1000
210 GOSUB3000:LPRINTTAB(22);M;M;M;M;TAB(50)M;"   ";M;TAB(60)M;M;
" LITTLE":GOSUB1000
220 GOSUB3000:LPRINTTAB(26)M;M;M;M;"  GREAT MISERY";TAB(50)M;TAB
(55)M;TAB(62)M;" MISERY ISLAND"
230 LPRINTSTRING$(29,H);M;"    ISLAND";TAB(49);M;H;H;H;H;H;M;TAB
(62)M;STRING$(17,H):U8=1:S=3:L=1
240 LPRINT"(I)";TAB(19)V;"(J)";TAB(29)M;TAB(48)M;" (K)";TAB(56)M
;TAB(62)M;M;" (L)":GOSUB1000
250 GOSUB3000:LPRINTTAB(29)M;TAB(33)DATE$;TAB(47)M;TAB(56)M;TAB(
63)" ";M;STRING$(70-PEEK(16539),32);D8:GOSUB1000
260 GOSUB3000:LPRINTTAB(25)M;M;M;M;M;TAB(46)M;TAB(57)M;M;M;TAB(6
3)" ";M;STRING$(70-PEEK(16539),32);D8:U6=1:GOSUB1000
270 GOSUB3000:LPRINTTAB(21)M;M;M;M;TAB(46)M;TAB(50)D6;TAB(60)M;M
;M;M;STRING$(70-PEEK(16539),32);D8:GOSUB1000
280 GOSUB3000:LPRINTTAB(21)M;TAB(46)M;TAB(50)D6;TAB(59)V;STRING$
(70-PEEK(16539),32);D8:U7=1:GOSUB1000
290 GOSUB3000:LPRINTM;TAB(45)M;TAB(50)D6;TAB(59)V;" "D7;STRING$(
70-PEEK(16539),32);D8:GOSUB1000
300 GOSUB3000:LPRINTM;TAB(43)M;M;TAB(50)D6;TAB(59)V;" "D7;STRING
$(70-PEEK(16539),32);D8:GOSUB1000
310 GOSUB3000:LPRINT" "M;TAB(43)M;TAB(50)D6;TAB(59)V;" "D7;STRIN
```

*Listing 5 continues*

*Listing 5 continued*

```
G$(70-PEEK(16539),32);D8:U8=0:GOSUB1000
320 GOSUB3000:LPRINTM;TAB(44)M;TAB(50)D6;TAB(59)V;" "D7;STRING$(
69-PEEK(16539),32);"* r N10":GOSUB1000
330 LPRINT" "D1;TAB(10)D2;TAB(18)M;M;TAB(44)M;TAB(50)D6;TAB(59)V
;" ";D7;U8=1;GOSUB1000
340 LPRINT" "D1;TAB(10)D2;TAB(18)M;TAB(44)M;TAB(50)D6;TAB(59)V;"
 ";D7;STRING$(70-PEEK(16539),32);D8
350 LPRINTSTRING$(17,H);M;TAB(44);M;STRING$(35,H):S=4:L=1
360 LPRINT"(M)";TAB(16)M;TAB(25)M;M;M;TAB(45)M;" (O)";TAB(59)V;"
 (P)":GOSUB1000
370 LPRINT" "D1;TAB(15)M;TAB(23)M;M;"    ";M;M;M;TAB(44)M;TAB(50)
D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8;GOSUB1000
380 LPRINT" "D1;TAB(16)M;M;M;M;M;M;M;TAB(31)M;M;M;M;M;M;M;"     "
;M;M;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSU
B1000
390 GOSUB3000:LPRINT" (N)";TAB(38)M;M;M;M;TAB(50)D6;TAB(59)V;" "
D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:GOSUB1000
400 GOSUB5:GOSUB1000
410 GOSUB5:GOSUB1000
420 GOSUB5:GOSUB1000
430 GOSUB5:GOSUB1000
440 GOSUB5:GOSUB1000
450 GOSUB5:GOSUB1000
460 GOSUB5:GOSUB1000
470 LPRINTSTRING$(80,H):S=5:L=1
480 LPRINT"(Q)";TAB(19)V;"(R)";TAB(39)V;"(S)";TAB(59)V;"(T)":GOS
UB1000
490 GOSUB5:GOSUB1000
500 GOSUB5:U1=0:U2=0:GOSUB1000
510 LPRINTTAB(12)A;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(5
0)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
520 LPRINTTAB(6)A;" "A;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;T
AB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB1000
530 LPRINTTAB(5)A;A;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "D5;TAB(
50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:GOSUB2000:GOS
UB1000
540 GOSUB5:GOSUB1000
550 GOSUB5:GOSUB1000
560 GOSUB5:GOSUB1000
570 GOSUB5:GOSUB1000
580 GOSUB5:GOSUB1000
590 LPRINTSTRING$(80,H):GOSUB7:GOTO4000
1000 REM ASSIGN D1>D8 HERE
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3
:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8
:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=X$(S+O1,B1):E$(2)=X$(S+O1,B2):E$(3)=X$(S+O2,B1):E$(4)
=X$(S+O2,B2):E$(5)=X$(S+O3,B1):E$(6)=X$(S+O3,B2):E$(7)=X$(S+O4,B
1):E$(8)=X$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X
))-2):GOTO 1090
1088      T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090      C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$
:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$
(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;:RETURN
3500 OPEN"I",1,"MISERY/DAT":INPUT#1,DATE$
3510 FORX=1TO20:FORX1=1TO20:INPUT#1,X$(X,X1):NEXTX1:NEXTX
3520 CLOSE:RETURN
3550 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"PEACH"
```

*Program Listing 6*

```
1 CLEAR3200:ON ERROR GOTO 3050:DEFSTRV,H,M,D,A:DEFINTU,B,O,X:DIM
X$(20,20):M="@":A="I":V=":":H=".":CLS:PRINT@465,"WORKING FILENAM
E....";CHR$(34);"PEACH";CHR$(34):GOSUB3000:GOTO8
2 LPRINTTAB(19)V;TAB(23)D3;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TA
B(59)V;:RETURN
3 LPRINTTAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING
$(70-PEEK(16539),32);D8:RETURN
4 LPRINTTAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(
16539),32);D8:RETURN
5 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)D4;TAB(39)V;" "
D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:RETURN
7 LPRINTTAB(19)V;TAB(39)V;TAB(59)V:RETURN
8 GOSUB7:LPRINTSTRING$(80,H):GOSUB2000
10 LPRINTTAB(23)M;" (B)";TAB(39);V;"(C)";TAB(59)V;"(D)":U1=0:U2=
0:S=1:L=1:GOSUB1000
20 LPRINTTAB(20)M;M;M;" "D3;:GOSUB3:GOSUB1000
30 LPRINTTAB(19)M;TAB(23)D3;:GOSUB3:GOSUB1000
40 LPRINTTAB(20)M;M;TAB(23)D3;:GOSUB3:GOSUB1000
50 LPRINTTAB(17)M;M;M;M;M;TAB(23)D3;:GOSUB3:GOSUB1000
60 LPRINTTAB(16)M;TAB(19)V;TAB(23)D3;:GOSUB3:U7=0:U8=0:GOSUB1000
```

*Listing 6 continues*

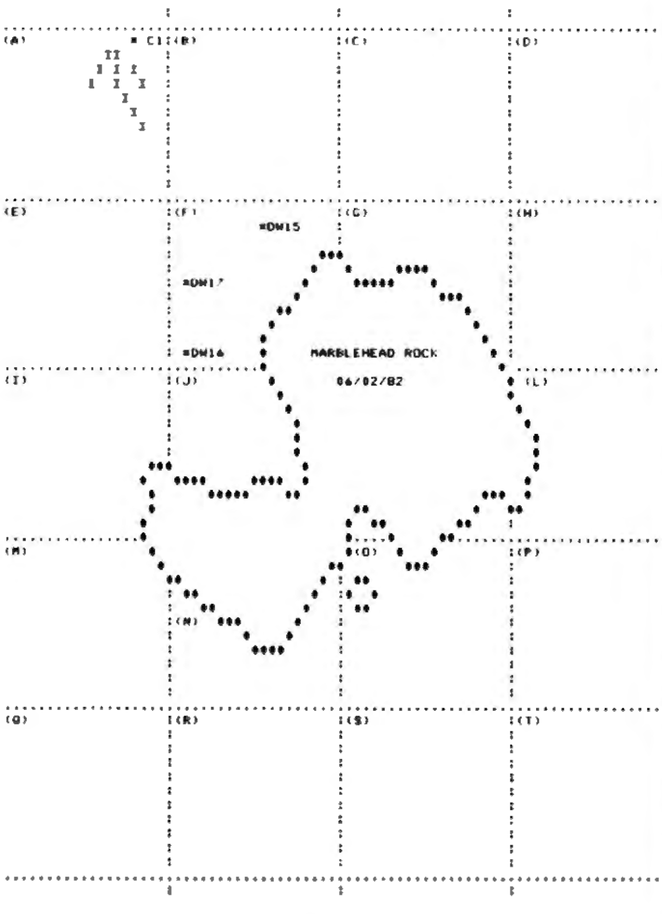



*Figure 6*



See List of Advertisers on Page 451

*Listing 6 continued*

```
70 LPRINTTAB(15)M;:GOSUB2:LPRINTSTRING$(76-PEEK(16539),32);A;A:G
OSUB1000
80 LPRINTTAB(15)M;:GOSUB2:LPRINTSTRING$(73-PEEK(16539),32);A;A:G
OSUB1000
90 LPRINTTAB(15)M;:GOSUB2:LPRINTSTRING$(69-PEEK(16539),32);A:U9=
1:U8=1:GOSUB1000
100 LPRINTTAB(15)M;TAB(19);V;TAB(23)D3;:GOSUB3:GOSUB1000
110 LPRINTTAB(16)M;TAB(19)V;TAB(23)D3;:GOSUB3
120 LPRINTTAB(17)M;" ";STRING$(60,H):S=2:L=1
130 LPRINTTAB(18)M;" ";"(F)";TAB(39)V;"(G)";TAB(59)V;"(H)":GOSUB
2000:U1=0:U2=0:GOSUB1000
140 LPRINTTAB(19)M;TAB(23)D3;:GOSUB3:GOSUB1000
150 LPRINTTAB(20)M;TAB(23)D3;:GOSUB3:GOSUB1000
160 LPRINTTAB(21)M;TAB(23)D3;:GOSUB3:GOSUB1000
170 LPRINTTAB(21)M;TAB(23)D3;:GOSUB3:GOSUB1000
180 LPRINTTAB(21)M;TAB(23)D3;:GOSUB3:U3=0:GOSUB1000
190 LPRINTTAB(22)M;M;:GOSUB3:GOSUB1000
200 LPRINTTAB(24)M;M;:GOSUB3:U8=0:GOSUB1000
210 LPRINTTAB(26)M;TAB(30)D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;"
 "D7;STRING$(76-PEEK(16539),32);"C1 *":U7=0:GOSUB1000
220 LPRINTTAB(27)M;M;" "D4;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;" E
NDEAVORS":U4=0:GOSUB1000
230 LPRINTTAB(29)M;M;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;TAB(62)"*
 RW":GOSUB1000
240 LPRINTTAB(30)M;STRING$(32,H);" BN ";STRING$(13,H):S=3:L=1
250 LPRINTTAB(30)M;" (J)";TAB(39)V;"(K)";TAB(59)V;"(L)":GOSUB200
0:U1=0:U2=0:U3=0:U4=0:GOSUB1000
260 LPRINTTAB(1)M;M;M;TAB(10)"PEACH POINT";TAB(30)M;:GOSUB4:U8=0
:GOSUB1000
270 LPRINTTAB(3)M;M;TAB(31)M;TAB(39)V;" "D5;TAB(50)D6;TAB(59)V;"
 "D7;STRING$(76-PEEK(16539),32);"RN2*":GOSUB1000
280 LPRINT"(I)";TAB(5)M;TAB(12)DATE$;TAB(32)M;TAB(39)V;" "D5;TAB
(50)D6;TAB(59)V;" "D7;STRING$(76-PEEK(16539),32);"--->":U8=1:GOS
UB1000
290 LPRINTTAB(6)M;TAB(27)M;M;M;"  ";M;:GOSUB4:GOSUB1000
300 LPRINTTAB(3)M;M;M;TAB(26)M;TAB(30)M;M;:GOSUB4:U4=1:GOSUB1000
310 LPRINTTAB(3)M;M;M;M;TAB(25)M;:GOSUB3:GOSUB1000
320 LPRINTTAB(7)M;M;M;M;M;TAB(25)M;:GOSUB3:GOSUB1000
330 LPRINTTAB(4)M;M;TAB(12);STRING$(10,M);"  ";M;:GOSUB3:GOSUB10
00
340 LPRINTTAB(8)M;M;M;TAB(19)V;TAB(22)M;M;:GOSUB3:GOSUB1000
350 LPRINTTAB(12)M;M;M;TAB(19)V;:GOSUB3
360 LPRINTSTRING$(80,H):GOSUB2000:S=4:L=1
370 LPRINT"(M)";TAB(19)V;"(N)";TAB(39)V;"(O)";TAB(59)V;"(P)":GOS
UB1000
380 FORC=1TO10:GOSUB5:GOSUB1000:NEXTC
400 LPRINTSTRING$(80,H):S=5:L=1
490 LPRINT"(Q)";TAB(19)V;"(R)";TAB(39)V;"(S)";TAB(59)V;"(T)":GOS
UB1000
500 FORC=1TO5:GOSUB5:GOSUB1000:NEXTC:GOSUB5:U4=0:GOSUB1000
560 LPRINT" "D1;TAB(10)D2;TAB(19)V;" "D3;TAB(30)"RW * BN";TAB(39
)V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8:
GOSUB1000
570 LPRINTTAB(1)D1;TAB(10)D2;TAB(19)V;" "D3;TAB(27)"KETTLEBOTTOM
";V;" "D5;TAB(50)D6;TAB(59)V;" "D7;STRING$(70-PEEK(16539),32);D8
:GOSUB2000:GOSUB1000
580 GOSUB5:GOSUB1000
590 GOSUB5:LPRINTSTRING$(80,H):GOSUB7:GOTO4000
1000 REM ASSIGN D1>D8 HERE
1010 IFL=1THENB1=1:B2=20ELSEIFL=2THENB1=2:B2=19ELSEIFL=3THENB1=3
:B2=18ELSEIFL=4THENB1=4:B2=17ELSEIFL=5THENB1=5:B2=16
1020 IFL=6THENB1=6:B2=15ELSEIFL=7THENB1=7:B2=14ELSEIFL=8THENB1=8
:B2=13ELSEIFL=9THENB1=9:B2=12ELSEIFL=10THENB1=10:B2=11
1030 IFS=1THENO1=0:O2=1:O3=2:O4=3:GOTO1080
1040 IFS=2THENO1=3:O2=4:O3=5:O4=6:GOTO1080
1050 IFS=3THENO1=6:O2=7:O3=8:O4=9:GOTO1080
1060 IFS=4THENO1=9:O2=10:O3=11:O4=12:GOTO1080
1070 O1=12:O2=13:O3=14:O4=15
1080 E$(1)=Z$(S+O1,B1):E$(2)=Z$(S+O1,B2):E$(3)=Z$(S+O2,B1):E$(4)
=Z$(S+O2,B2):E$(5)=Z$(S+O3,B1):E$(6)=Z$(S+O3,B2):E$(7)=Z$(S+O4,B
1):E$(8)=Z$(S+O4,B2)
1082 FOR X=1 TO 8:IFE$(X)=""THEN1091ELSET$=RIGHT$(E$(X),LEN(E$(X
))-2):GOTO 1090
1088        T$=RIGHT$(E$(X),LEN(E$(X))-3)
1090        C$=LEFT$(T$,1):IFC$="S"ORC$="D"ORC$="W"THENE$(X)="*"+T$
:GOTO1091ELSEGOTO1088
1091 NEXT X
1095 D1=E$(1):D2=E$(2):D3=E$(3):D4=E$(4):D5=E$(5):D6=E$(6):D7=E$
(7):D8=E$(8):L=L+1:RETURN
2000 U1=1:U2=1:U3=1:U4=1:U5=1:U6=1:U7=1:U8=1:RETURN
3000 OPEN"I",1,"PEACH/DAT":INPUT#1,DATE$
3010 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
3020 CLOSE:RETURN
3050 CLOSE:END
4000 LPRINTSTRING$(3,138):RUN"DAT/COL"
```

*Program Listing 7*

```
100 REM .....
110 ON ERROR GOTO 370
120 CLEAR 5000:DIMZ$(20,20),D$(20,20):CLS:U=1:V=448:GOSUB390
130 GOSUB470:PRINT@0,"":INPUT"ENTER MAP NAME";MAP$:IF MAP$="CAT"
ORMAP$="EAGLE"ORMAP$="BAKERS"ORMAP$="MARBLE"ORMAP$="MISERY"ORMAP
$="PEACH"THEN140ELSEPRINT@0,STRING$(63," "):GOTO130
140 GOSUB470:PRINT@192,"READ DATA DISK ? (Y OR N)":GOSUB500:IFS$
="Y"THENGOSUB430
150 GOSUB470:PRINT@192,"";:INPUT"ENTER DATE OR (ENTER) MM/DD/YY"
;A$:IFA$<>""THENDATE$=A$:GOSUB310ELSEGOSUB310
160 GOSUB470:PRINT@192,"ERASE KEEPERS ?? (Y OR N)":GOSUB500:IFS$
="Y"THEN460ELSEIFS$="N"THEN170ELSE160
170 CLS:PRINT@0,"DATA INPUT ROUTINE FOR MAP DESIG. ";CHR$(34);MA
P$
CHR$(34);TAB(53);DATE$:PRINTSTRING$(63,"=")
180 GOSUB470:PRINT@192,"ENTER SECTOR OR COMMAND LETTER":GOSUB500
:IF S$="" THEN180ELSEIFASC(S$)=>65ANDASC(S$)<=84 AND LEN(S$)=1 T
HEN190 ELSEIFS$="W"THEN330ELSEPRINT@192,STRING$(63," "):GOTO180
190 U=1:FORX=1TO20:PO=X:IFZ$(ASC(S$)-64,X)=""THEN200ELSEGOSUB400
```

*Listing 7 continues*

*Listing 7 continued*

```
:IFU=0THENU=1ELSEU=0
200 NEXT X:PO=0
210 PRINT@192,STRING$(63," "):PRINT@192,"WORKING ";CHR$(34);S$;C
HR$(34);" SECTOR..............."
220 GOSUB470:PRINT@320,"";:INPUT"POSIT #, POT CODE, SHORTS/KEEPE
RS (PO,CO,SH/KE)";PO,PC$,K$
230 IF PO>0ANDPO<=20THEN240ELSEIFPO>20ORPO=0ANDPC$<>""THENGOSUB3
20:GOTO220:ELSEIFPO=0ANDPC$=""THENGOSUB320:PRINT@192,STRING$(63,
" "):U=1:V=448:GOSUB420:GOTO180
240 IF PC$=""THENGOSUB320:GOTO220
250 IFPC$="NA"THENZ$(ASC(S$)-64,PO)="":D$(ASC(S$)-64,PO)="":PO=0
:PC$="":U=1:V=448:GOSUB320:GOSUB420:GOTO190
260 I$=LEFT$(PC$,2):IFI$="DW"ORI$="SW"THEN280ELSE270
270 I$=LEFT$(PC$,1):IF I$="W"THEN280ELSEPC$="":GOSUB320:GOTO220
280 GOSUB380:GOSUB400
290 IFU=0THENU=1ELSEU=0
300 PO=0:PC$="":GOSUB320:GOTO220
310 PRINT@192,STRING$(63," "):RETURN
320 PRINT@320,STRING$(64," ");:RETURN
330 MAP$=MAP$+"/DAT":OPEN"O",2,MAP$:PRINT#2,DATE$
340 FORX=1TO20:FORX1=1TO20:PRINT#2,Z$(X,X1):NEXTX1:NEXTX
350 FORX=1TO20:FORX1=1TO20:PRINT#2,D$(X,X1):NEXTX1:NEXTX
360 CLOSE:END
370 CLOSE:RESTORE:OUT254,1:V=448:GOSUB420:GOTO120
380 Z$(ASC(S$)-64,PO)=S$+RIGHT$(STR$(PO),LEN(STR$(PO))-1)+PC$:D$
(ASC(S$)-64,PO)=K$:RETURN
390 OUT254,0:FORX=1TO20:FORX1=1TO20:Z$(X,X1)="":NEXTX1:NEXTX:RES
TORE:OUT254,1:RETURN
400 IFU=1THEN PRINT@V,Z$(ASC(S$)-64,PO);TAB(9);"S/K=";D$(ASC(S$)
-64,PO):RETURN
410 PRINT@V+32,Z$(ASC(S$)-64,PO);TAB(41);"S/K=";D$(ASC(S$)-64,PO
):V=V+64:RETURN
420 PRINT@V,STRING$(63," ");:IFV=960THENV=448:RETURNELSEV=V+64:G
OTO420
430 OPEN"I",1,MAP$+"/DAT":INPUT#1,DATE$
440 FORX=1TO20:FORX1=1TO20:INPUT#1,Z$(X,X1):NEXTX1:NEXTX
450 FORX=1TO20:FORX1=1TO20:INPUT#1,D$(X,X1):NEXTX1:NEXTX:CLOSE:R
ETURN
460 FORX=1TO20:FORX1=1TO20:D$(X,X1)="":NEXTX1:NEXTX:GOTO170
470 OUT 255,1:FOR X=1TO100:NEXTX:OUT 255,0:RETURN
500 S$=INKEY$:IFS$=""THEN500ELSERETURN
```

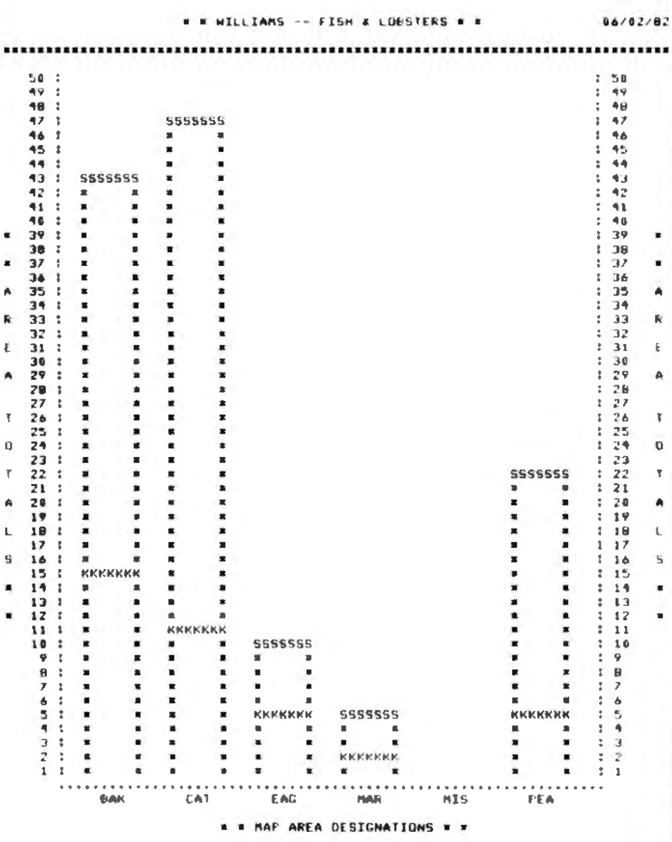



*Figure 7*

*Program Listing 8*

```
100 ON ERROR GOTO 450: REM FILENAME "DAT/COL"
110 CLEAR 5000:DEFINTX,K,C:DIMP$(6,50),Z$(20,20),D$(20,20),E$(10
0)
120 DATA BAKERS/DAT,CAT/DAT,EAGLE/DAT,MARBLE/DAT,MISERY/DAT,PEAC
H/DAT
130 DATABA,CA,EA,MA,MI,PE
140 REM INPUT THE SIX FILES AND ASSIGN TO P$ ARRAY
```

*Listing 8 continues*

```
                  * * WILLIAMS --  FISH & LOBSTERS * *               06/02/82
*****************************************************************************
MAP DESIG............BAKERS
  L7DW36*2/1   L9DW35*9/1   L12W28*0/0   L18DW37*4/1  M5SW40*0/0   M14SW50*0/0
  M15SW49*2/2  N15DW30*7/0  N19DW29*2/2  O6DW31*1/1   O14W51*1/1   O17DW32*3/2
  O19SW33*5/0  P1DW34*2/3   P6W52*0/0    P17DW26*5/1  P20W27*0/0


BAKERS TTL'S..      POTS= 17   SHORTS= 43  KEEPERS= 15   RATIO KPRS/POTS.. 0.9
*****************************************************************************
MAP DESIG............CAT
  C12DW24*5/0  G14DW13*1/1  G20DW14*4/1  I4DW10*2/0   K20DW12*4/3  L1DW11*3/0
  L13DW10*3/0  M2DW1*1/1    M17DW2*0/0   N5DW3*6/0    N16DW4*1/0   O6DW5*4/1
  O12DW6*2/0   P15DW9*7/0   S16DW7*3/1   T9DW8*1/3


CAT TTL'S..         POTS= 16   SHORTS= 47  KEEPERS= 11   RATIO KPRS/POTS.. 0.7
*****************************************************************************
MAP DESIG............EAGLE
  D9W38*0/0    D17DW58*0/0  H15W39*0/1   L2W40*0/0    L9DW42*3/2   N11DW43*0/0
  O18DW41*4/1  Q18W47*0/0   Q5W46*0/0    Q15W45*2/0   R5DW44*1/1


EAGLE TTL'S..       POTS= 11   SHORTS= 10  KEEPERS= 5    RATIO KPRS/POTS.. 0.5
*****************************************************************************
MAP DESIG............MARBLEHEAD
  F5DW17*1/1   F10DW16*3/1  F20DW15*1/0


MARBLEHEAD TTL'S..  POTS= 3    SHORTS= 5   KEEPERS= 2    RATIO KPRS/POTS.. 0.7
*****************************************************************************
MAP DESIG............MISERY
NO DATA AVAILABLE FOR MISERY


MISERY TTL'S..      POTS= 0    SHORTS= 0   KEEPERS= 0    RATIO KPRS/POTS.. 0
*****************************************************************************
MAP DESIG............PEACH
  C13W60*0/0   H3DW25*7/1   H5DW23*3/0   H7DW22*2/0   H14SW53*1/0  H15SW59*1/1
  K20SW55*0/0  L1SW54*0/0   L2DW21*3/2   L5DW20*0/0   L7DW19*4/1   L9SW57*0/0
  S7SW56*1/0


PEACH TTL'S..       POTS= 13   SHORTS= 22  KEEPERS= 5    RATIO KPRS/POTS.. 0.4
*****************************************************************************
TOTALS FOR PERIOD 06/02/82.......ALL AREAS.............

TOTAL POTS WORKED 60        TOTAL SHORTS 127             TOTAL KEEPERS 30

% KEEPERS VS. TOTAL 23      SHORTS/POT... 2.12           KEEPERS/POT... 0.63

1982 TO DATE TOTALS........ YTD SHORTS 1002              YTD KEEPERS 262
```

*Figure 8*

*Listing 8 continued*

```
150 FORX=1TO6:READT$:CLS:PRINT@448,CHR$(23);"WORKING FILE-NAME "
;T$:GOSUB480:C=1
160 OPEN"I",1,T$
170 INPUT#1,DATE$:IFX=1THENH$=DATE$
180 IF DATE$<>H$THENGOTO450
190 FORX1=1TO20:FORX2=1TO20:INPUT#1,Z$(X1,X2):NEXTX2:NEXTX1
200 FOR X1=1TO20:FORX2=1TO20:INPUT#1,D$(X1,X2):NEXTX2:NEXTX1
210 CLOSE
220 FORX1=1TO20:FORX2=1TO20:IFZ$(X1,X2)=""THEN230ELSE260
230 NEXT X2
240 NEXT X1
250 NEXTX:GOTO270
260 P$(X,C)=Z$(X1,X2)+"*"+D$(X1,X2):C=C+1:GOTO230
270 REM ASSIGN FILENAME FROM DATE$
280 A$=LEFT$(DATE$,2):C$=MID$(DATE$,4,2)+":1":IFA$="04"THENB$="A
PR/D"+C$:GOTO370
290 IFA$="05"THENB$="MAY/D"+C$:GOTO370
300 IFA$="06"THENB$="JUN/D"+C$:GOTO370
310 IFA$="07"THENB$="JUL/D"+C$:GOTO370
320 IFA$="08"THENB$="AUG/D"+C$:GOTO370
330 IFA$="09"THENB$="SEP/D"+C$:GOTO370
340 IFA$="10"THENB$="OCT/D"+C$:GOTO370
350 IFA$="11"THENB$="NOV/D"+C$:GOTO370
360 GOTO 450
370 REM OUTPUT TO DISK
380 CLS:PRINT@460,"PACKING DATA TO DRIVE #1..";B$
390 OPEN"O",2,B$:PRINT#2,DATE$
400 FOR X=1 TO 6:X1=1:READT$
410 IFP$(X,X1)=""THEN 420 ELSE PRINT#2,T$+P$(X,X1):X1=X1+1:GOTO4
10
420 NEXT X
430 CLOSE:CLS:GOTO520
440 GOSUB480:PRINT@464,CHR$(23);"JOB COMPLETE":FORJ=1TO500:NEXTJ
:PRINT@464,STRING$(12," "):GOTO440
450 REM ERROR ROUTINE
460 CLOSE:CLS
470 GOSUB480:PRINT@464,CHR$(23);"E R R O R":FORJ=1TO500:NEXTJ:PR
INT@464,STRING$(12," "):GOTO470
480 REM BEEP
490 OUT255,1:FORJ=1TO50:NEXTJ:OUT255,0:RETURN
500 REM READ & LPRINT TOTAL DATA FOR THE WEEK
510 CLS:INPUT"ENTER DATA FILE TO READ";B$
520 CLS:PRINT@460,"* * READING BACK ",B$;" TO CHECK * *":GOSUB49
0
530 X=1:OPEN"I",1,B$:INPUT#1,DATE$
540 IFEOF(1)THEN560:'CHECK FOR END OF FILE
550 INPUT#1,E$(X):X=X+1:GOTO540:'INPUT COMPRESSED FILE INTO E$(X
)
560 CLOSE:FORC=1TO6:FORC1=1TO50:P$(C,C1)="":NEXTC1:NEXTC:'NULL A
LL P$
570 C1=1:C2=1:C3=1:C4=1:C5=1:C6=1
580 CLS:PRINT@465,"* * STRIPPING STRINGS * *":GOSUB490
590 FORK=1TOX-1:GOSUB810:NEXTK:'LOOP # FROM READ COMPRESSED FILE
 - SUB900 TO SEPARATE COMPRESSED FILE INTO SECTORS
```

*Listing 8 continues*

*Listing 8 continued*

```
600 REM LPRINT ROUTINE
610 CLS:PRINT@460,"* * PRINTING POT INFORMATION * *";GOSUB490
620 DATA BAKERS, CAT, EAGLE, MARBLEHEAD, MISERY, PEACH
630 LPRINT TAB(20)"* * WILLIAMS --  FISH & LOBSTERS * *";STRINGS
(70-PEEK(16539),32);DATES
640 LPRINTSTRINGS(80,"*")
650 FORX=1TO6:READ MAP$:C=0
660 LPRINT"MAP DESIG............";MAP$:IFP$(X,1)=""THENLPRINT"NO
 DATA AVAILABLE FOR ";MAP$:GOTO680:'CHECK FOR NULL
670 LPRINT TAB(2);P$(X,1+C);TAB(15);P$(X,2+C);TAB(28);P$(X,3+C);
TAB(41);P$(X,4+C);TAB(54);P$(X,5+C);STRING$(67-PEEK(16539),32);P
$(X,6+C)
680 C=C+6:IFC=36THEN690ELSE670:'CHECK FOR SIXTH LINE
690 GOSUB890:LPRINTMAP$;" TTL'S..";TAB(20);"POTS=";K;TAB(31);"SH
ORTS=";SS;TAB(43);"KEEPERS=";SK;TAB(57);"RATIO KPRS/POTS..";:Z2$
="##.#";IF K=0 THEN LPRINTKELSELPRINTUSINGZ2$;SK/K:GOSUB930
700 S1(X)=SS:K1(X)=SK:'SS=SECTOR SHORTS  SK=SECTOR KEEPERS
710 LPRINTSTRING$(80,"=");K=0:SS=0:SK=0:NEXT X
720 LPRINT"TOTALS FOR PERIOD ";DATE$;".......ALL AREAS..........
...":LPRINT" "
730 LPRINT"TOTAL POTS WORKED";TP;TAB(30);"TOTAL SHORTS";TS;TAB(6
0);"TOTAL KEEPERS";TK
740 Z2$="##.##";LPRINT" ";LPRINT"% KEEPERS VS. TOTAL";INT(100*(T
K/(TK+TS)));TAB(30);"SHORTS/POT...";:LPRINTUSINGZ2$;TS/TP;:LPRIN
TTAB(60);"KEEPERS/POT...";:LPRINTUSINGZ2$;TK/TP
750 OPEN"I",1,"YEAR/TTL":INPUT#1,YEAR$:'YS=TOTAL YEAR SHORTS  YK
=TOTAL YEAR KEEPERS
760 FORX=1TO6:INPUT#1,YS(X),YK(X):NEXTX:'INPUT TOTALS FROM FILE

770 CLOSE
780 FORX=1TO6:YS(X)=YS(X)+S1(X):YK(X)=YK(X)+K1(X):NEXTX:'ADD EAC
H SECTOR TOTAL TO YEAR TOTAL
790 LPRINT" ":LPRINT YEAR$;" TO DATE TOTALS........";TAB(30);"YT
D SHORTS";YS(1)+YS(2)+YS(3)+YS(4)+YS(5)+YS(6);TAB(60);"YTD KEEPE
RS";YK(1)+YK(2)+YK(3)+YK(4)+YK(5)+YK(6)
800 LPRINT STRING$(2,138):GOTO1000
810 REM - ASSIGN C & P$
820 Z$=LEFT$(E$(K),2):IFZ$="BA"THENC=1:P$(C,C1)=RIGHT$(E$(K),LEN
(E$(K))-2):C1=C1+1:RETURN
830 IFZ$="CA"THENC=2:P$(C,C2)=RIGHT$(E$(K),LEN(E$(K))-2):C2=C2+1
:RETURN
840 IFZ$="EA"THENC=3:P$(C,C3)=RIGHT$(E$(K),LEN(E$(K))-2):C3=C3+1
:RETURN
850 IFZ$="MA"THENC=4:P$(C,C4)=RIGHT$(E$(K),LEN(E$(K))-2):C4=C4+1
:RETURN
860 IFZ$="MI"THENC=5:P$(C,C5)=RIGHT$(E$(K),LEN(E$(K))-2):C5=C5+1
:RETURN
870 IFZ$="PE"THENC=6:P$(C,C6)=RIGHT$(E$(K),LEN(E$(K))-2):C6=C6+1
:RETURN
880 IFE$(K)="END"THEN460
890 REM ASSIGN SS-SK
900 IFP$(X,1)=""THENSS=0:SK=0:RETURNELSEIFX=1THENK=C1-1ELSEIFX=2
THENK=C2-1ELSEIFX=3THENK=C3-1ELSEIFX=4THENK=C4-1ELSEIFX=5THENK=C
5-1ELSEIFX=6THENK=C6-1
910 FORK1=1TOK:SK=SK+VAL(RIGHT$(P$(X,K1),1)):TM$=RIGHT$(P$(X,K1)
,3):SS=SS+VAL(LEFT$(TM$,1))
920 NEXTK1:RETURN
930 TP=TP+K:TS=TS+SS:TK=TK+SK:RETURN
940 REM OUTPUT TO DISK YTD TOTALS
950 CLS:PRINT@460,"* * UPDATING YTD FILES * *":GOSUB490
960 OPEN"O",2,"YEAR/TTL":PRINT#2,YEAR$
970 FORX=1TO6
980 PRINT#2,YS(X),YK(X):NEXTX:'YS=YEAR TOTAL SHORTS  YK=YEAR TOT
AL KEEPERS
990 CLOSE:CLS:GOTO440
1000 REM.....START OF GRAPH
1010 LPRINTSTRING$(2,138):V$=":":H$=".":A$=" ":G1$="SSSSSSS":G2$
="KKKKKKK":G3$="*     *":G4$="BBBBBBB":G5$="B     B"
1020 LPRINTTAB(21);"* * WILLIAMS -- FISH & LOBSTERS * *";STRING$
(70-PEEK(16539),32);DATE$:LPRINT" ":LPRINTSTRING$(80,"*"):LPRINT
" ":L=50
1030 GOSUB1120:LPRINTTAB(2)L;TAB(5);V$;:GOSUB1180:L=L-1:IFL=39TH
EN1040ELSE1030
1040 READA$:GOSUB1120:LPRINTA$;TAB(2);L;TAB(5);V$;:GOSUB1180:L=L
-1:IFL=11THEN1050ELSE1040
1050 A$=" ":GOSUB1120:LPRINTTAB(2)L;V$;:GOSUB1180:L=L-1:IFL=9THE
N1060ELSE1050
1060 GOSUB1120:LPRINTTAB(3)L;TAB(5);V$;:GOSUB1180:L=L-1:IFL=0THE
N1070ELSE1060
1070 LPRINTTAB(6)STRING$(64,H$)
1080 LPRINTTAB(11)"BAK";TAB(21);"CAT";TAB(31);"EAG";TAB(41);"MAR
";TAB(51);"MIS";STRING$(61-PEEK(16539),32);"PEA":LPRINT" "
1090 LPRINTTAB(25);"* * MAP AREA DESIGNATIONS * *"
1100 LPRINTSTRING$(52,138):GOTO940
1110 DATA*,,*,,A,,R,,E,,A,,,T,,O,,T,,A,,L,,S,,*,,*
1120 FORN=1TO6 : REM.....ASSIGN A$(1)>A$(6)
1130 IFS1(N)=K1(N)ANDS1(N)=LTHENA$(N)=G4$:GOTO1170
1140 IFS1(N)=K1(N)ANDS1(N)>LTHENA$(N)=G5$:GOTO1170
1150 IFS1(N)=LTHENA$(N)=G1$ELSEIFK1(N)=LTHENA$(N)=G2$:GOTO1170
1160 IF S1(N)>L OR K1(N)>L THEN A$(N)=G3$
1170 NEXTN:RETURN
1180 REM........PRINT LINE
1190 LPRINTTAB(9);A$(1);TAB(19);A$(2);TAB(29);A$(3);TAB(39);A$(4
);TAB(49);A$(5);TAB(59);A$(6);STRING$(69-PEEK(16539),32);V$;L;"
";A$
1200 RETURN
```

*Build an ideal workplace for your TRS-80*

# $30 System Desk

*Sanford J. Asman*
*59 Willis Drive*
*West Trenton, NJ 08628*

After accumulating the various components of my TRS-80 computer system (a Level II Model I with an expansion interface and 48K of RAM, an MPI disk drive, and an Anderson-Jacobson AJ841 printer) I realized the office desk the components had been accumulating on was becoming less and less accessible. In addition, it was harder and harder to avoid tripping over the rats nest of wiring created by six power cords plugged into various wall outlets. Finally, the computer itself had never been at an ideal height; the desk top was too high for convenient keyboard entry or for viewing the elevated monitor. The system, expanded from the original 16K computer, had clearly outgrown the desk on which it was sitting.

I began looking for an alternative arrangement. My first thought was that the $200 system desk sold by Radio Shack, while not an insignificant investment, might be worthwhile. However, because Radio Shack designed their system desk with the computer and expansion interface openings on its left side, the desk would not fit in the location I intended to place it. Also, its modern appearance did not fit the decor of the room, it did not eliminate the multiple electrical outlets required to operate the system components, and I did not like placing the disk drive into a cubby hole. The price tag was steep for a desk that failed to be ideal.

On the other hand, the openings in the Radio Shack system desk would eliminate much of the visible wiring and would provide for free air circulation around the computer and expansion interface, both of which are cooled solely by convection.

### My Own Design

I decided to build my own computer system desk. I created a design I like and which is a part of the system, as well. It provides recesses for the computer and the expansion interface which allow for hidden wiring and adequate convection cooling, and the need for multiple power outlets is eliminated by including a six electrical outlet unit controlled by a single switch. The unit has a single power cord going from the desk to a wall outlet; thus, the entire system is turned on or off with one switch. There is space on the desk for my printer, disk drive, and floppy disk storage, and the desk fits the room decor. Best of all, the desk was simple to build, and the wood cost less than $30.

I have learned to build things that don't require exacting measurements. Rather than measuring, I mark one piece directly against the piece with which it fits, so the fit is certain and I needn't worry about whether I can measure accurately. Where measurements are absolutely required, I make them as uncritical as possible.

I decided to incorporate the openings of the Radio Shack system desk, but to make the desk in the form of a Parson's table. This leaves room on the left side of the computer for the printer and on the right side for the disk drive and disk storage. The tape recorder can occupy space behind the disk drive. The only important dimensions were the overall size of the desk top and the height of the desk (low enough to type comfortably, high enough to fit my knees under it).

The result is shown in the photograph and described more fully in the plans. While I have included the actual measurements of my desk, you should follow the general principles rather than the specifics to get the desk that completes your system.

### Non-Critical Cuts

I wanted a top surface of 49¾ inches by 28¼ inches. The top is made out of a piece of 5/8 inch particle board, an extremely hard, dense material which comes in 4 foot by 8 foot sheets selling for about $6 per sheet. To keep things simple while providing a finished edge (apron), I sawed a 26½ inch piece off one end of a sheet of particle board. The 48 inch by 26½ inch piece would provide me with the desired top surface after the apron was added. To ensure a straight cut, saw the top from the sheet using a sabre saw guided along a straight board clamped to the sheet by a pair of C-clamps or adjustable wood clamps.

After cutting the desk top from the sheet, make the openings for the computer and the expansion interface. Lay out the openings in the top and follow the lines with a hand-guided sabre saw. Start the cuts by drilling a hole some place within the openings or by using a plunge cut with the sabre saw. A

***The Key Box***

**Model I**
**Expansion Interface**
**Printer**

plunge cut is made by starting the saw and then tilting it into the waste area of the opening. The openings for the computer and the expansion interface are sized to allow wires to pass through and are not critical, so do not worry if your measurements are off; just try to keep the cuts fairly straight. I rounded the corners by drawing arcs with a quarter. Without a single critical cut or measurement, the top is complete.

I selected poplar for the apron; clear pine could be used and is more widely available, but I prefer to use hardwoods and suggest poplar, walnut, maple, or oak. The width of the apron is not critical, but for best appearance it should be approximately the same as the legs, which are made of square stock. I chose to use stock that was 7/8 inch thick by 2⅞ inches wide, but these dimensions are not critical. 1 by 3 clear pine, which has actual dimensions of ¾ inch by about 2¾ inches, could also be used.

Next the end pieces are cut to length. Since they should be the same length as the ends of the desk top, mark them off directly against the desk top and cut them to length. The front and back apron pieces are cut to equal the lengths of the front and the back of the desk top (which might not be exactly equal) plus twice the thickness of the apron material.

After the apron pieces have been cut to size, the legs are cut to length from square stock. I used 2½ inch square poplar, but 3 by 3 fir can be used. The length of the legs is equal to the height of the desk top, minus the final thickness of the top. My desk has a top that is 5/8 inch thick and legs that are 25¼ inches long. This provides me with adequate knee clearance under the front apron (23½ inches), while keeping the keyboard and monitor at comfortable typing and viewing heights (on a desk 25-7/8 inches high).

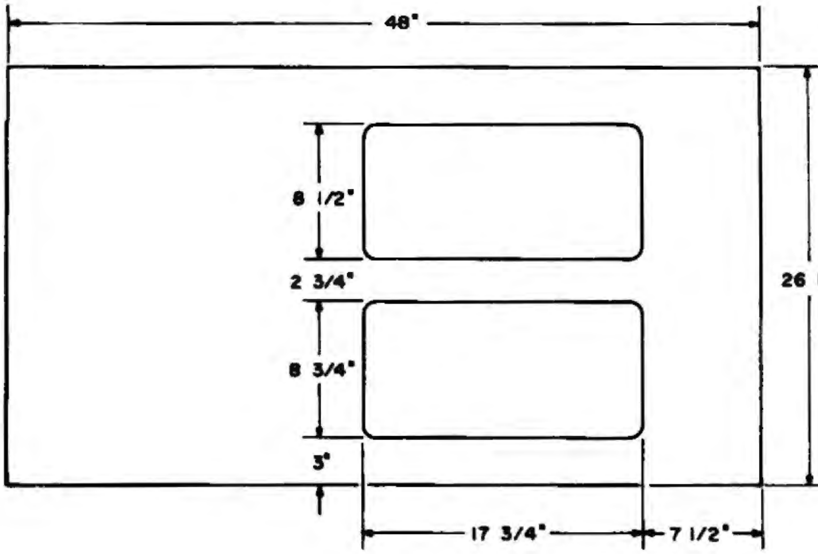
The tops of the legs have to be dadoed on two adjacent sides to receive the aprons. The depth of the dadoes equals the thickness of the apron material (7/8 inch for my desk). The dadoes extend down from the tops of the legs by an amount equal to the width of the apron material minus the final thickness of the top (1¾ inches for my desk). The dadoes are most easily cut on a radial arm saw or on a table saw, but a hand saw, a router, or a circular saw could be used instead.

Begin assembling the desk by attaching the aprons to the legs. Make sure the legs are square to the aprons and to the ground. The desk can be assembled using finishing nails or screws, or glue and clamps. The top should fit into the assembled frame, resting on the tops of the legs.

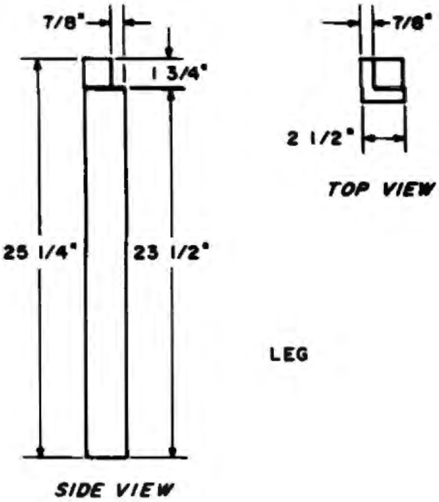
*Photo 1. The finished desk.*



✓ See List of Advertisers on Page 451




DESK TOP
(5/8" PARTICLE BOARD)



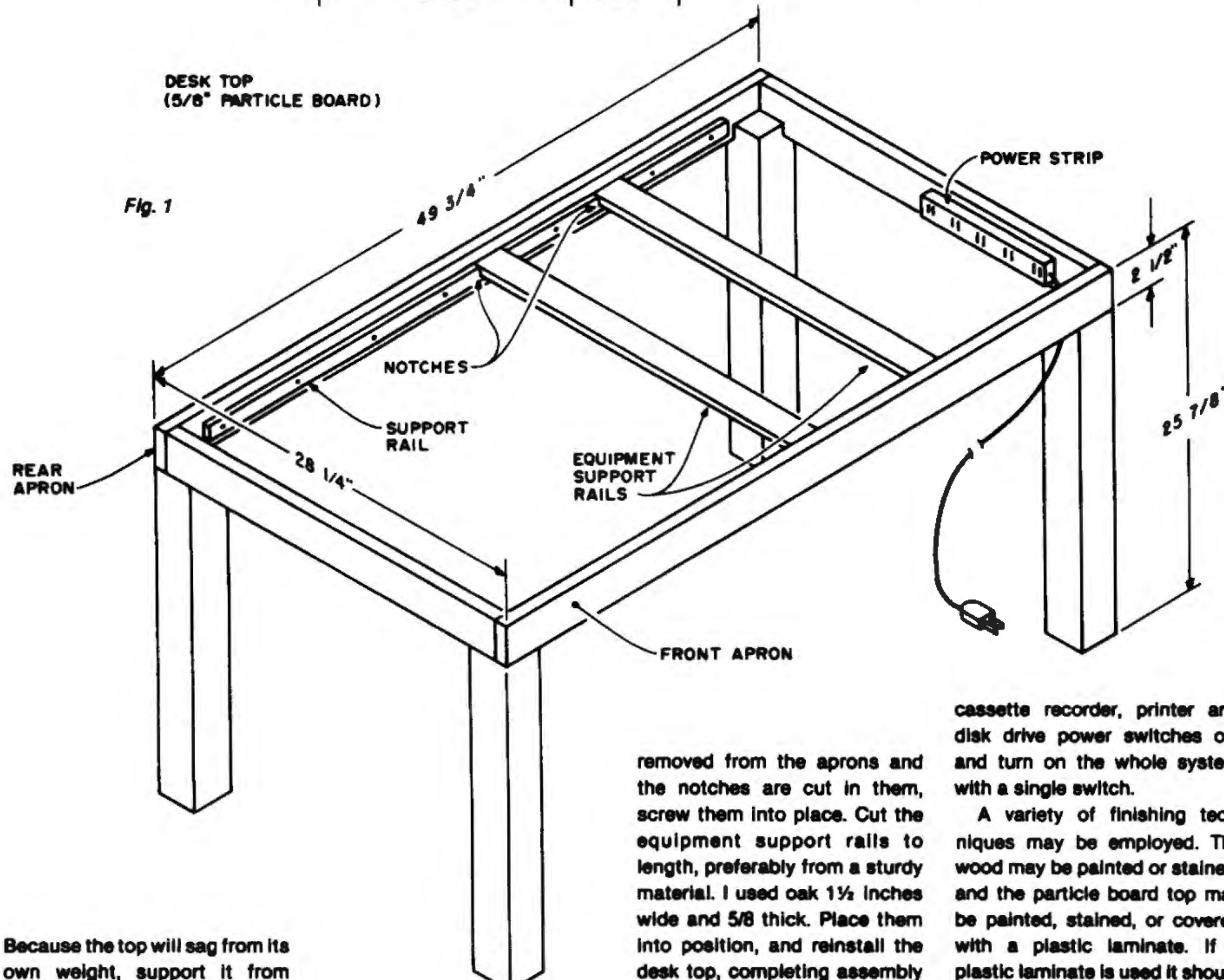




Fig. 1

Because the top will sag from its own weight, support it from below with rails screwed to the inside of the front and back apron pieces. The support rails can be any available scrap material, but their dimensions do not matter. Place and clamp the support rails into position when the desk top is flush with the top of the apron. Then remove the desk top, and drill pilot holes for screws through the support rails into the front and rear apron pieces. Mark locations for notches in which equipment support rails for the computer and expansion interface rest in the support rails. After the support rails are removed from the aprons and the notches are cut in them, screw them into place. Cut the equipment support rails to length, preferably from a sturdy material. I used oak 1½ inches wide and 5/8 thick. Place them into position, and reinstall the desk top, completing assembly of the desk.

I installed a switched six outlet unit on the inside of the right apron, and ran all the wires through the openings in the desk top. The outlet unit which I used includes a single switch that controls power to all the outlets; I can leave the computer, expansion interface, video monitor, cassette recorder, printer and disk drive power switches on, and turn on the whole system with a single switch.

A variety of finishing techniques may be employed. The wood may be painted or stained, and the particle board top may be painted, stained, or covered with a plastic laminate. If a plastic laminate is used it should be applied after the top is cut out. The thickness of the laminate should be considered when the thickness of the top is a factor in the construction of the desk (the length of the legs and the length of the dadoes in the legs will have to be decreased by the additional thickness of the plastic laminate). ■


DATE
DOLLARS
POWER

# We don't care which computer you own. We'll help you get the most out of it.

## CompuServe puts a world of information, communications, and entertainment at your fingertips.

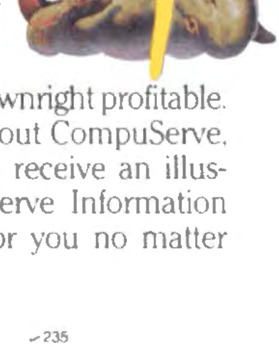
**CompuServe is** the versatile, easy to use interactive videotex service designed especially for the personal computer user. It's dynamic, growing and changing daily to satisfy its subscribers' needs. It's an industry leader, created and managed by the same communications professionals who provide business information and network services to over one fourth of the FORTUNE 500 companies.

**From current events to current assets,** CompuServe offers a wealth of useful, profitable or just plain interesting information. Electronic magazines and national news wires plus worldwide weather, current movie reviews, electronic banking and shop at home services, and some of the most sophisticated financial information available are all offered to current subscribers.

**From words to music.** CompuServe offers a communications network that gives special interest groups from hardware enthusiasts to computer composers a chance to get together. There's a bulletin board for selling, swapping, and personal notices and a CB simulator for real-time communications between subscribers. There's electronic mail, the fastest, surest, way to communicate with other users across the street or across the country, plus file retention and editing, and lots, lots more.

**Fun and games** are expected whenever computer users interact, and CompuServe has the best. Games you can play alone or with other CompuServe subscribers anywhere in the country. Classic puzzlers, sports and adventure games, and fantastic space games featuring MegaWars, the "ultimate computer conflict."

But, that's just the tip of the chip. CompuServe offers a menu of thousands of items that make subscribing educational, fun and sometimes downright profitable. If you'd like to know more about CompuServe, call toll free, 800-848-8990 to receive an illustrated guide to the CompuServe Information Service. A videotex service for you no matter which computer you own.



**CompuServe** ✓235

P.O. Box 20212
5000 Arlington Centre Blvd. Columbus, Ohio 43220
**800-848-8990**
In Ohio call 614 457 8650
An H&R Block Company

*Add a disk controller to your system.*

# Hardware Hacker—Part III

*Philip M. Van Praag*
*1630 West Jagged Rock Road*
*Tucson, AZ 85704*

This article describes the design and construction of a disk drive controller for about $200. The controller is compatible with the TRS-80 Model I, does not require the Expansion Interface, and handles up to four 5¼ inch disk drives. Its features include integral double density data separation to eliminate a potential source of read errors, single and double density operation to maintain compatibility with existing software and optimize disk storage capacity, an operator-selected disk drive disable switch enabling normal use of the TRS-80's reset function, a fully buffered interface between the controller and the disk drives, a double density status indicator and digital write precompensation to optimize data recovery accuracy.

The main logic board includes a connector card edge for mating to the standard disk drive ribbon cable assembly. The controller does not require any modification to the TRS-80 and may be combined with the 32K Memory Mod or the Centronics-compatible printer interface (presented in previous articles) or both if you desire those additional features.

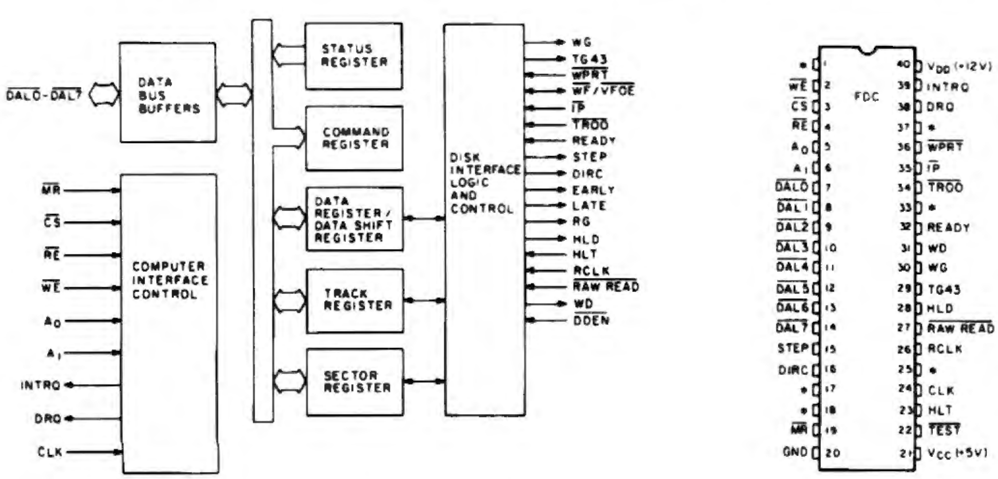
The disk controller's most useful feature may be its ability to operate in double density mode. Although you must purchase a double density DOS to use this feature, it doubles the storage capacity of your disk, eventually reimbursing you for the purchase. This is accomplished by virtually eliminating clock bits (interspersed with data bits in single density operation) and doubling the data transfer rate.

#### Requirements

The heart of the disk controller is a large-scale integrated circuit (LSI) called a floppy disk controller (FDC). As the key link between the TRS-80 and the disk drives, it performs two major tasks: first, coordinating the writing and reading of data; and second, supervising, interpreting and in some cases generating the control signals necessary for disk operation. Although the processes by which the FDC performs these functions become rather complex, all functions fall within these two general categories.

The TRS-80 always manipulates data in a parallel mode; that is, all eight data bits defining a character or instruction are presented on the eight-line data bus simultaneously. The disk drives, however, must handle data serially (one bit at a time). The FDC converts parallel data to a serial form before writing it onto the disk, and converts serial data from the disk to parallel before sending it back to the TRS-80. A data register within the FDC holds the parallel-form data, and a data shift register converts it from parallel to serial and vice versa (see Fig. 1). In addition, the data must be precisely clocked when in the write mode, and this clocking must be properly interpreted when reading data from the disk.

The FDC's clock bit/data formats are referred to as frequency modulation (FM) in the single density mode, and modified frequency modulation (MFM) in the double density mode. These formats are illustrated in Fig. 2. To understand how these FM sys-




*UNIQUE PIN FUNCTIONS

| PIN | 1771 | 1791 |
|---|---|---|
| 1 | $V_{BB}$ (-5V) | NC |
| 17 | PH3 | EARLY |
| 18 | $\overline{3PM}$ | LATE |
| 25 | $\overline{XTDS}$ | RG |
| 33 | $\overline{WF}$ | $\overline{WF}/\overline{VFOE}$ |
| 37 | $\overline{DINT}$ | $\overline{DDEN}$ |

*Fig. 1. Internal functional diagram and pin assignments for FDC*

tems operate, let's look at how the disk is magnetically divided to store data.

Five levels of data organization are defined and controlled by hardware and software design. In order of size, these are tracks, granules, sectors, bytes and bit cells.

Figure 3 shows two levels of data organization on the disk: tracks (concentric rings around the disk center) and sectors (which divide the tracks into discrete pieces). A small hole in the disk (not shown) defines the relative position of the sectors around the tracks. Granules, a specific number of sectors determined by the DOS, define the smallest accessible block of data when writing to or reading from the disk. (For example, TRSDOS uses five sectors per granule, defining the smallest possible file size.) The smallest level of division is the bit cell. Each byte contains eight data bits.

This brings us back to Fig. 2. Each bit cell recorded on the disk contains two pieces of information: definition of the physical boundaries of that bit cell, and the data bit (a binary logic level high or low). In the single density mode, both data bits and clock bits are recorded. A pulse between two clock pulses defines a logic high for the bit cell. If no pulse exists, a logic low is defined for that cell. In the double density mode, clock bits are not required at each bit cell location. Instead, the relative position (or in some cases, the lack) of a logic high level bit at each bit cell location defines the cell and determines whether the cell data is a logic high or low. The rules of MFM data bit placement are in Fig. 2.

To summarize data organization on the disk, consider a few examples. In single density mode using a 35-track single-sided disk drive, there are 8 bit cells to define each byte of data, a 256 byte capacity in each sector, 10 sectors within each track, and a total of 35 tracks. Multiplying, we have 8 times 256 times 10 times 35, or 716,800 defined bit cell locations. In double density mode using a 40-track single-sided drive, there are 8 bit cells per byte, 256 bytes per sector, 18 sectors per track and 40 tracks. This translates to 1,474,560 defined bit cell locations! Although disk storage capacity is usually specified in bytes, remember that the smallest unit of division is really the individual bit cell.

The FDC processes three general categories of signals: TRS-80-generated data, address bus codes and control signals, disk drive-generated data/feedback signals, and FDC-generated control and feedback signals.

Table 1 lists the TRS-80-generated addresses decoded by the FDC and support circuitry. These decoded addresses perform two basic functions: supplying DOS commands to operate the drives and perform housekeeping chores for FDC support circuitry; and furnishing TRS-80 memory locations (actually accessed FDC registers providing information back to the TRS-80). The TRS-80 provides read (RD), write (WR), and system reset (SYSRES) control signals. The read and write signals define the function of the TRS-80 addresses listed in Table 1 as well as provide read/write disk drive mode instruction to the FDC. System reset, used during TRS-80 power up and reset modes, clears the FDC status register, establishes single density operating mode and performs other initialization chores in the FDC.

Disk drive-generated data and

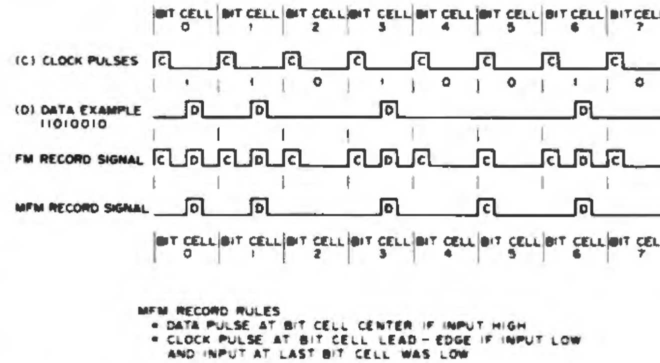



*Fig. 2. Single density (FM) and double density (MFM) recording formats*

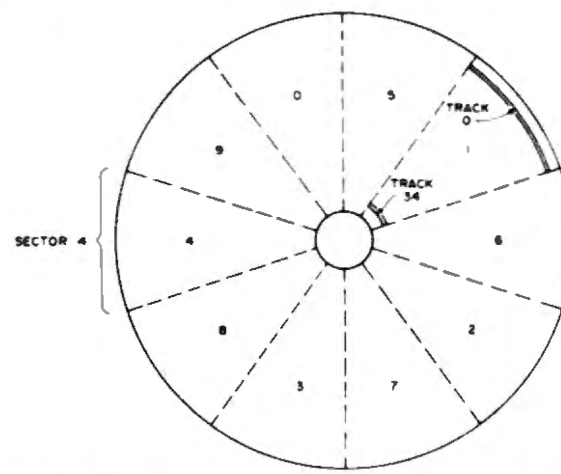



*Fig. 3. Disk data organization (single-density track standard shown)*

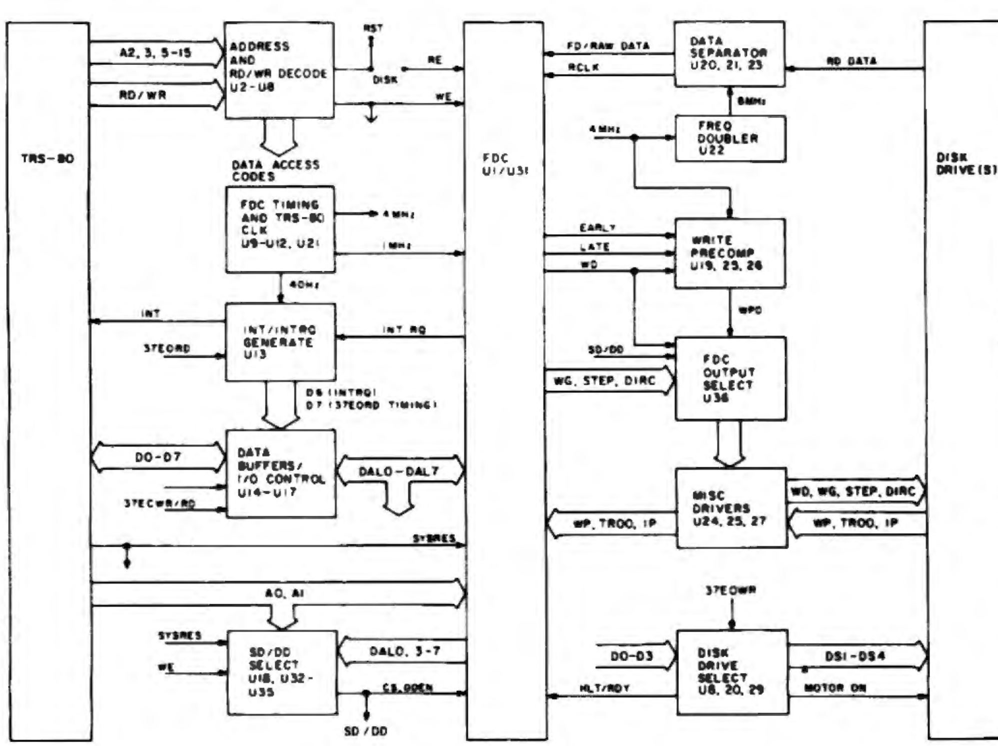



*Fig. 4. Disk controller logic circuitry*

↙See List of Advertisers on Page 451

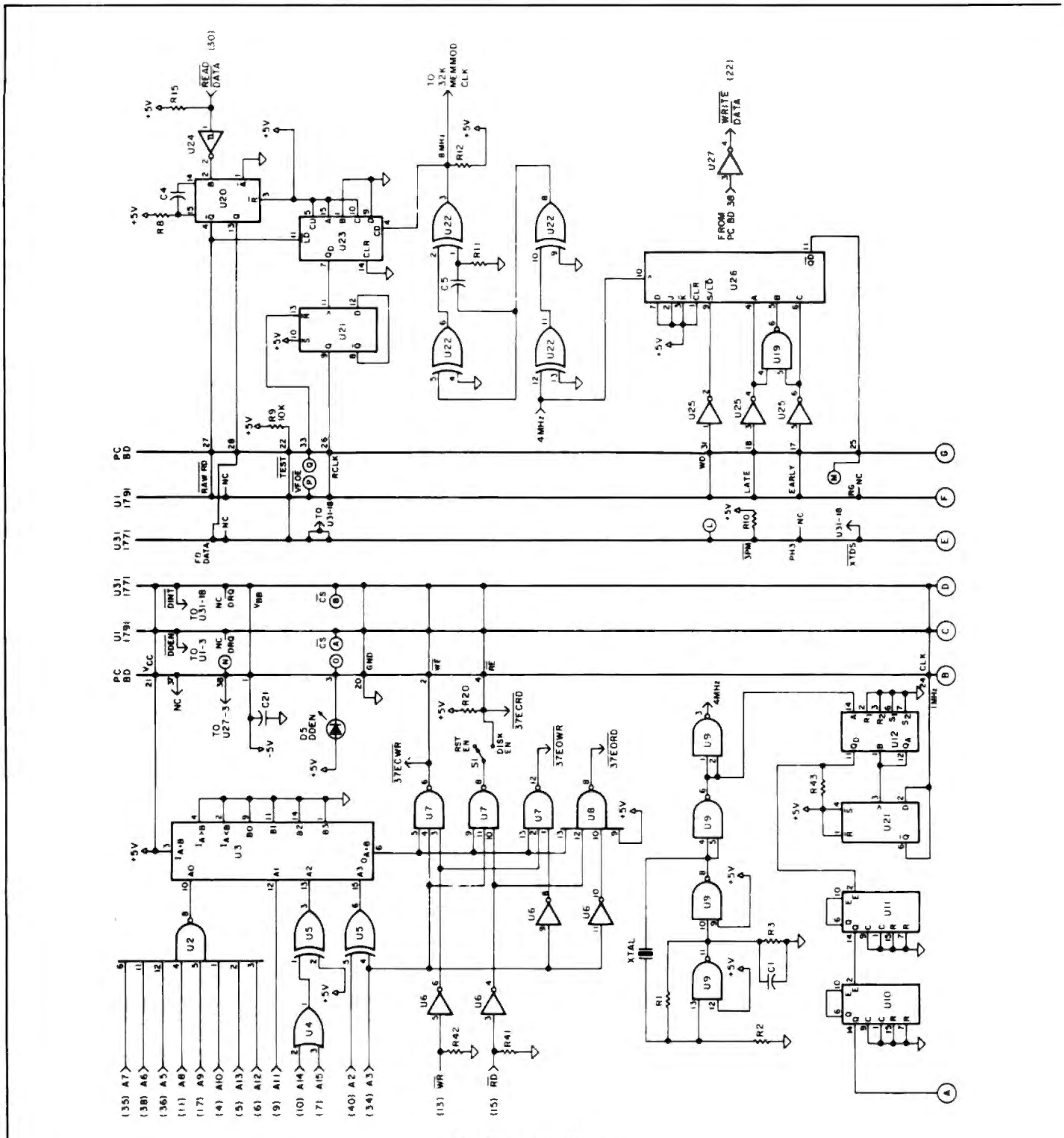
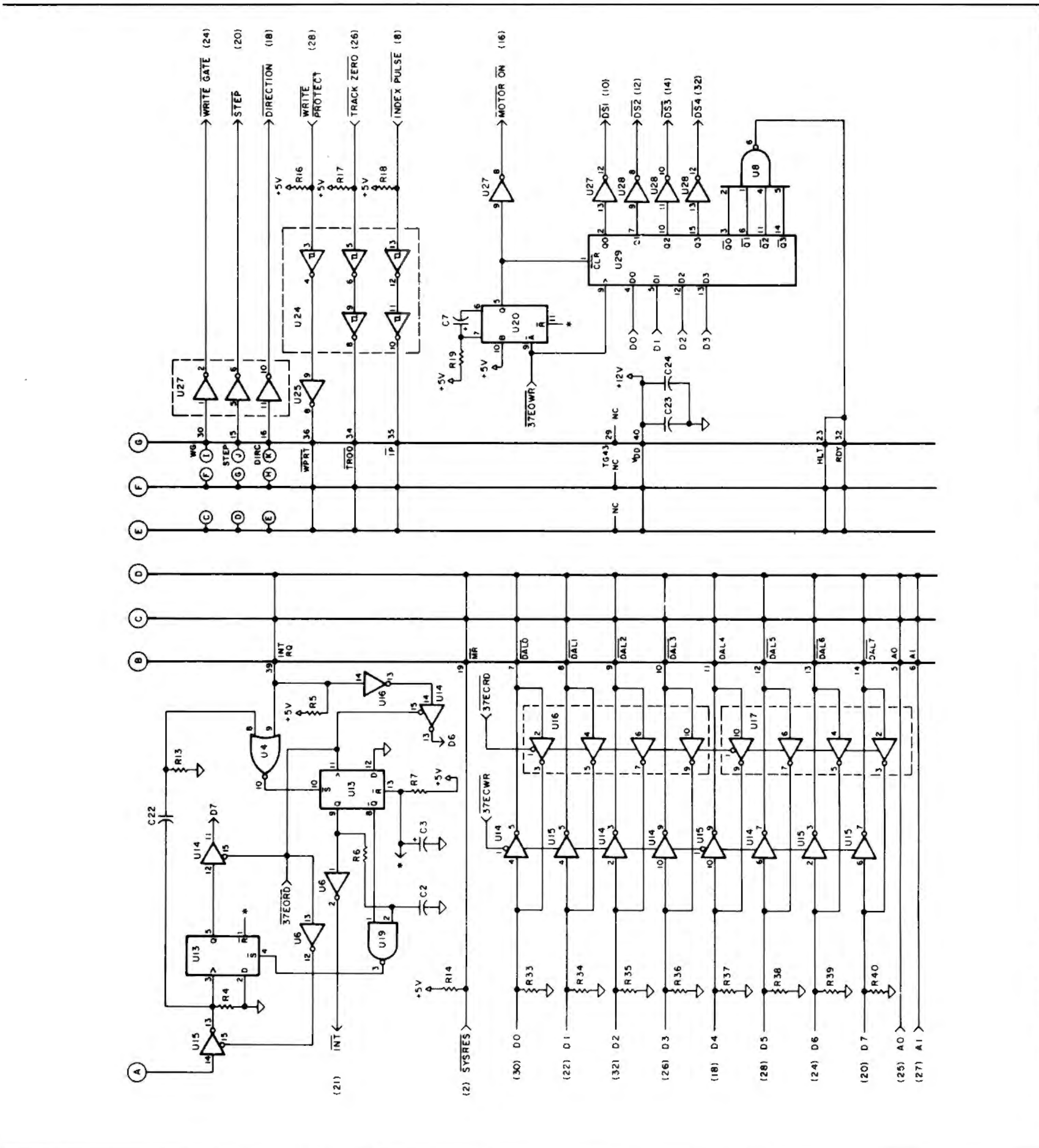


*Fig. 5a. Logic circuitry*

feedback include read data (RD DATA), track zero (TR00), index pulse (IP), and write protect (WPRT) signals. The read data signal contains serial clock and data bit pulses. The track zero signal indicates when the read/write head is positioned on the outermost track (track zero). The index pulse signal provides disk sector identification by producing a pulse once per revolution when the index hole is sensed. If the disk is write protected the Write command is terminated. This line is sampled every time the TRS-80 issues a Write command.

The FDC and its support circuitry generate the following control and feedback signals: interrupt request (INT RQ), interrupt (INT), and various register output signals going to the TRS-80, as well as step (STEP), direction (DIRC), write gate (WG), and write data (WD) signals going to the disk drives. VFO enable (VFO E), EARLY, LATE, head load timing (HLT), ready (RDY) and read clock (R CLK) signals are used for internal control. Additional FDC I/O capabilities not used in this project will not be discussed.

First, let's discuss FDC signals going to the TRS-80. An interrupt request signal is generated whenever an FDC operation is completed, and is reset when a new command is loaded into the command register. The interrupt request is sent to the TRS-80 on the bit-6 data line when 37EORD is present. Its presence also causes an interrupt signal to be generated. The interrupt signal is reset at the




termination of 37EORD.

The five FDC registers listed in Table 1 communicate with the TRS-80 via the bidirectional data bus. The TRS-80 selects a particular register by providing a unique address (listed in the table). These addresses share the same binary address code except for the two least significant bits. Table 2 illustrates this relationship.

Now, consider FDC signals going to the disk drive. The step and direction signals control movement of the head stepping motor. A high level on the direction line sets the head inward toward the higher track numbers, while a low level sets the head outward. High level pulses on the step line move the head in increments of one track per step pulse. The write gate essentially gives permission for data to be written onto the disk, while the actual composite data (clock and data bits) is sent to the disk drives on the write data line.

Finally, let's look at FDC signals used for internal control. VFO enable indicates that the disk head is in contact with the disk and data pulses are being received. It disconnects read clock pulses when these conditions are not met, preventing read-data-line noise from affecting the FDC during non-read operations. Early and Late, generated by an FDC algorithm during write mode, aid write precompensation efforts. HLT and RDY inform the FDC that the disk drive is ready for read or write operations.

Table 3 summarizes the sig-

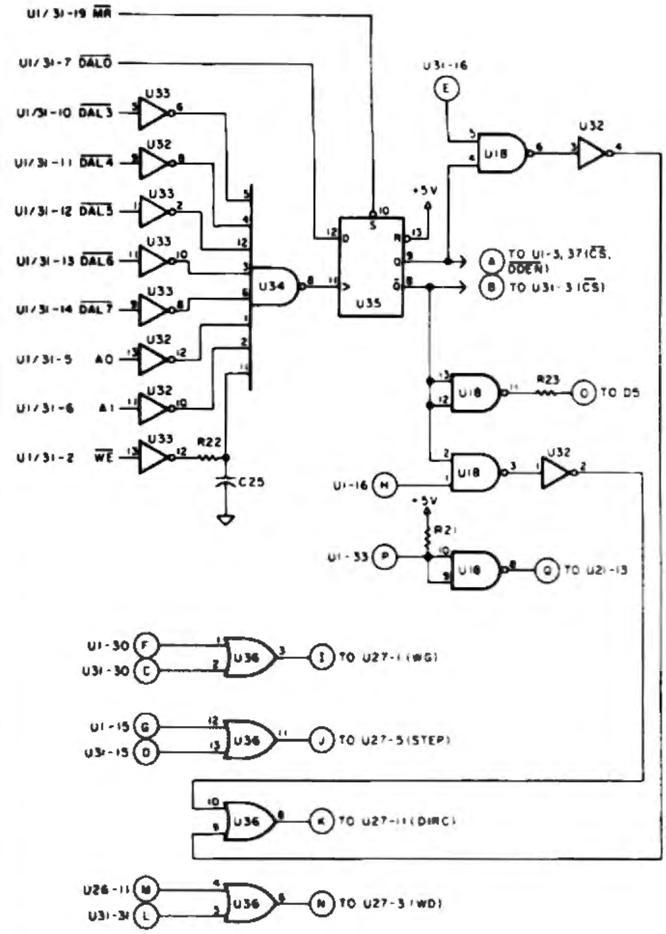



*Fig. 5b. Disk controller piggyback board*

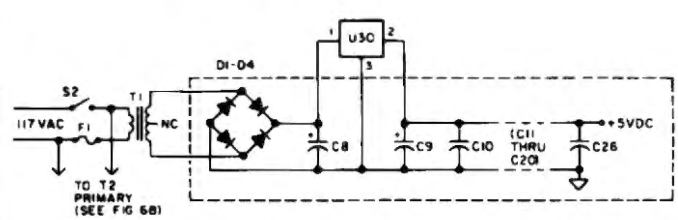



*Fig. 6a. +5 V power supply*

nals necessary to provide data and control for the disk drives.

**How the Circuit Works**

Figure 4 contains a block diagram and Fig. 5 schematic diagrams of the disk controller logic circuitry. Table 4 contains all the miscellaneous circuit connections not shown in Fig. 5, including IC power/ground requirements and terminations for unused IC sections.

The address and read/write decoder function is accomplished by U2 through U8. U3, a 4-bit magnitude comparator, is the central focus of the decoder. It provides a high level logic signal at pin 6 whenever its two sets of 4-bit inputs are identical. Following address bus logic levels through the input gating, pin 6 of U3 will be high when 37E0,1,2,3,C,D,E, or F appears. The presence of read and write signals, plus the gating of U7 and U8, yields the four desired outputs: 37E0WR, 37E0RD, 37ECWR, and 37ECRD. A0 and A1 address lines go directly into the FDC to be used for register access together with the decoded 37ECWR and 37ECRD signals (see Table 2). The decoder does not need the A4 address line as it is always assumed low. (If it were high, with all other disk-related addresses set for disk operations, an address in the 37FX range not used for any TRS-80 function would result). Switch S1 breaks the read communication link between the TRS-80 and the disk controller when in Reset Enable mode, fooling the TRS-80 into believing the controller is disconnected. This achieves normal operation of the Reset button on the back of the keyboard.

FDC timing and TRS-80 clock signals are provided by U9 through U12, U21 and U22. U9 functions as a 4 MHz crystal-controlled oscillator. This signal is the basis for all timing signals produced by the disk controller, including a 1 MHz FDC clock signal, a 40 Hz TRS-80 clock signal, a 4 MHz write precompensator signal and an 8 MHz timing signal used by the data separator. The 4 MHz clock is divided by two via a section of U12 and again by two using part of U21, producing the 1 MHz FDC clock. The 4 MHz clock is also divided by 10 in U12, by 100 in U11, and again by 100 in U10 to produce the 40 Hz clock signal used by the TRS-80's internal clock display routine. The 8 MHz timing signal produced by U22 functions as a frequency doubler. U22-11, U22-8, and U22-6 provide a phase-delayed 4 MHz signal to one input of U22-3. C5 and R11 produce a differentiated version of this signal for the other input. The exclusive-or function of U22-3 produces a high level output when its inputs are at different logic levels. In this application, that occurs once for every half-cycle of the 4 MHz clock, thus producing pulses at an 8 MHz rate.

U13 and its related logic circuitry—portions of U4,6,14,15, and 19—produce interrupt and interrupt-request signals. This circuitry receives two input signals: interrupt request as generated by the FDC, and the 40 Hz clock signal from U10-1. When U1-39 goes high, an interrupt signal is produced at U6-2. An interrupt is also produced when a 40 Hz signal occurs, as long as 37EORD is not active. The 37EORD signal also gates the status of the 40 Hz latch (U13-5) to the D7 data bus line, and gates INT RQ to the D6 data bus line. The termination of 37EORD, a positive-going edge, clocks a low level through U13-9 which resets the interrupt signal and, via U19-3, resets U13-5.

U14 through U17 buffer the bidirectional data bus between the TRS-80 and the FDC. These are tri-state buffers: Their outputs assume a high impedance state as long as they are disabled. All input buffers have one common enable line 37ECWR), as do all output buffers (37ECRD). The address 37EC is a memory location in the TRS-80 system to which data can be written or from which data can be read. Remember, however, that the A0 and A1 address lines are not part of the 37EC signal decoder. Therefore, enabling of the buffers also occurs for addresses 37ED, 37EE, and 37EF (see Table 2). The A0 and A1 lines, going directly to the FDC, determine which FDC registers will be accessed.

The data separator function, including U20, 21, and 23, is used in double density mode. (I didn't include a single density data separator. Considering the greater efficiency of double density, you are likely to use single density only while converting your existing programs to double density. If interest is sufficient, I will design a single density separator for a later article.) It works with two input signals—read data and a clock signal—and produces two output signals—raw read (RAW RD) and read clock. The term "data separator" as used with the 1791 is actually a misnomer as it does *not* separate data and clock bits. Rather, it is a data recovery system which outputs both data and clock bits on the raw read line to the FDC, and also provides a strobe or window signal for U1's R CLK to clock the raw read pulses. If a raw read pulse occurs outside the window, it will cause an FDC read error. The data separator exploits the relatively predictable recurrence of data bits to produce a clocking window which predicts the occurrence of the next data bit. The phase of the window is continuously corrected to compensate for motor speed variation and read cir-

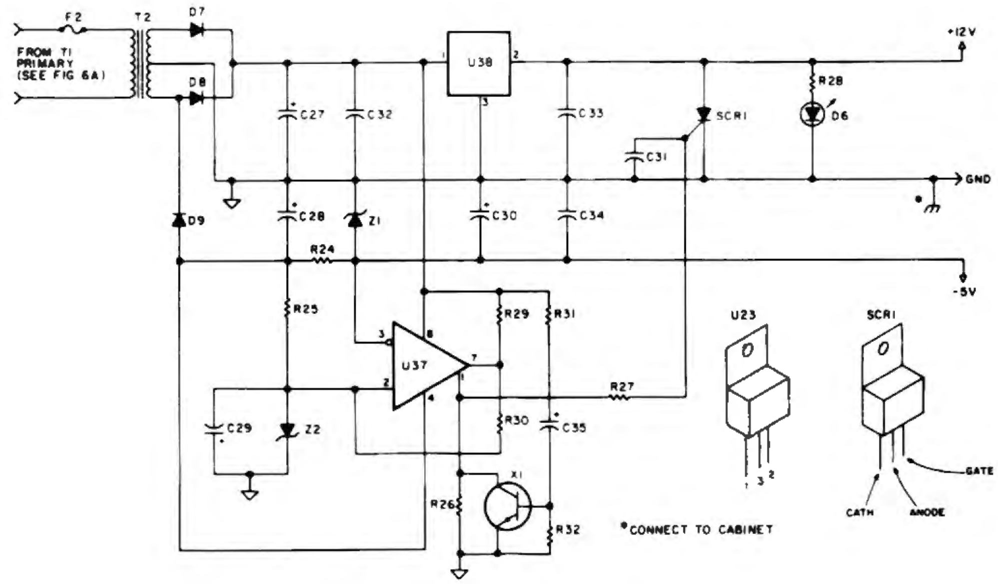



*Fig. 6b. + 12/ − 5 V power supply*

cuitry phase shifts. Remember that the read clock signal is *not* the clock bit signal coming off the disk.

U20-4,13 is a monostable oscillator which produces fixed-duration pulses for the FDC based on incoming read data bits. U23 is a 4-bit count-down counter that is continually clocked at an 8 MHz rate. U21-9 divides the counter output by two, forming the read clock signal. However, this signal is terminated when the system is not in read mode by controlling U21's reset input with U1's VFOE signal.

When nothing is present on the read data line, U23 divides the incoming clock by 16. Thus, read clock signals arrive at the FDC at a 250 KHz rate. When a pulse occurs on the read data line, U20-4 goes low, which loads a 5 into the counter. Assuming the pulse occurs during a window period, this loading action ensures that the window remains open long enough to admit the entire pulse. The loading action also phases the clock signal to the read data rate by adjusting the period of the window (and the length of time before another window opens) to suit the incoming data. This error-correcting phasing action is the essence of accurate data recovery.

Write precompensation is accomplished by U26 and portions of U19 and U25. It is used only in double-density mode to enhance data transfer accuracy. U26, a shift register, continually outputs a low level at pin 11 when no write data is present. This is a result of being in the shift state with the J and K inputs connected as shown. When a write data pulse occurs, U26 goes into the load state, during which one of the A, B, or C inputs receives a low level signal. Termination of the write data pulse reinstates the shift mode resulting in that low level bit being stepped through the shift register. The bit is output at pin 11 as a high level pulse some time later. The delay between the occurrence of the FDC's write data pulse and the output at U26-11 is a function of which U26 input received a low level during the load operation. The FDC determines the optimum delay that should be given to the present data pulse, based on previous write data bit patterns. As a result, a LATE or EARLY signal may be sent to U26 along with the data pulse. If neither an EARLY nor LATE signal appears, the B input (U26-5) will be low. This produces a nominal delay which would have been 250 nanoseconds longer if a LATE pulse was desired, or 250 nanoseconds shorter if an EARLY pulse was desired.

The FDC write gate, step, direction, and write data output lines are incapable of tristate operation. Thus these outputs must be isolated between the two FDCs, accomplished by U36. Fortunately, these output lines go to a logic low level when not selected, making a simple OR gate configuration possible. There is one exception: After certain operating sequences, the direction output from the operating FDC remains at a high level after switching to the other FDC. To correct for this anomaly, the direction outputs are gated with the chip select (CS) lines, ignoring the non-operating direction output line.

U27 and U28, high-current TTL buffers between the FDC and the disk drives, are capable of driving four disk drives on the common output bus. U24 is a buffer containing Schmitt-trigger action at its input to optimize signal recovery.

Switching between single and double density operation is accomplished by U32–U35. U32 through U34 form a simple decoder similar to the read/write decoder discussed earlier. The primary difference is that here several data lines are decoded. DAL0 determines whether single or double density mode is selected. If DAL0 is high when U35-11 is clocked, single density is selected; if DAL0 is low, double density is selected. The master reset (MR) line, called SYSRES in the TRS-80, sets U35 to single density at powerup and any time Reset is pressed. U18-11 provides a drive signal to an LED indicating double density operation.

The disk drive select circuitry activates the desired disk drive, informs the FDC when a drive is selected, and provides an enable signal with time-out protection for the drive motors. U29 is a four section latch clocked by 37EOWR. If 37EOWR does not go low at least once every three seconds U20-5 will go low and U29 is cleared, turning off the drive motors and the previously selected drive. When any drive is selected U8-6 will be high.

**Construction**

Table 5 contains the parts lists for this project. Use of printed circuit boards is almost mandatory due to circuit complexity. You might try wire-wrap if you have considerable experience in this area, but wiring errors are difficult to dynamically trace because most signals processed by the controller are non-recurrent.

Figure 7 contains full-size, double-sided etch patterns for the disk controller and power supplies. If you choose to build your own board and cannot provide plated-through holes, then you must *not* use IC sockets. Without plated-through holes, provide conductor path connections from one side of the board to the other by inserting a wire through each hole and soldering the wire on *both* sides. Use of IC sockets precludes the possibility of soldering to the donut pads on the component side of the board. All non-component holes (over 200!) require bare wires inserted, soldered on both sides, and then clipped short. It is really much easier to use a board that has plated-through holes.

Figure 8 contains a parts layout guide for the PC boards. Be careful to orient the ICs, diodes, and capacitors as shown. If you use IC sockets, line up the dot or notch on the socket (indicating pin 1) with Fig. 8 to aid subsequent IC insertion. Use only a low-wattage



✓See List of Advertisers on Page 451

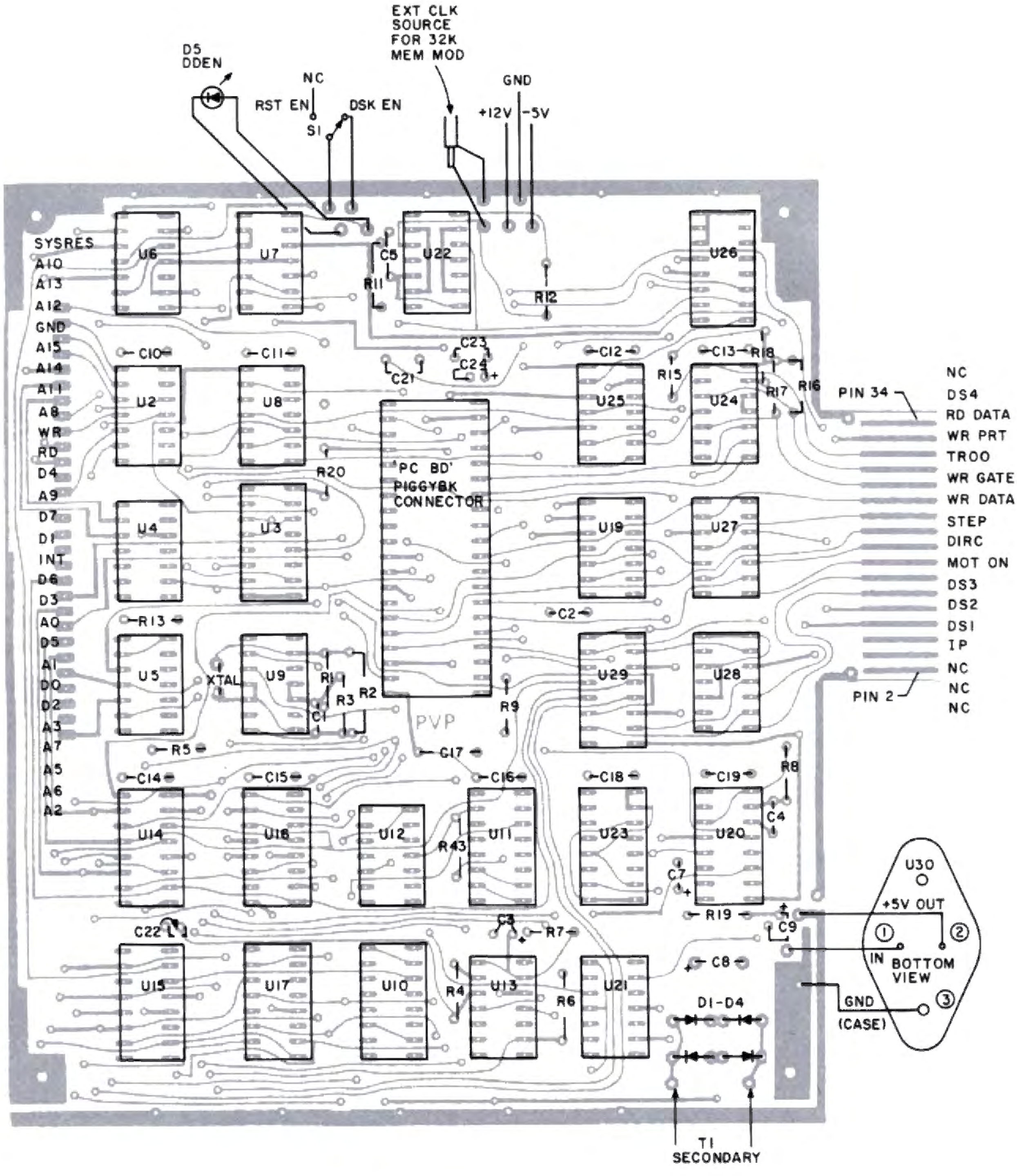



*Fig. 7a. Component side, main PC board*

iron (25 watts) with a fine tip, and .030 inch solder.

The piggyback PC board connects to the main PC board via two 20-pin strips arranged in a 40-pin-IC-style pattern. Insert and solder the shorter side of the pin strips at the second and fourth 20-pad rows from the left (see Fig. 8c) on the piggyback board. Insert the pins through the board from the non-component side, allowing about 1/16 inch of the pins to protrude above the component side; then solder in place.

When all components except U1 and U31 are installed on the PC boards, take a few moments to carefully inspect your work. Use a magnifying glass and good lighting to check all solder connections, component placement, polarities, and so on. A few extra moments is good insurance against trouble later on. When satisfied with your work, plug the piggyback board into the 40-pin socket on the main PC board, being careful to observe proper pin 1 orientation. Apply firm even force to avoid bending the pins.

Three power supply voltages are required for this project: +5 volts, +12 volts, and −5 volts. The +5 volt supply is on the main PC board, except its power transformer (T1), fuse and voltage regulator IC (U30). Since the case of U30 is at ground potential it can be mounted to whatever metal cabinet you choose to house the controller, eliminating the cost of a separate heat sink. If you mount U30 to the cabinet, be sure to remove any paint from the regulator mounting surface, and use heat sink compound to optimize heat transfer.

The +12 and −5 supply voltages are from a separate PC board (see Fig. 8d). (This is the same PC board used in the 32K Memory Mod project.) A few components are deleted from the 32K design. If you have already built the 32K Memory Mod, use its +12 and −5 volt outputs to power this project; there is more than enough current reserve for those voltages. However, the +5 volt supply on the 32K mod version is *not* adequate to operate the disk controller, and should not be used for that purpose. (This is why the +5 volt components are not on this power supply board, and a +5 supply is included on the main PC board.) Mount R24 and R25 on the +12/−5 volt supply about 1/4 inch above the PC board to aid heat dissipation. Three wires (+12, −5 and ground) provide cabling from the +12 and −5 volt supply outputs shown on Fig. 8d to the appropriate inputs on the main PC board as shown on Fig. 8a. U38 may be heatsunk to the cabinet in the same manner as U30, since its mounting tab is at ground potential.

The cabinet chosen for this project should be metal, preferably aluminum for ease of cutting and drilling. The cabinet in Table 5 is all aluminum, large enough to accommodate the disk controller and the 32K Memory Mod and printer Interface projects presented previously. Figure 9 illustrates the proper orientation and connection of the TRS-80 interface ribbon cable to the controller PC board. It also shows the proper orientation of the disk drive cable connector with respect to the output card edge. If the board is mounted in the cabinet as shown (with the component side facing down) the disk drive connector will be oriented in the traditional manner; that is, the ribbon cable will exit the bottom of the connector.

If you use the ribbon cable and connector listed in Table 5 first slip the cable through the slot in the connector rear. Then, while holding the cable straight, squeeze the connector closed using an ordinary vise.

Finally you must add 10 termination resistors at D0–D7, WR, and RD. Resistors R33–R42 (shown in Fig. 5a) may be mounted on the underside of the main PC board (see Fig. 8b). If you intend to combine the disk controller with one or both of the previous projects, you may mount the resistors on the ribbon cable PC board to be presented in the next article. Even if you do not intend to add the other two projects at this time, you may prefer to include the cable PC board to allow future expansion and ease addition of the termination resistors.

If you opt for adding resistors to the disk controller board, refer to Fig. 8b. Mount the resistors flat against the PC board, with one lead of each resistor soldered to an appropriate IC pad. The other leads of the resistors are soldered together and then to any convenient location on the ground bus running along the outer edge of the board. Use insulated wire to prevent inadvertent shorting to the PC board.

### Installation and Operation

Before inserting U1 and U31 it is wise to perform the following test. With all other ICs installed and double-checked for proper orientation, apply power to the circuit. The LED (D6) should illuminate. If it does not, disconnect power immediately and find the problem before reapplying power. A voltmeter/ohm-

| TRS-80 Address Code | Function |
|---|---|
| 37E0 WR | Supplies CLK sig to activate disk motor in selected drive |
| 37E0 RD | Clks 40Hz Timing and INTRQ sigs to TRS-80, then resets INT |
| 37EC WR | Writes TRS-80 data into FDC Command Register |
| 37EC RD | Resets INTRQ and accesses FDC Status Register to TRS-80 |
| 37ED WR/RD | Accesses FDC Track Register to TRS-80 |
| 37EE WR/RD | Accesses FDC Sector Register to TRS-80 |
| 37EF WR/RD | Accesses FDC Data Register to TRS-80 |

*Table 1. Addresses used for disk operations*



✓See List of Advertisers on Page 451

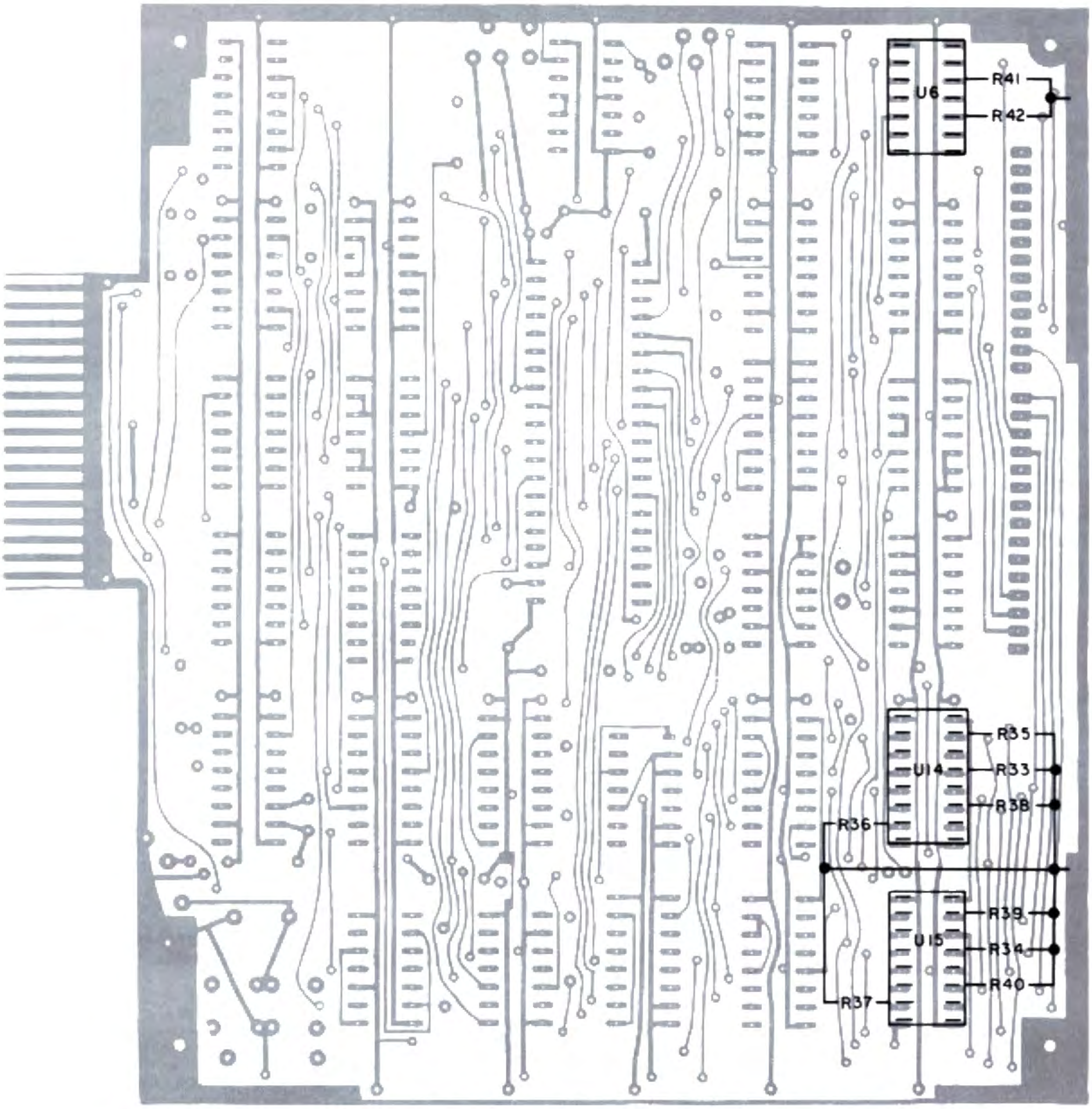
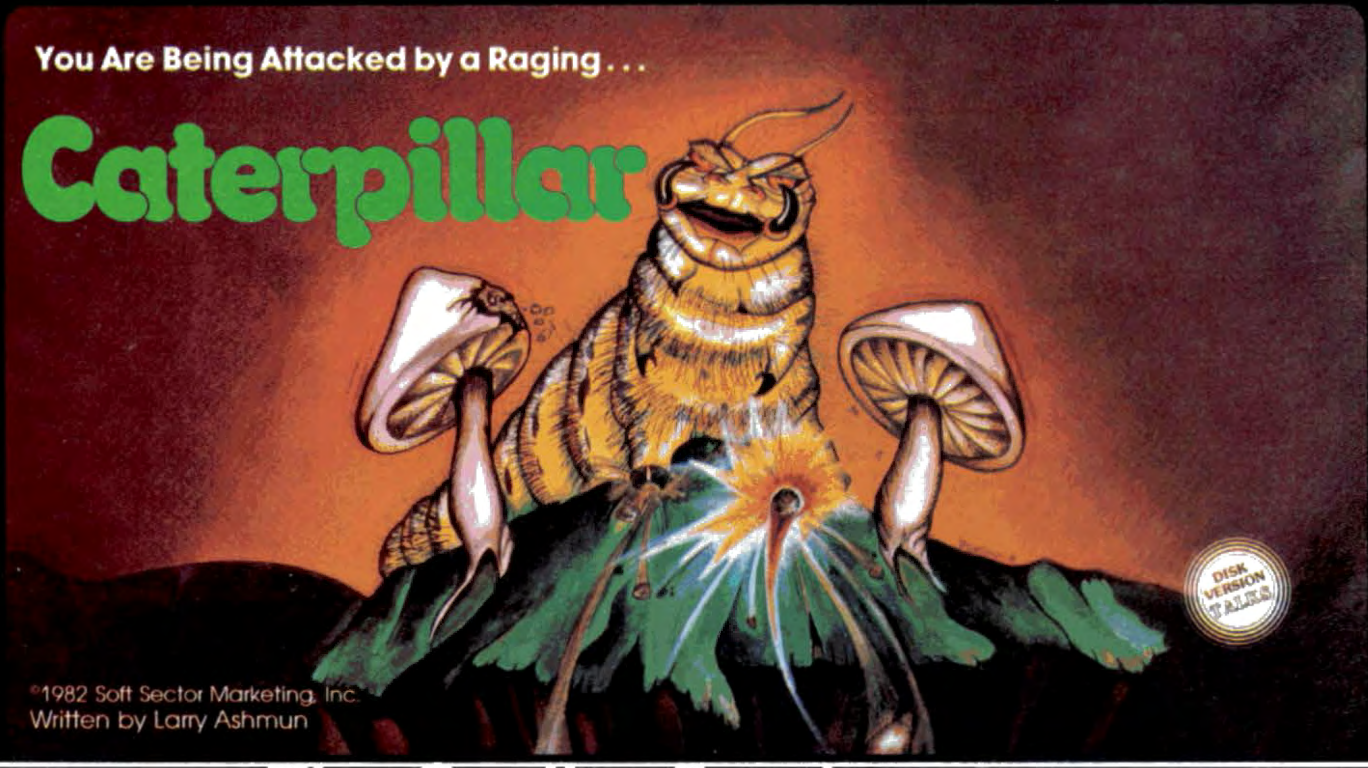



*Fig. 7b. Reverse side, main PC board*

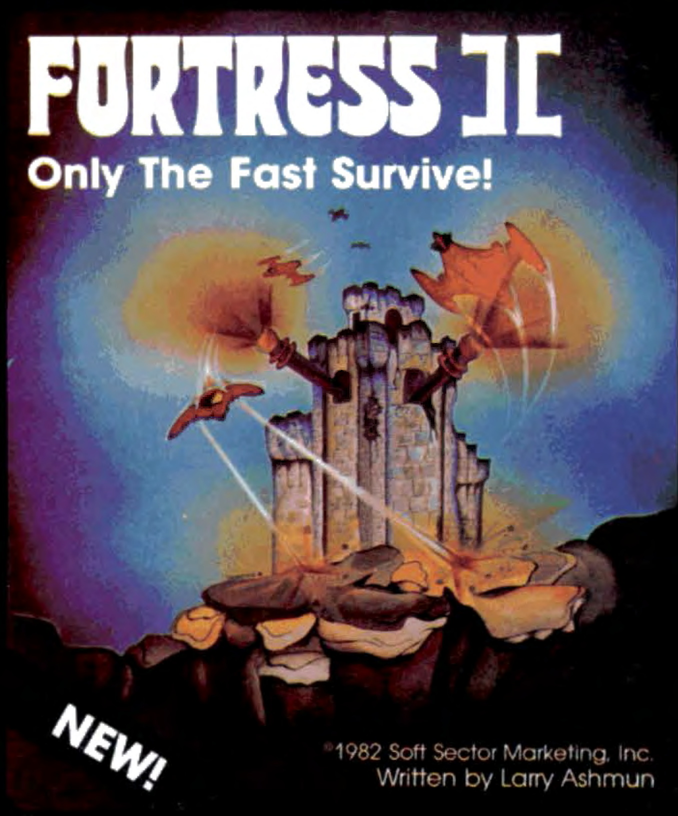


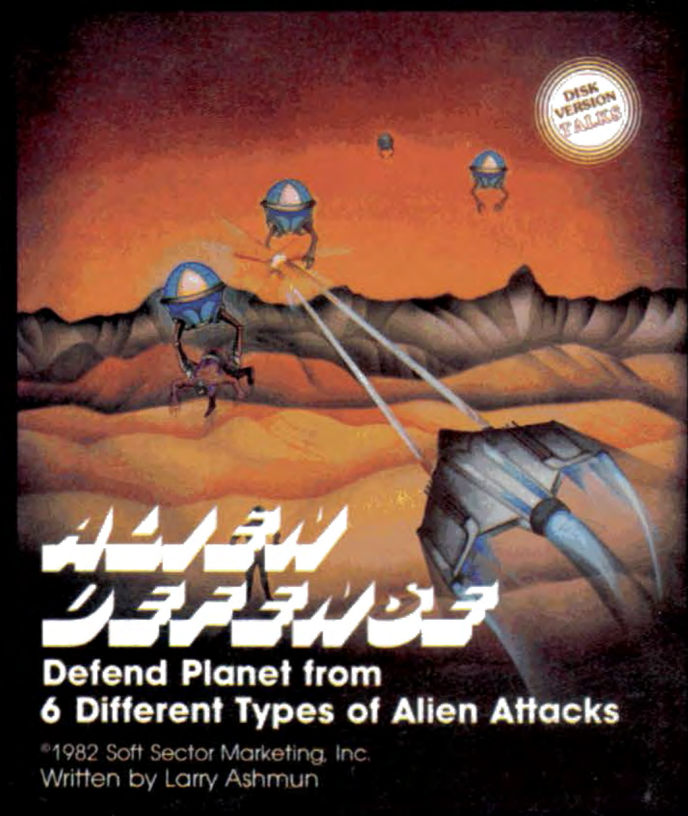


Circle 84 on Reader Service card →


You Are Being Attacked by a Raging . . .
Caterpillar
DISK VERSION TALKS
©1982 Soft Sector Marketing, Inc.
Written by Larry Ashmun




FORTRESS II
Only The Fast Survive!
NEW!
©1982 Soft Sector Marketing, Inc.
Written by Larry Ashmun


DISK VERSION TALKS
ALIEN DEFENSE
Defend Planet from
6 Different Types of Alien Attacks
©1982 Soft Sector Marketing, Inc.
Written by Larry Ashmun

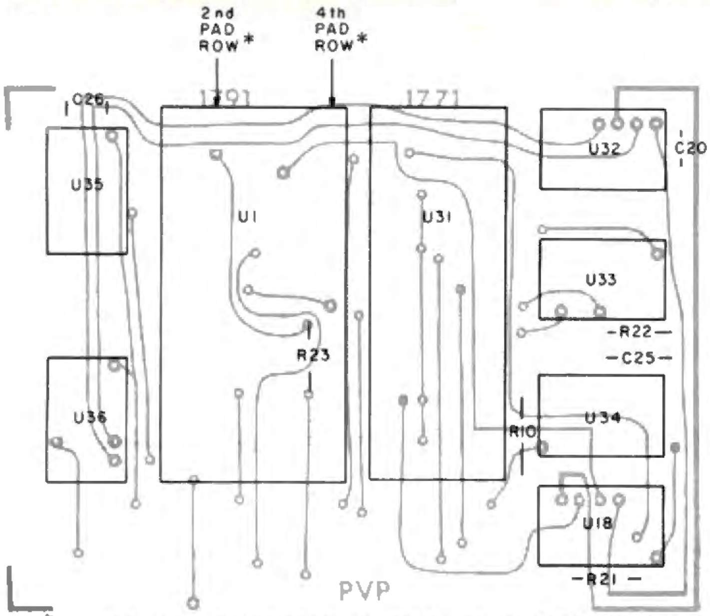



*Fig. 7c. Component side, piggyback PC board*

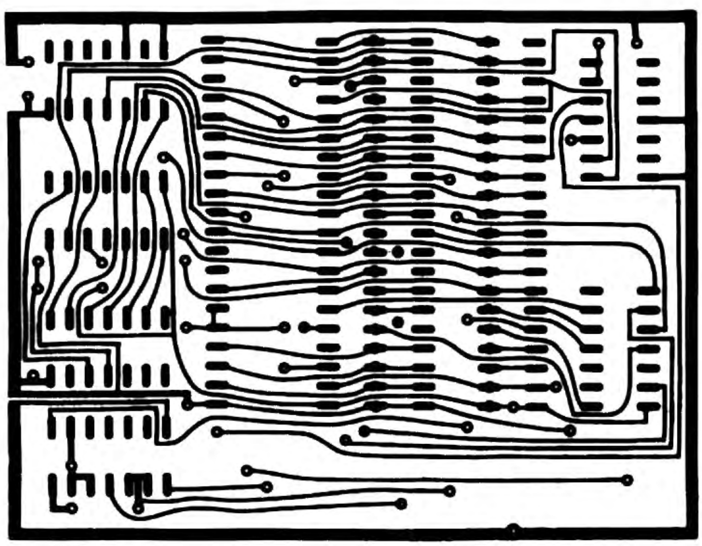


*Fig. 7d. Reverse side, piggyback PC board*

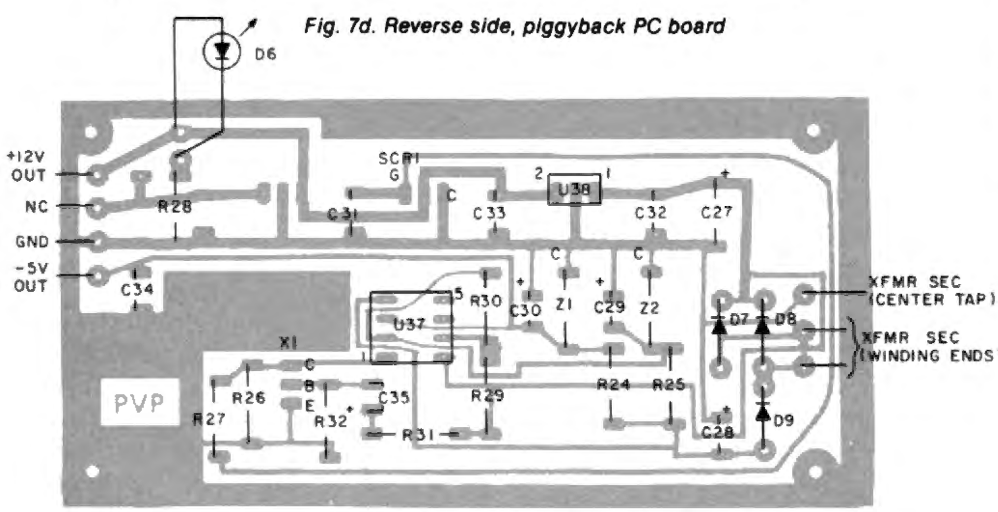



*Fig. 7e. + 12/ – 5 V power supply*

meter is handy for this type of troubleshooting. Assuming the LED illuminates, leave the power on for 30 minutes and then touch the power transformers to see that they are moderately warm, not hot. If you have access to a voltmeter, check voltages at U31 socket pins 1, 21 and 40 for – 5, + 5 and + 12 volts respectively. Then shut off the power and install U1 and U31. (Never install or remove ICs while power is applied). Repeat the power supply temperature test. If all is well, connect the disk controller to the TRS-80 and your disk drives. To keep things simple, it is best to test the system in single density mode (be sure to have S1 in the Disk Enable position). Do not use master disks until you are certain all functions are operating correctly.

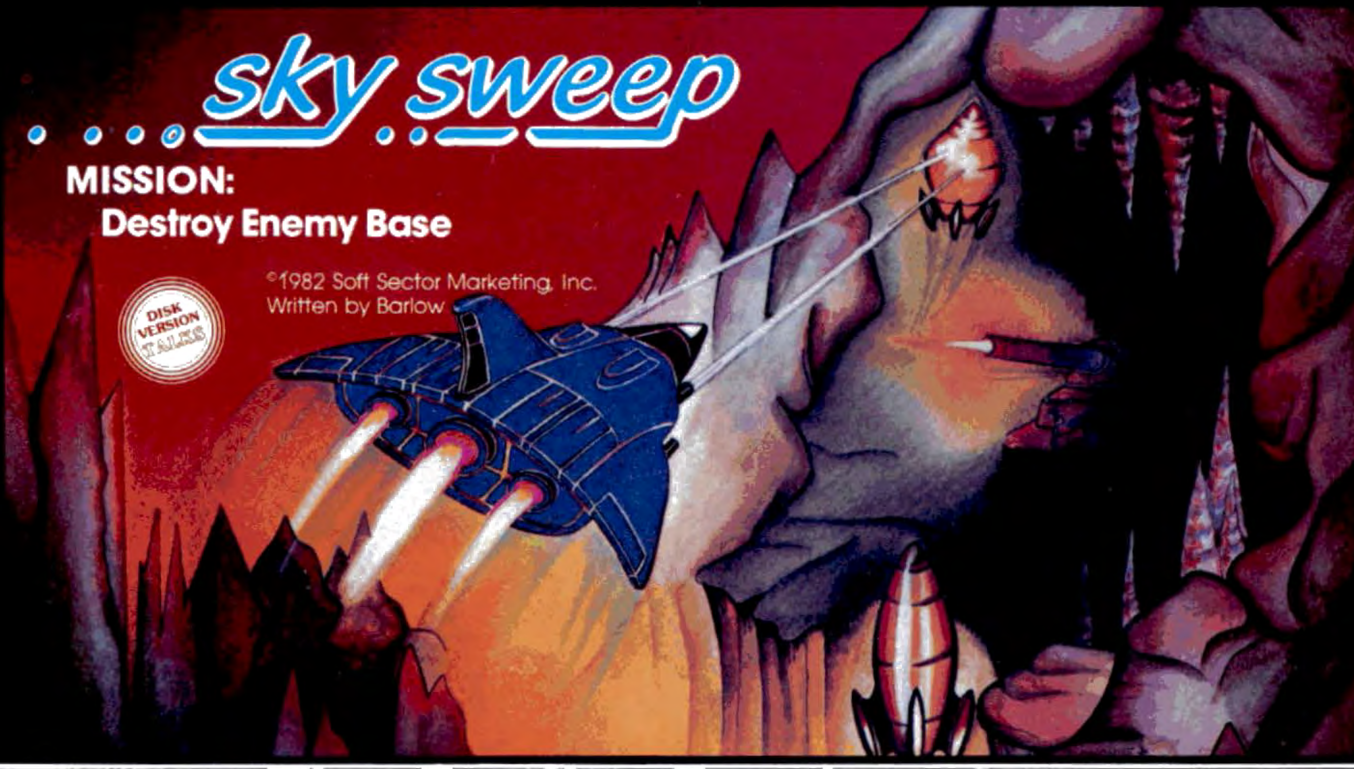
After everything is operating properly, proceed to the double density checkout using your double density DOS. The double density indicator LED can be used as an educational tool as well as a software troubleshooting aid when operating in double density mode. The lighting sequence provides insight into the way double density is implemented on the TRS-80.

For example, what occurs during a double density system-backup operation? Assuming that the system disk boots properly, the LED is *on* prior to the backup routine. When the appropriate backup command is entered, the program first formats the blank disk. The disk head/carriage assembly moves toward track zero to begin formatting and then the LED goes *off* briefly. We are trying to reproduce a double density program on what will be a double-density formatted disk—why does this happen? By default, the TRS-80 system must go to single density on powerup and Reset modes to preserve compatibility with all existing single density software. The command telling the system to go to double density must, itself, be written in single density! At powerup and Reset (via ROM) the disk drive reads track zero; track zero must be formatted for single density operation. Watch both the LED and the video monitor track indication. The LED goes back *on* at track one. Similarly, when formatting is complete and the write/read sequence begins, the LED again shuts off at track zero.

When S1 is switched to the Reset Enable position, the disk drives are disconnected as far as the TRS-80 is concerned, permitting normal operation of the Reset button. S1 must be in the Disk Enable position when the disk drives are to be used.



Circle 85 on Reader Service card →




## The Quality Continues . . .

**SSM SOFT SECTOR MARKETING, INCORPORATED** ~85

**P.O. Box 340 • Garden City, MI 48135 • 800-521-6504/(313) 425-4020**

Prices per Game: TRS-80* 16K Level II Mod I/Mod III Cassette ....... $15.95
TRS-80* 32K Level II Mod I/Mod III Diskette ........ $19.95

10% discount for 2 items, 15% for 3 or more. Please add $2.50 per order for postage & handling. Michigan residents add 4% sales tax. Outside USA (except Canada) please add $10.00 per order for postage & handing.

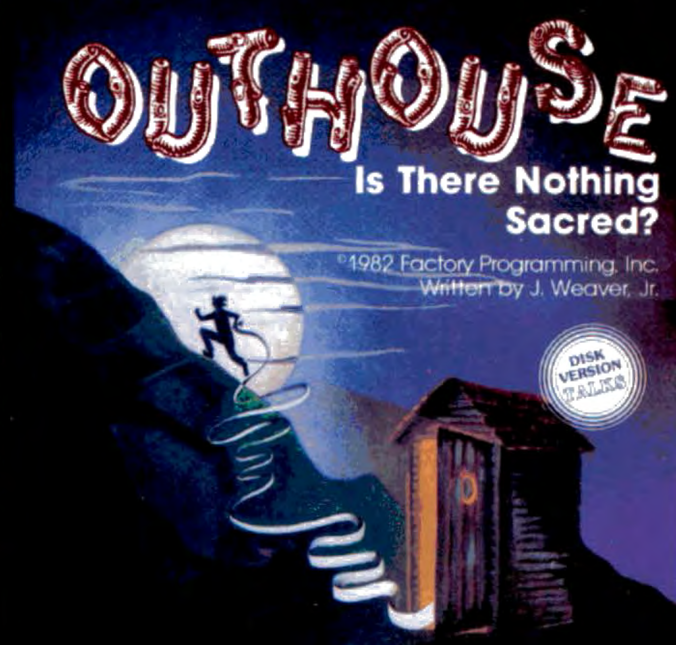
Talking and sound effects are playable through the cassette AUX plug. High scores are automatically saved after each game on disk version. Joystick compatible.

Call or write for our complete catalog.

MasterCard VISA American Express


**Don't Chicken Out!**

©1982 Factory Programming, Inc.
Written by J. Weaver, Jr.



| Address | Binary Code | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 37EC | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 37ED | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 37EE | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 37EF | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

*Table 2. FDC register access codes*

| Signal | Source |
|---|---|
| (Parallel) data | TRS-80 |
| Address bus codes | |
| Read | |
| Write | |
| System Reset | |
| Read data | Disk drive(s) |
| Track zero | |
| Index pulse | |
| Write protect | |
| Interrupt request | FDC and support circuitry |
| Interrupt | |
| Register outputs | |
| Step | |
| Direction | |
| Write gate | |
| Write data | |
| VFO enable | |
| Early | |
| Late | |
| Head load timing | |
| Ready | |
| Read clock | |

*Table 3. Data/control signal summary*

| IC Power and Ground Connections | | | | Additional IC Connections | | |
|---|---|---|---|---|---|---|
| IC | Type | +5V | GND | IC | Pin | Destination |
| U1* | FD1791-02 | 21 | 20 | U4 | 5 | GND |
| U2 | 74LS30 | 14 | 7 | U4 | 6 | GND |
| U3 | 74LS85 | 16 | 8 | U4 | 11 | GND |
| U4 | 74LS02 | 14 | 7 | U4 | 12 | GND |
| U5 | 74LS86 | 14 | 7 | U5 | 9 | +5 |
| U6 | 74LS04 | 14 | 7 | U5 | 10 | +5 |
| U7 | 74LS10 | 14 | 7 | U5 | 12 | +5 |
| U8 | 74LS20 | 14 | 7 | U5 | 13 | +5 |
| U9 | 74LS00 | 14 | 7 | U15 | 12 | GND |
| U10 | CD4518 | 16 | 8 | U16 | 12 | GND |
| U11 | CD4518 | 16 | 8 | U16 | 14 | GND |
| U12 | 74LS90 | 5 | 10 | U16 | 15 | GND |
| U13 | 74LS74 | 14 | 7 | U17 | 12 | GND |
| U14 | 74LS367 | 16 | 8 | U17 | 14 | GND |
| U15 | 74LS367 | 16 | 8 | U17 | 15 | GND |
| U16 | 74LS367 | 16 | 8 | U19 | 9 | +5 |
| U17 | 74LS367 | 16 | 8 | U19 | 10 | +5 |
| U18 | 74LS00 | 14 | 7 | U19 | 12 | +5 |
| U19 | 74LS00 | 14 | 7 | U19 | 13 | +5 |
| U20 | 74LS123 | 16 | 8 | U21 | 1 | GND |
| U21 | 74LS74 | 14 | 7 | U25 | 11 | GND |
| U22 | 74LS86 | 14 | 7 | U25 | 13 | GND |
| U23 | 74LS193 | 16 | 8 | U28 | 1 | +5 |
| U24 | 74LS14 | 14 | 7 | U28 | 3 | +5 |
| U25 | 74LS04 | 14 | 7 | U28 | 5 | +5 |
| U26 | 74LS195 | 16 | 8 | U32 | 5 | GND |
| U27 | 7416 | 14 | 7 | U35 | 4 | GND |
| U28 | 7416 | 14 | 7 | | | |
| U29 | 74LS175 | 16 | 8 | | | |
| U30 | LM309K | (see Fig. 8a) | | | | |
| U31** | 1771-1 | 21 | 20 | | | |
| U32 | 74LS04 | 14 | 7 | | | |
| U33 | 74LS04 | 14 | 7 | | | |
| U34 | 74LS30 | 14 | 7 | | | |
| U35 | 74LS74 | 14 | 7 | | | |
| U36 | 74LS32 | 14 | 7 | | | |
| U37 | LM311N | (see Fig. 6b) | | | | |
| U38 | LM340T-12 | (see Fig. 6b) | | | | |

*Pin 40: +12 V
**Pin 1: −5 V, Pin 40: +12 V

*Table 4. IC power, ground and additional connections*

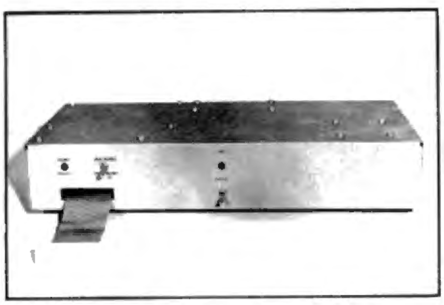


*Photo 1. Disk controller cabinet*

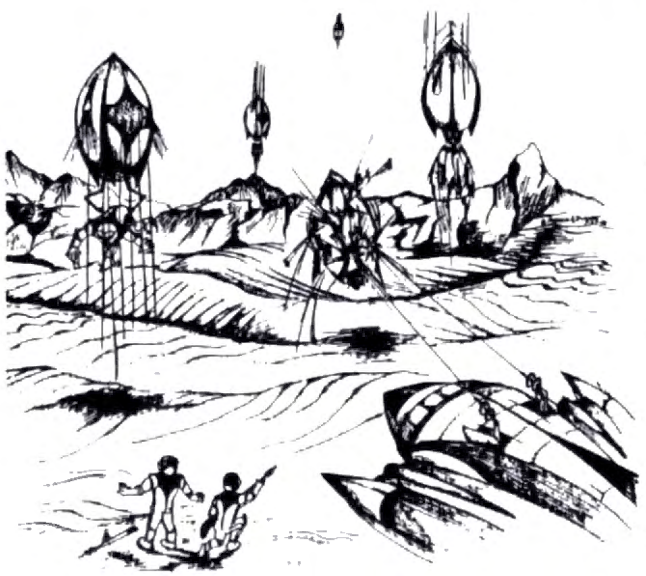
Several good double density DOS programs are available. Percom's DBLDOS is a double density system operationally very similar to Radio Shack's TRSDOS. DBLDOS routines convert files from single to double density and vice versa. A more sophisticated DOS, NEWDOS80, is now available in a double density version. Other double density operating systems compatible with the hardware supported in these two systems should also work, but I have not used them. The disk controller can be used with any single density DOS compatible with Radio Shack's Expansion Interface.

> *"Percom's DBLDOS is a double density system operationally very similar to Radio Shack's TRSDOS."*

Any double-density-rated Model I-compatible disk drives (such as Tandon's TM100-1) can be used with this system. Shugart SA400 disk drives (these are the Radio Shack-provided Model I drives), probably operate properly in double density; but they are officially rated only for single density. Shugart

# NEW BOOKS

## NEW from Larry Ashmun!



His secrets are exposed in this new, super book.

## ALIEN DEFENSE

The book contains:

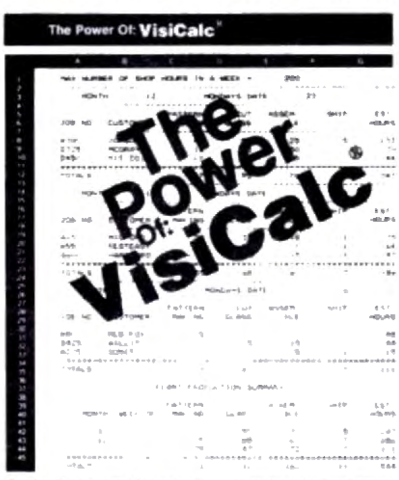
1. COMPLETE SOURCE CODE LISTING of Alien Defense (Mod III tape version).
2. Details of one approach to game writing.
3. Explains some of the more complicated routines.
4. You are informed of the concept behind the structure of the routines.

**Never Before Has Anyone Sold a Book Like This!**

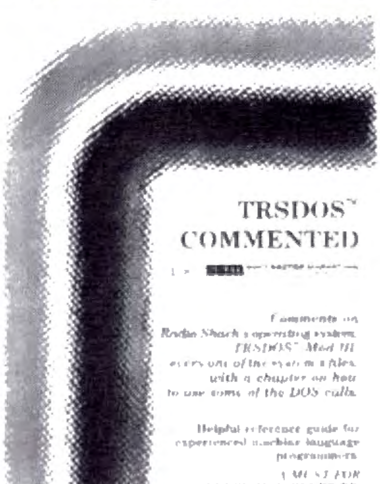
If you write in Machine language; if you have thought of writing a Machine language game; if you ever just wondered what a Machine language program looked like in its uncompiled form – This is Your Book!

Regularly $29.95 – INTRODUCTORY OFFER **$24.95**
Save $5.00 – Order Today
Offer Good thru Dec. 31, 1982

## Other Fine Books Sold by Soft Sector Marketing, Inc.




**$9.95**




**$29.95**

**$22.50**

**– DEALER INQUIRIES WELCOME –**

✓86

**SSM SOFT SECTOR MARKETING, INCORPORATED**
P.O. Box 340 • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions 313-425-4020

VISA · MasterCard · American Express

**PAYMENT** – payment accepted by charge, personal check or C.O.D. only, under the following conditions: **Charges** processed when shipped, usually within 48 hours. **Personal Checks** delay shipping, pending 3 weeks to clear. **C.O.D.** orders are certified check or cash only, add $1.50. MI residents must add 4% sales tax.
**SHIPPING & HANDLING – Shipping Charges:** Send the larger amount, 2% or $2.50, unless stipulated otherwise. Any order received without shipping and handling will be shipped **freight collect. Air Mail Shipping** outside of North America, please send the larger amount, 10% or $10.00. Overpayment will be refunded.

✓ See List of Advertisers on Page 451

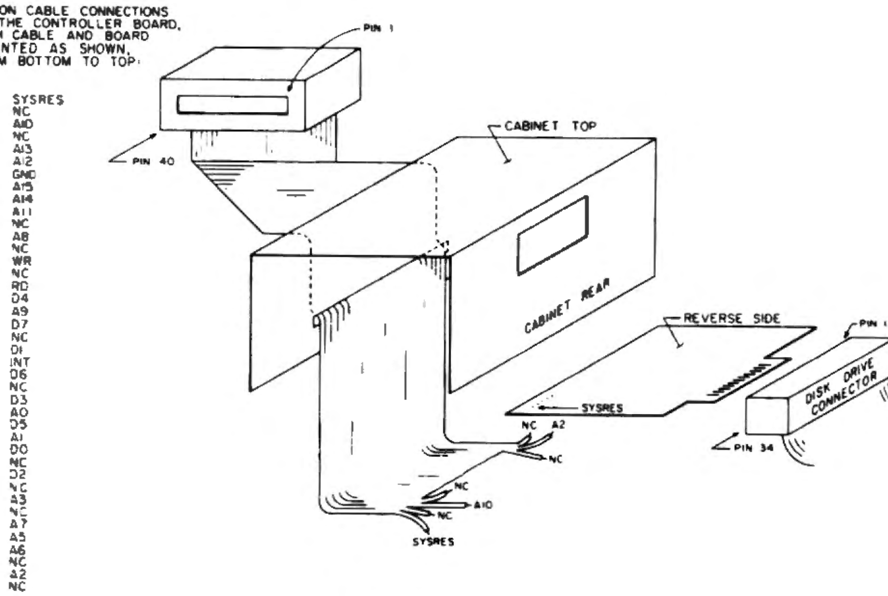



*Fig. 9. Ribbon Cable*

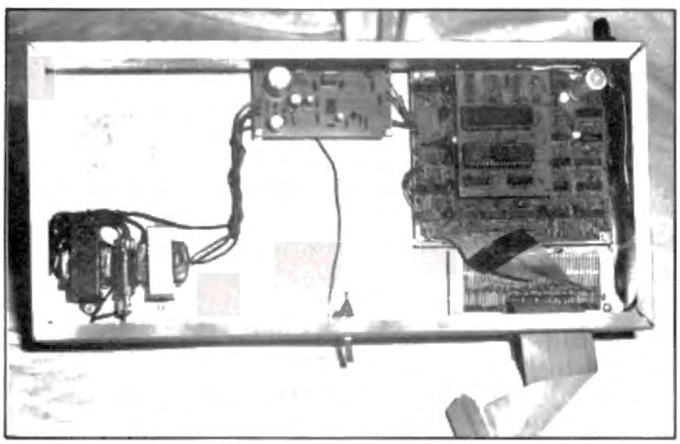


*Photo 2. Internal view of disk controller*

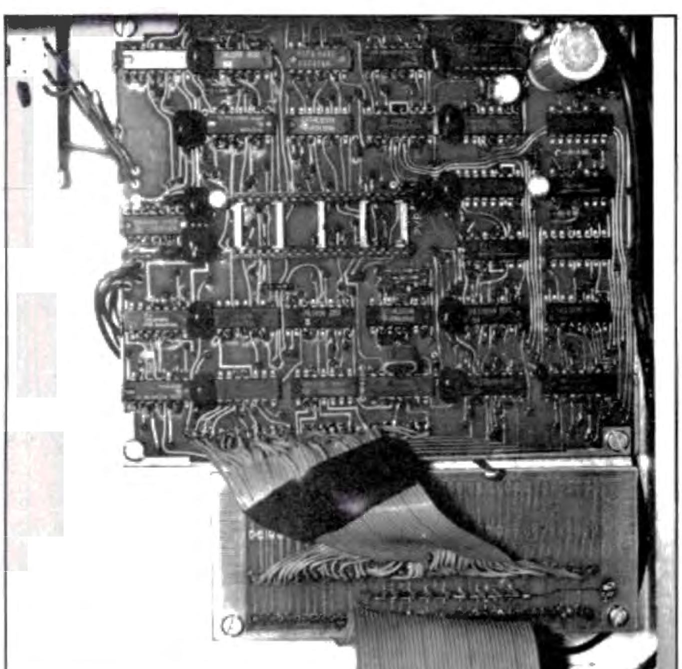


*Photo 3. Detail of disk controller and ribbon cable PC boards*

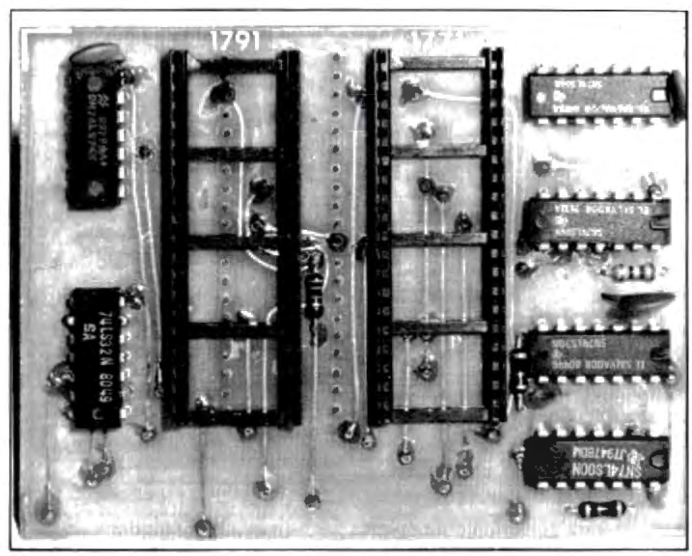



*Photo 4. Disk controller. Component side of piggyback PC board*

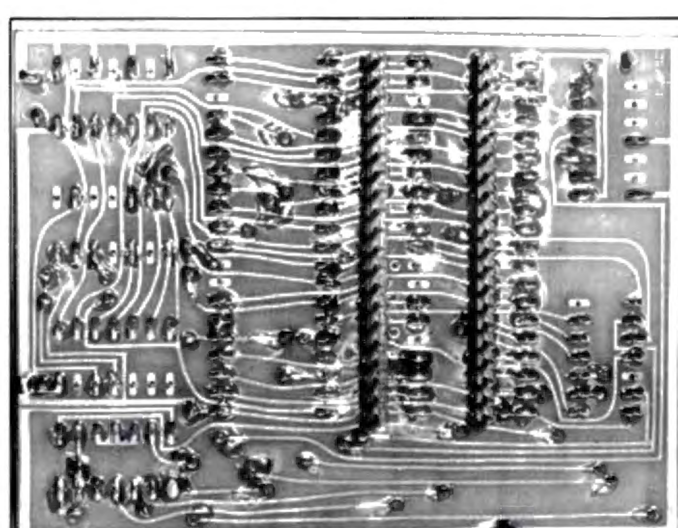


*Photo 5. Disk controller. Reverse side of piggyback PC board*

NOW!

## The Ultimate in Mailing Packages

# POSTMAN MASS MAILING SYSTEM

©1982 Alger Software, Distributed exclusively by Soft Sector Marketing, Inc.

For the serious businessman who has as little as 100 name mailing list or 200,000 names, THERE IS ONLY ONE SYSTEM FOR YOU!

### FEATURES OF THE NEW POSTMAN MASS MAILING SYSTEM

The Postman system (version 2) is an almost COMPLETE rewrite, rethink, redesign of the original POSTMAN. The many features of the new POSTMAN system are quickly outlined below.

**MULTI-DRIVE** – True multi-drive operation is possible. POSTMAN will search all drives for address files and connect them together into one large file for the duration of that session. Once POSTMAN has found the data files on the disks, the operator "sees" just ONE CONTIGUOUS sorted list of addresses. The operator does not need to tell POSTMAN when to "switch" drives or manually "swap" sections of the data file in and out of the computer's memory. This is the foremost among the list of features because of its relative uniqueness among mail list handlers written for the TRS-80.

**LARGE LIST SUPPORT** – The multi-drive operation allows the user to access data files on ALL configured drives CONCURRENTLY (at the SAME time) for truly large mailing lists. Files need not be sectioned into smaller "byte size" chunks to fit into memory.

**HARD DISK SUPPORT** – (HARD DISK POSTMAN only) The FULL utilization of the space and speed of the new hard disk drives is possible with POSTMAN. For example, a 7.5 megabyte drive can be configured to hold almost 60,000 labels. Multiple hard drives can be accessed CONCURRENTLY allowing 200,000++ entry mailing lists.

**FORM LETTER CAPABILITY** – With the purchase of the separate POSTRITE program, the user is provided with an easy to use form letter generator which will merge a generalized letter produced from a word processing system (i.e. LAZY WRITER, etc.), with the name and address information from the POSTMAN MASS MAILER data base. POSTWRITER allows the user to insert any field from a POSTMAN label entry anywhere in the letter.

**MENU OPERATION** – As you would in a restaurant, choose your dinner from a list (or MENU). POSTMAN will allow you to direct its actions by selecting from various menus that it will display. A complete discussion of each menu is presented in the manual.

**INSERT** – New names can be quickly added to your list at any time. The new addresses are placed into the file in their proper sorted order eliminating the need for a separate sort operation after entering a stack of new names. POSTMAN will allow the operator to enter a "batch" of labels without returning to the control menu between each label insertion, thus speeding entry and reducing the aggravation of extra menu control keystrokes.

**DELETE** – Names can be removed at any time when they are no longer needed.

**EDIT** – Information in any name entry can be quickly changed at will with "word processor" ease. A "transparent" cursor simply is moved to the label displayed on the computer screen and corrections are just typed over the existing label. If you happen to change a field which is also used as a sort key, POSTMAN will automatically move the changed label to its correct position in the list to maintain the sorted arrangement of the labels.

**OVERLAY** – When identical changes are needed on many addresses, the OVERLAY feature can make them with one keystroke. The needed changes which are common to many labels are entered into the "overlay mask". When you wish to apply these common changes to any label, one command will do it.

**SORT** – Arrange your list in any alphabetic or numeric order. The ordering may use one or more fields to control the sort. A machine language heap sort assures fast execution. The sort need only be performed once, the sorted list will stay sorted through all subsequent insertions, deletions, and changes to existing labels. NO NEED to leave the POSTMAN program to use a separate program to sort your data. Your data is sorted quickly and after sort completion, POSTMAN is ready for your next command!

**SPECIAL STREET ADDRESS SORT** – For the user with many addresses on the same street, POSTMAN will sort your entries by the house NUMBER after grouping those on the same street together. Local city lists can be quickly sorted to aid post office dispatching.

**PURGE** – Unwanted duplicate addresses can be removed from your list automatically or under operator control.

**SEARCH** – Any address in your list can be quickly found with fast search and positioning commands. Three different types of searches are provided. A "fast" search which uses a hashing technique, a "selective sequential" search for labels with common fields, and "quick" positioning using the first or major sort field to get you into the general "ball park" of a label or sequence of labels.

**LABEL PRINTING** – One, a few or all addresses in your list can be printed on standard or non-standard label stock. Up to 6 labels across can be printed with a format YOU can easily control. TWO user definable 'ATTN' lines are provided for any use. Labels can be printed from many of POSTMAN's menus, search, edit, or during label insertion.

**EFFICIENCY** – POSTMAN is written in the machine's native language to gain the full advantage of the microcomputer's speed. Extensive use of program segmentation reduces the amount of use RAM needed to hold the program, allowing a greater number labels to be kept in core, resulting in faster operation. Little used routines need only be brought into memory when they are needed and once through with their task, release their space back to POSTMAN.

**REPORT LISTINGS** – A special program to produce columnar listings of address data from your label data base is provided. You can easily specify the information to be printed.

**DATA DISK MERGING** – Labels can be quickly transferred from one disk to another with the PSTMERGE program callable from the main POSTMAN SYSTEM menu. Source and destination drives needed not be separate drives, prompts to exchange diskettes if the same drive is used, are provided.

**DATA DISK PREPARATION UTILITY** – Provided with POSTMAN is the DPREP program which allows the user to prepare a floppy/hard disk for use with POSTMAN. This easy to use utility can be told to prepare any portion of the available space on a disk.

**DATA INTEGRITY** – All data transfers to the disk files are made using special write commands which instructs the operating system to check the validity of EACH write to the disk.

**DATA GUARD*** – Is a special programming technique only offered by Soft Sector Marketing, Inc. If by chance your machine resets while writing information to the disk, you only lose the information that you were writing. Your files are always protected from the danger of losing all the work that you have put in that day. NO OTHER PROGRAM ON THE MARKET OFFERS THIS PROTECTION. If you reset with ANYBODY'S MAILING PACKAGE DURING WRITING you would destroy your ENTIRE data disk. We can't stop your machine from failing but we can protect your data.

### Description of Label Record Fields:

| Length | Name | Description | Length | Name | Description |
|---|---|---|---|---|---|
| 10 | Code | User defined printable field | 15 | City | City, township, village |
| 15 | Last Name | Last name of addressee | 5 | State | State, province, territory |
| 15 | First Name | First name of addressee | 9 | Zip | Zip code, zone, route |
| 26 | Company | Name of company | 2 | Data 1 | User definable field |
| 26 | Address | Street address | 5 | Data 2 | User definable field |

### IDEAL SYSTEM

Mod III 48K 1-40 Track Drive • 2-80 Track Dual Headed Drives • Dosplus or LDOS Operating Systems

Gives space for over 11,000 names - 5 second average name insertion - time sorts all 11,000 names in less than 4 minutes

*Special version to work on Dosplus 4.0 Hard Disk operating system.

– Overview Available –

The POSTMAN system requires Mod I or Mod III, 48K, 2 disk drives minimum.

Standard Version<br>Only **$125.00**

Standard Version with<br>POSTWRITER form letter writer<br>**$175.00**

For DOSPLUS Hard Disk 4.0 Operating System<br>& Radio Shack's Hard Drive System<br>**$225.00**

✓87

**– DEALER INQUIRIES WELCOME –**

**SSM SOFT SECTOR MARKETING, INCORPORATED**

P.O. Box 340 • Garden City, Michigan 48135

Order Line **800-521-6504**

Michigan Orders & Questions 313-425-4020

VISA MasterCard

PAYMENT -

Checks

Charges<br>Personal<br>C.O.D.

SHIPPING & HANDLING Shipping Charges

freight

collect Air Mail Shipping


✓See List of Advertisers on Page 451

**Parts required for disk controller including +5 volt power supply:**
**(Designator C6 is not used.)**

| | |
|---|---|
| U1 | FD1791-02 IC |
| U2,U34 | 74LS30 IC |
| U3 | 74LS85 IC |
| U4 | 74LS02 IC |
| U5,U22 | 74LS86 IC |
| U6,U25,U32,U33 | 74LS04 IC |
| U7 | 74LS10 IC |
| U8 | 74LS20 IC |
| U9,U18,U19 | 74LS00 IC |
| U10,U11 | CD4518BE IC |
| U12 | 74LS90 IC |
| U13,U21,U35 | 74LS74 IC |
| U14,U15,U16,U17 | 74LS368 IC |
| U20 | 74LS123 IC |
| U23 | 74LS193 IC |
| U24 | 74LS14 IC |
| U26 | 74LS195 IC |
| U27,U28 | 7416 IC |
| U29 | 74LS175 IC |
| U30 | LM309K IC |
| U31 | FD1771-01 or INS 1771-1 |
| U36 | 74LS32 IC |
| R1,R7,R33–R40,R42 | 1,000 ohm, ¼ W, 5%, resistor |
| R2,R3,R12,R43 | 3,300 ohm, ¼ W, 5%, resistor |
| R4,R6 | 4,700 ohm, ¼ W, 5%, resistor |
| R5,R8,R9,R10,R14,R20,R21 | 10,000 ohm, ¼ W, 5%, resistor |
| R11 | 8,200 ohm, ¼ W, 5%, resistor |
| R13 | 1,500 ohm, ¼ W, 5%, resistor |
| R15,R16,R17,R18 | 150 ohm, ¼ W, 5%, resistor |
| R19 | 68,000 ohm, ¼ W, 5%, resistor |
| R23,R41 | 470 ohm, ¼ W, 5%, resistor |
| R22 | 330 ohm, ¼ W, 5%, resistor |
| C1,C25 | 100 pF, ceramic disk capacitor |
| C2 | 200 pF, ceramic disk capacitor |
| C3,C7,C9 | 100 uF, 10 V, electrolytic capacitor |
| C4 | 47 pF, ceramic disk capacitor |
| C5 | 10 pF, ceramic disk capacitor |
| C8 | 4,500 uF, 10 V, electrolytic capacitor |
| C10–C19,C21,C23 | .1 uF, 50 V, mylar capacitor |
| C20C26 | .1 uF, 12 V, ceramic disk capacitor |
| C22 | .001 uF, mylar capacitor |
| C24 | 10 uF, 16 V, electrolytic capacitor |
| S1,S2 | SPST toggle switch |
| D1–D4 | 1 A, 100 PIV diode |
| D5 | LED (Digikey NSL5053 or equivalent) |
| F1 | 3/8 A, 3AG (fast acting) fuse |
| XTAL | 4 MHz crystal, HC18/U style (NDK-040S or equivalent) |
| T1 | 6.3 volt AC, 1.2A "filament" transformer (Triad F-14X or equivalent) |

Miscellaneous

PC boards, cabinet (Bud AC-412/BPA-1520 or equivalent—see text), AC line cord, two 20-pin male header strips (AP Products 929834-08 or equivalent), 40-conductor ribbon cable with matching "TRS-80 keyboard card-edge compatible" connector (Radio Shack 278-771/276-1558 or equivalent), IC sockets (one of the 40-pin sockets must be capable of accepting .025 inch square posts—JDR Microdevices "40PIN-ST" is a good choice), hardware (including threaded standoffs to mount the PC boards).

**Parts required for +12/−5 volt power supply:**

| | |
|---|---|
| U37 | LM311N IC (do not substitute) |
| U38 | LM340T-12 IC |
| SCR1 | R122B (R/S 276-1067 or equivalent) SCR |
| D6 | LED (Digikey NSL5053 or equivalent) |
| Z1 | 5.1 V, 5%, 1/2 W (1N5231B) zener diode |
| Z2 | 4.3 V, 5%, 1/2 W (1N5229B) zener diode |
| C27 | 1000 uF, 25 V, electrolytic capacitor |
| C28 | 220 uF, 25 V, electrolytic capacitor |
| C29 | 220 uF, 10 V, electrolytic capacitor |
| C30 | 4.7 uF, 10 V, electrolytic capacitor |
| C31 | .001 uF, 25 V, ceramic disk capacitor |
| C32–C34 | .1 uF, 50 V, ceramic disk capacitor |
| C35 | 10 uF, 25 V, electrolytic capacitor |
| R24,R25 | 390 ohm, 1/2 W resistor |
| R26 | 2.2K ohm, 1/4 W resistor |
| R27 | 1.2K ohm, 1/4 W resistor |
| R28,R29 | 1K ohm, 1/4 W resistor |
| R30 | 100K ohm, 1/4 W resistor |
| R31,R32 | 10K ohm, 1/4 W resistor |
| D7–D9 | 1A, 100 PIV, diode (1N4002 or equivalent) |
| T2 | 24 VAC, .8A center-tapped transformer (Burnstein Applebee 18A1833-3 or equivalent) |
| F2 | 3/8A, 3AG (fast acting) fuse |
| X1 | MPS3705 transistor (general purpose, NPN) |

Miscellaneous

PC board, in-line fuseholder, hardware.

*Table 5. Parts list*

believes all units after serial number 80000 are okay, as are about 80 percent of all those below 80000). I experimented with two very early units (serial numbers well below 10,000) and achieved partial compatibility. They boot the double-density DOS and perform flawlessly most of the time. However, they occasionally fail operations which involve writing and subsequent read/verifying "worst case" data bit combinations (eg: 6DB6H).

It is best to apply power to the disk controller before turning on the keyboard. Reverse this sequence when removing power. Also, apply power to the controller before inserting disks and remove all disks before turning off power. That way, even if transient voltages conspire against you, your disks will be safe. ■

*Readers of 80 Micro are welcome to use the etch patterns contained in this article to construct the Disk Controller for their personal use only.*

*An etched and drilled epoxy-glass PC board set with plated-through holes as shown in Fig. 7 is available for $54.95. A complete kit of parts is also available, including PC boards, the 1791/1771 controllers, cabinet (no holes or cutouts, but dimensional layout is included), IC sockets, and all other parts listed, including the +12/−5 volt power supply for $219.95. The 1791/1771 controller set is available separately for $69.95. (There will be a predrilled, punched, painted, and labelled version of the cabinet available for the complete system in the final article of this series. If you would like to order the kit of parts as above, but without the blank cabinet, deduct $20.00 from the kit price.)*

*If you have already built the +12/−5 volt supply for the 32K memory mod project, you may order either the PC board set or parts kit for the disk controller without components for this supply. The PC board set without power supply board is $49.95, and the parts kit without +12/−5 volt power supply components is $204.95.*

*The above prices are postpaid in the continental U.S. Arizona residents add 4 percent sales tax. Order from PVP Industries, P.O. Box 35667, Tucson, AZ 85740.*

# COLOR COMPUTER

## MASTER CONTROL

©1981 Soft Sector Marketing, Inc.
Written by A. Schwartz

Requires 16-32K

1. 50 preprogrammed command keys. Standard and Extended command.
2. Direct control of motor, trace, and audio from keyboard.
3. Automatic line numbering.
4. Programming line numbering.
5. Direct Run Button.
6. Keyboard overlay for easy program use.
7. Easy entry of entire commands into computer.

Load Master Control into your machine then either type in a BASIC program or load one in from tape to edit. Cuts programming time by 50% or more

**$24.95**

For The Radio Shack
**Color Computer***

# COLOR BONANZA

Some 4K,
Some 16K,
Some 32K

Written by E.R.I.

**50 PROGRAMS IN ONE PACKAGE**

**$49.95**

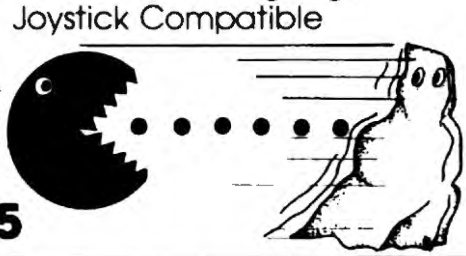
*The Color Computer is a product of Radio Shack, division of the Tandy Corp.

## GHOST GOBBLER

NEW

16K – Machine Language
Joystick Compatible

Superior to:
PACKET MAN &
PAC ATTACK



**$21.95**

## TAPE DIRECTORY

Copyright ©1982 Soft Sector Marketing, Inc.


NEW

Creates index or your programs for each tape. To screen or printer.

A MUST FOR ALL COLOR COMPUTER USERS!

Cassette . . . . . . . . . . . . . . . . . . . . . . . **$14.95**

– For 4K Color Users –

## COLOR SCARFMAN

NEW

GREAT GRAPHICS
Machine Language
Only **$19.95**

**– DEALER INQUIRIES WELCOME –**

✓88

**SSM SOFT SECTOR MARKETING, INCORPORATED**
P.O. Box 340 • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions 313-425-4020

PAYMENT - VISA MasterCard American Express Checks Charges Personal C.O.D.

SHIPPING & HANDLING - Shipping Charges collect. Air Mail Shipping freight

LOAD 80

*Add search to Level II Basic.*

# The Looking for Z$ Blues

*J. E. Hieber*
*725 Idlewild Road*
*Bel Air, MD 21014*

The lack of a search function is a serious limitation of the TRS-80 Level II cassette system. How often have you wanted to know every place a variable is used in a program? The only way is to eyeball the code line by line.

The Assembly routine in the Program Listing adds the omitted search function. The program interfaces with Level II's normal command structure; the new commands are entered from the Basic prompt.

The search is executed from the current line pointer toward the end of the Basic program in memory. If the search argument is found, the line is displayed and the current line pointer is set to that line. Continue the search by entering the search command with no argument. If the search argument is not found, the Basic Ready is displayed and the current line pointer is not changed.

Along with searching I added the ability to advance the display forward or backward one line, the ability to reset the current line pointer to the beginning of the program, and the ability to enter the Edit mode with an abbreviated command.

I thought the new commands should be meaningful, short, and handy. All commands end with the enter key. Table 1 lists the commands.

*"The length of the search argument should be limited to no more than 63 characters."*

The quotation marks call the search function. This works well with variables and characters within REM statements. Basic keywords such as GOTO, IF, and SET cannot be found. These keywords are stored as single character tokens in program memory. An alternate search command, the colon, is used to search for the keywords. The length of the search argument should be no more than 63 characters. Excessively long search arguments can cause the system to bomb.

The semicolon command advances the display one line. Press the semicolon and enter keys repeatedly to step through a program.

The minus sign steps the display one line backward. Minus can also be hit repeatedly.

The period resets the current line pointer to the first line of the program. This is handy to reset the pointer before a new search sequence.

The edit command is entered with the first letter (E) reducing the keystrokes needed to enter the edit mode. The semicolon and minus commands can be appended to the edit abbreviation to edit the next line and previous line, respectively.

The at sign (@) displays the current line.

The techniques and hooks in the program can be used to add other commands and features. ■

*John Hieber, a graduate engineer and mathematician, first programmed an IBM 650 in 1963. He has owned a micro for about two years.*

**The Key Box**

**Basic Level II**
**Model I**
**16K RAM**

✓See List of Advertisers on Page 451

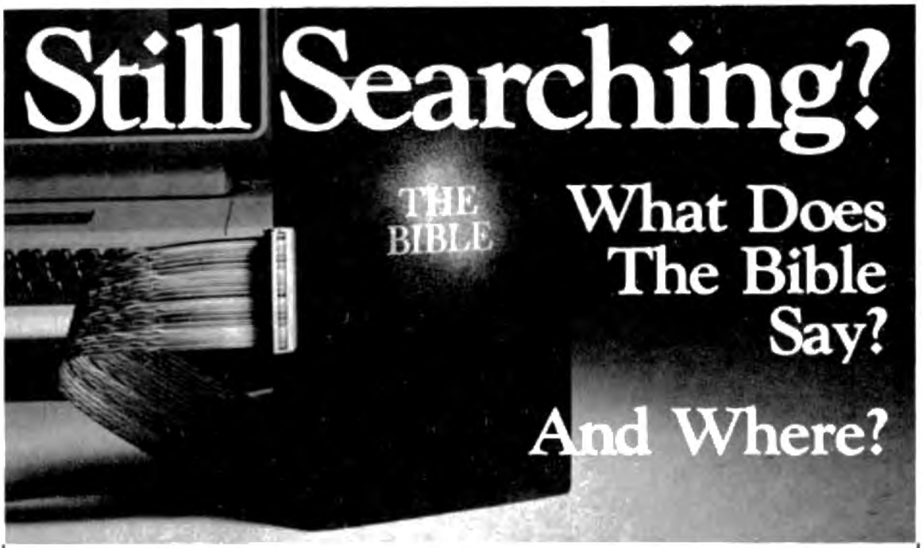
| Command | Explanation |
|---|---|
| "I1 | Search for variable I1 (argument should be < 64 characters) |
| :GOTO | Search for keyword GOTO |
| . | Reset current line pointer to first line of the program |
| – | Step display one line backward |
| @ | Display current line |
| ; | Step display one line forward |
| E | Edit current line |
| E – | Edit previous line |
| E; | Edit next line |

*Table 1. New Level II Commands*

*Program Listing*

```
                00100 ;THIS ROUTINE ADDS A SEARCH FUNCTION TO LEVE
L II, TO
                00110 ;PROVIDE THE FACILITY TO LOCATE GIVEN CHARACTER
                00120 ;STRINGS IN A BASIC PROGRAM.
                00130 ;THIS ROUTINE WILL ALSO PROVIDE SHORT NOTATI
ON FOR
                00140 ;LIST AND EDIT COMMANDS.  THE ABILITY TO SIN
GLE
                00150 ;STEP FORWARD AND BACKWARD THROUGH A PROGRAM IS
                00160 ;INCLUDED.
                00170 ;
                00180 ;       WRITTEN BY:  J.E. HIEBER
                00190 ;                    BEL AIR, MD. 21014
                00200 ;
                00210 ;       COPYRIGHT APRIL 1981
                00220 ;
                00230 ;THE FOLLOWING IS A SUMMARY OF THE COMMANDS
ADDED:
                00240 ;
                00250 ;  E; WILL EDIT NEXT LINE
                00260 ;  E  WILL EDIT THIS LINE
                00270 ;  E- WILL EDIT PREVIOUS LINE
                00280 ;   ;  WILL LIST NEXT LINE
                00290 ;   @  WILL LIST THIS LINE (AT SIGN)
                00300 ;   -  WILL LIST PREVIOUS LINE
                00310 ;   .  WILL SET THE POINTER TO START OF THE P
ROGRAM
                00320 ; "CHARAC STRING   WILL SEARCH THE BASIC PRCGRAM
                00330 ;                  FOR A MATCH
                00340 ; :CHARAC STRING   WILL SEARCH FOR BASIC KEY
WORDS
                00350 ;                  SUCH AS GOTO, IF, THEN, ETC.

                00360 ; " OR : WITH A NULL STRING WILL USE THE PRE
VIOUS
                00370 ;        STRING AND CONTINUE THE SEARCH
                00380 ;
41B2            00390 CMDHK:  EQU     41B2H    ;HOOK FROM ROM COMMAND
                00400                          ;PROCESSOR
1B2C            00410 FLNO:   EQU     1B2CH    ;FIND THIS LINE NO =
 DE
40EC            00420 TLNO:   EQU     40ECH    ;THIS LINE NO.
06CC            00430 BASCE:  EQU     06CCH    ;ROM BASIC ENTRY
41E6            00440 BUFAD:  EQU     41E6H    ;COMMAND BUFFER
40A4            00450 BASPT:  EQU     40A4H    ;BASIC PROG BUFFER
403E            00460 PRVRQ:  EQU     403EH    ;PREV DEC SIZE
4040            00470 STRLN:  EQU     4040H    ;STRING LENGTH
4041            00480 LNPTR:  EQU     4041H    ;TEMP LINE POINTER
4043            00490 STRING: EQU     4043H    ;SEARCH STRING
                00500 ;
7ECD            00510         ORG     7ECDH    ; 32461
7ECD 21E37E     00520 JEHMN2: LD      HL,START          ;SETUP HOOK
ADDR
7ED0 22B341     00530         LD      (CMDHK+1),HL
7ED3 3EC3       00540         LD      A,0C3H   ;JP CMD TO ROUTINE
7ED5 32B241     00550         LD      (CMDHK),A
7ED8 C3CC06     00560         JP      BASCE    ;JUMP TO ROM BASIC
7EDB 2AA440     00570 MOVTP:  LD      HL,(BASPT)        ;PT TO BASIC
 BUFFER
7EDE CD667F     00580         CALL    LODLA    ;LOAD LINE NO OF FIR
ST LINE
7EE1 1834       00590         JR      BLDC3    ;LIST LINE
7EE3 F5         00600 START:  PUSH    AF       ;SAVE REG
7EE4 D5         00610         PUSH    DE
7EE5 E5         00620         PUSH    HL
7EE6 C5         00630         PUSH    BC
7EE7 3E05       00632         LD      A,5      ;CHECK FOR NULL CMD
7EE9 B9         00634         CP      C        ;LINE NO WITH NO LINE
7EEA 2867       00636         JR      Z,EXIT3  ;YES RETURN
7EEC 23         00638         INC     HL       ;POINT TO NEXT CHARA
C IN CMD
7EED 7E         00640         LD      A,(HL)   ;LOAD 1ST BYTE
7EEE FE22       00650         CP      '"'      ;IS THIS A SEARCH RE
QUEST
7EF0 2802       00660         JR      Z,PSRX   ;YES
7EF2 FE3A       00670         CP      ':'      ;IS THIS A PACKED SR
CH REQ
7EF4 CA8F7F     00680 PSRX:   JP      Z,PSRCH  ;YES
7EF7 FE2E       00690         CP      '.'      ;IS THIS A REQ FOR T
OP OF BUF
7EF9 28E0       00700         JR      Z,MOVTP  ;YES
7EFB FE45       00710         CP      'E'      ;IS THIS AN EDIT REQ
UEST
```

*Listing continues*




Now you can use computer technology to search the Bible on any subject. With THE WORD processor (including a disk copy of the complete KJV Bible text) you can create indexes on keywords. Or phrases. Even on concepts.

A full spectrum of text search functions allows detailed analysis for the real student of the Bible.

Print or display selected verses and/or references for personal study or Bible classes. Build your own computerized library of research material.

**THE WORD** processor **$159.95***

Bible Research Systems • 8804 Wildridge Drive • Austin, Texas 78759 • (512) 346-2181

Requires 48K, 1 disk drive, APPLE, TRS-80, or IBM-PC** *Plus $2.50 postage/handling
**Trademarks of APPLE Computer, Inc., Tandy Corp., and IBM Corp., respectively.

## SPIKE-SPIKERS™

*Protect-Control-Organize*

Computers & sensitive electronic equipment
Helps prevent software "glitches", unexplained memory loss, and equipment damage

**PROTECTS** with transient absorber & dual five-stage filter. Absorbs power line transients & filters out RFI "hash."

**CONTROLS** with 8 individually switched 120 vac grounded outlets. Main on-off switch, fuse & indicator light.

Deluxe Power Console **$79.95**

**ORGANIZE** your computer & other equip. power cords. No more overloaded octopus cube taps.

2-Socket Plug-In Models

**MINI-I**
Transient absorber only
**$34.95**

**MINI-II**
Transient absorber plus 3-stage RFI "Hash" filter
**$44.95**

4-Socket Plug-In Model with Light

**QUAD-I**
Transient absorber only
**$49.95**

**QUAD-II**
Transient absorber plus dual 3-stage RFI "Hash" filter
**$59.95**

Order Factory Direct
215-865-0006
Out of State Order Toll Free
800-523-9685
PA Res. add 6%
COD Add 3.00 + Shipping
Dealers Invited

KALGLO ✓485
Electronics Co. Inc.
Colony Drive Ind. Park
6584 Ruch Rd., Dept. 80
Bethlehem, PA 18017

# TRS-80 Model I is alive and well at the Micromint. We still have the expansion interfaces you need!

## Disk-80 Expansion Interface

As featured in Ciarcia's Circuit Cellar
Byte Magazine, March 1981

Reviewed in March '82 "80 Microcomputing"

| | | |
|---|---|---|
| DSK01 | Disk-80 Expansion Interface with 32k RAM A & T | $330.00 |
| DSK02 | Disk-80 Expanstion Interface with 32k RAM & Printer Port A & T | 380.00 |
| DSK03 | Disk-80 Complete Kit with 32k RAM & Printer Port | 275.00 |
| DSK04 | Disk-80 Bare Printed Circuit Board | 48.00 |
| DSK05 | Printer/Power Supply Circuit Board | 16.00 |

The Disk-80 Expansion Interface is the perfect peripheral for converting your TRS-80 Model I into a professional computer system. The Disk-80 controls up to four 35 to 77 track mini-disk drives, and contains a hardware data separator which substantially increases the reliability of data transfers. Attaches to the CPU/Keyboard connector and comes complete with mini-disk controller, 32K expansion memory, power supply, optional Centronics compatible printer port, real time clock and buffered bus expansion connector.

*"Reviewing Disk-80 is almost incongruous, because any comments can be summarized with the sentence, "It works." Dennis Bathory Kitsz, 80 Microcomputing, March 1982*

All interfaces are Radio Shack hardware and software compatible and carry a 60 day warrantee including parts and labor.
All units include user's manual, power supply & auxiliary TRS-BUS connector for future expansion.

**Dealer Inquires Invited**

N.Y. State residents please add appropriate sales tax

**To Order: Call Toll Free: 1-800-645-3479**
**(In N.Y. State Call: 1-516-374-6793)**
**For Information Call: 1-516-374-6793**

**MICROMINT INC.**
**917 Midway**
**Woodmere, N.Y. 11598**

MICRO MINT

## Comm-80

As featured in Ciarcia's Circuit Cellar
Byte Magazine, May and June 1980

The COMM-80 is the only interface you need to turn your TRS-80 Model I into a time sharing terminal with provisions for a printer. The COMM-80 combines the most used features of the RS expansion interface in a low cost unit containing a built-in RS-232-C interface, a full 8-bit parallel port and a 40 pin bus connector for future expanstion. Terminal software is included at no extra cost.

- RS-232-C Serial output port (50-19200 baud) with standard DB25 connector
- Centronics compatible parallel printer port (34 pin)
- 40 pin card edge connector for bus expansion
- Connects directly to CRT terminals, modems, printers, other computers
- Includes case, power supply & interconnecting cable

| | | |
|---|---|---|
| CM01 | COMM-80 Serial/Parallel I/O Interface A&T | $150.00 |

TRS-80 is a trademark of Tandy Corp. ✓117





*Listing continued*

```
7EFD 2002       00720          JR     NZ,CHK2 ;CHECK FOR LIST REQ
7EFF 23         00722          INC    HL      ;POINT TO NEXT CHARA
C
7F00 7E         00730          LD     A,(HL)  ;LOAD 2ND BYTE
7F01 FE3B       00740 CHK2:    CP     ';'     ;NEXT LINE REQ
7F03 2859       00750          JR     Z,FNXT  ;YES, FIND CURRENT L
INE
7F05 FECE       00760          CP     0CEH    ;PREV LINE REQ
7F07 286A       00770          JR     Z,FPREV ;YES, FIND CUR LINE
7F09 FE40       00780          CP     '@'     ;THIS LINE REQ (AT S
IGN)
7F0B 2803       00790          JR     Z,BLDCM ;YES, BUILD CMD
7F0D B7         00800          OR     A       ;THIS LINE REQ
7F0E 2043       00810          JR     NZ,EXIT3  ;NOT A SHORT NOTAT
ION
7F10 3AE641     00820 BLDCM:   LD     A,(BUFAD) ;RESTORE 1ST CHARA
C
7F13 FE45       00830          CP     'E'     ;WAS THIS A EDIT REQ

7F15 2804       00840          JR     Z,BLDC4 ;YES SET TOKEN
7F17 3EB4       00850 BLDC3:   LD     A,0B4H  ;LIST TOKEN
7F19 1802       00860          JR     BLDC6
7F1B 3E9D       00870 BLDC4:   LD     A,09DH  ;EDIT TOKEN
7F1D 21E641     00880 BLDC6:   LD     HL,BUFAD        ;BUFFER
7F20 77         00890          LD     (HL),A  ;STORE IT
7F21 23         00900          INC    HL
7F22 3E2E       00910          LD     A,2EH   ;THE PERIOD
7F24 77         00920          LD     (HL),A  ;STORE IT
7F25 23         00930          INC    HL
7F26 AF         00940          XOR    A       ;LAST BYTE
7F27 77         00950          LD     (HL),A
7F28 C1         00960          POP    BC      ;RESTORE STACK
7F29 ED4B3E40   00970          LD     BC,(PRVRQ)      ;PREV REQUES
T
7F2D 2A2040     00980          LD     HL,(4020H)      ;PROMPT CURS
OR
7F30 37         00990          SCF
7F31 3F         01000          CCF
7F32 ED42       01010          SBC    HL,BC   ;DECR CURSOR POS.
7F34 7C         01020          LD     A,H  ;WAS DECR PAST START OF
 SCREEN
7F35 FE3C       01030          CP     3CH
7F37 3803       01040          JR     C,NOINC ;DONT MOVE PAST STAR
T
7F39 222040     01050          LD     (4020H),HL      ;STORE IT
7F3C 3AE641     01060 NOINC:   LD     A,(BUFAD)       ;SAVE COMMAN
D
7F3F 218000     01070          LD     HL,080H ;DEC FOR LIST
7F42 FEB4       01080          CP     0B4H
7F44 2803       01090          JR     Z,EXIT0
7F46 214000     01100          LD     HL,040H ;DEC FOR EDIT
7F49 223E40     01110 EXIT0:   LD     (PRVRQ),HL      ;STORE IT
7F4C 010700     01120          LD     BC,7    ;SET CMD LENGTH
7F4F E1         01130 EXIT1:   POP    HL      ;RESTORE REGS
7F50 D1         01140          POP    DE
7F51 F1         01150 EXIT4:   POP    AF
7F52 C9         01160          RET            ;RETURN TO ROM PROCE
SSOR
7F53 C1         01170 EXIT3:   POP    BC
7F54 18F9       01180          JR     EXIT1
7F56 ED5BEC40   01190 FLINE:   LD     DE,(TLNO)   ;FIND CURRENT LIN
E NO.
7F5A CD2C1B     01200          CALL   FLNO    ;USE ROM ROUTINE
7F5D C9         01210          RET
7F5E CD567F     01220 FNXT:    CALL   FLINE   ;FIND CUR LINE
7F61 CD667F     01230          CALL   LODLA   ;INCR TO NEXT LINE
7F64 18AA       01240          JR     BLDCM   ;BUILD LIST OR EDIT
CMD
7F66 7E         01250 LODLA:   LD     A,(HL)  ;CHECK FOR END OF PR
OG
7F67 23         01260          INC    HL      ;POINT TO LINE NO.
7F68 B6         01270          OR     (HL)    ;END IF ZERO
7F69 C8         01280          RET    Z       ;END ENCOUNTERED
7F6A 23         01290 LODLN:   INC    HL
7F6B 4E         01300 LODLP:   LD     C,(HL)  ;LOAD LINE NO.
7F6C 23         01310          INC    HL
7F6D 46         01320          LD     B,(HL)
7F6E ED43EC40   01330          LD     (TLNO),BC       ;STORE IT
7F72 C9         01340          RET
7F73 CD567F     01350 FPREV:   CALL   FLINE   ;LOCATE CUR LINE
7F76 2AA440     01360          LD     HL,(BASPT)  ;POINT TO BASIC
PROG
7F79 EB         01370          EX     DE,HL
7F7A EB         01380 FPRV1:   EX     DE,HL
7F7B 5E         01390          LD     E,(HL)  ;LOAD POINTER
7F7C 23         01400          INC    HL
7F7D 56         01410          LD     D,(HL)
7F7E 7A         01420          LD     A,D     ;COMPARE BC-DE
7F7F 90         01430          SUB    B
7F80 38F8       01440          JR     C,FPRV1
7F82 200C       01450          JR     NZ,BLDCM
7F84 7B         01460          LD     A,E
7F85 91         01470          SUB    C
7F86 38F2       01480          JR     C,FPRV1
7F88 2086       01490          JR     NZ,BLDCM ;CANT FIND BUT SEND
 COMMAND
7F8A CC6A7F     01500          CALL   Z,LODLN ;LOAD LINE NO
7F8D 1881       01510          JR     BLDCM   ;BUILD COMMAND
                01520 ;
                01530 ;    ROUTINE TO SEARCH BASIC PROG BUFFER FOR
GIVEN
                01540 ;    CHARAC STRING
                01550 ;
```

*Listing continues*

*Listing continued*

```
7F8F 79        01560 PSRCH:   LD      A,C       ;WAS STRING INPUT
7F90 FE06      01570          CP      6
7F92 2053      01580          JR      NZ,LODBF ;YES LOAD BUFFER
7F94 CD567F    01590          CALL    FLINE    ;FIND CUR LINE
7F97 E5        01600          PUSH    HL       ;SAVE PTR
7F98 181B      01610          JR      SRCHX    ;CONT SEARCH
7F9A CD567F    01620 PSRCX:   CALL    FLINE ;FIND LOC OF CURRENT L
INE
               01630                           ;BC POINTS TO CURREN
T LINE
               01640                           ;HL POINTS TO THE NE
XT LINE
7F9D E5        01650 SRCHA:   PUSH    HL       ;SAVE POINTER
7F9E 03        01660          INC     BC       ;SKIP PAST POINTER A
DDR
7F9F 03        01670          INC     BC
7FA0 ED434140  01680          LD      (LNPTR),BC ;SAVE POINTER TO
LINE NO
               01690                              ;FOR LATER USE ON
 MATCH
7FA4 03        01700          INC     BC       ;ADJUST FOR LINE NO
7FA5 37        01710          SCF
7FA6 ED42      01720          SBC     HL,BC    ;NO. OF CHARAC IN TH
IS LINE
7FA8 03        01730          INC     BC ;PT TO START OF LINE TEXT

7FA9 C5        01740          PUSH    BC       ;EXCHANGE HL/BC
7FAA E5        01750          PUSH    HL       ;EXCHANGE HL/BC
7FAB C1        01760          POP     BC
7FAC E1        01770          POP     HL       ;RESTORE POINTER TO
NEXT LINE
7FAD 114340    01780 SRCHK:   LD      DE,STRING  ;ADDR OF SEARCH S
TRING
7FB0 1A        01790          LD      A,(DE)   ;LOAD FIRST CHARAC
7FB1 EDB1      01800          CPIR             ;SEARCH FOR FIRST CH
ARAC
7FB3 280E      01810          JR      Z,SRCH2  ;MATCH FIRST CHARAC
7FB5 E1        01820 SRCHX:   POP     HL       ;RESTORE POINTER TO
NEXT LINE
7FB6 E5        01830          PUSH    HL
7FB7 C1        01840          POP     BC
7FB8 5E        01850          LD      E,(HL)   ;LOAD NEXT POINTER
7FB9 23        01860          INC     HL
7FBA 56        01870          LD      D,(HL)
7FBB EB        01880          EX      DE,HL    ;EXCHANGE REGS
7FBC 7C        01890          LD      A,H      ;CHECK FOR END OF BU
FFER
7FBD B5        01900          OR      L
7FBE 20DD      01910          JR      NZ,SRCHA ;NO, CONTINUE SEARC
H
7FC0 C3CC06    01920          JP      06CCH    ;RETURN GO BASIC
7FC3 3A4040    01930 SRCH2:   LD      A,(STRLN)      ;LOAD STRING
 LENGTH
7FC6 B7        01940          OR      A        ;IS STRING A SINGLE
CHARAC
7FC7 280B      01950          JR      Z,SRCH4 ;YES
7FC9 47        01960          LD      B,A
7FCA E5        01970          PUSH    HL       ;SAVE LINE POINTER
7FCB 13        01980 SRCH3:   INC     DE       ;GET NEXT CHARAC
7FCC 1A        01990          LD      A,(DE)
7FCD BE        02000          CP      (HL)
7FCE 23        02010          INC     HL
7FCF 200D      02020          JR      NZ,SRCH6         ;CONT SEARCH

7FD1 10F8      02030          DJNZ    SRCH3
7FD3 E1        02040          POP     HL       ;RESTORE STACK
7FD4 E1        02050 SRCH4:   POP     HL
7FD5 2A4140    02060          LD      HL,(LNPTR)       ;LOAD LINE N
O.
7FD8 CD6B7F    02070          CALL    LODLP
7FDB C3177F    02080          JP      BLDC3    ;REQUST LIST
7FDE E1        02090 SRCH6:   POP     HL       ;RESTORE POINTER
7FDF 0600      02100          LD      B,0      ;RESTORE COUNTER
7FE1 0D        02110          DEC     C        ;MOVE SEARCH POINTER
 1 POS
7FE2 28D1      02120          JR      Z,SRCHX ;CONT SRCH
7FE4 23        02130          INC     HL       ;MOVE POINTER 1 POS
7FE5 18C6      02140          JR      SRCHK    ;CONT SRCH THIS LINE

               02150 ;
               02160 ;LOAD SEARCH STRING BUFFER
               02170 ;
7FE7 21E741    02180 LODBF:   LD      HL,BUFAD+1       ;POINT TO BU
FFER
7FEA 0600      02190          LD      B,0      ;CLEAR COUNTER
7FEC 114340    02200          LD      DE,STRING        ;PT TO SRCH
BUFFER
7FEF 7E        02210 LOADB:   LD      A,(HL)   ;MOVE CHARAC TO BUFF
ER
7FF0 B7        02220          OR      A        ;CHECK FOR END
7FF1 2806      02230          JR      Z,LOADM
7FF3 12        02240          LD      (DE),A   ;STORE IT
7FF4 04        02250          INC     B
7FF5 23        02260          INC     HL       ;PT TO NEXT CHARAC
7FF6 13        02270          INC     DE       ;PT TO NEXT STRING C
HARAC
7FF7 18F6      02280          JR      LOADB    ;LOOP TIL END
7FF9 05        02290 LOADM:   DEC     B
7FFA 78        02300          LD      A,B      ;STORE STRING LEN
7FFB 324040    02310          LD      (STRLN),A
7FFE 189A      02320          JR      PSRCX    ;GO SEARCH
7ECD           02330          END     JEHMN2
00000 TOTAL ERRORS
```



✓See List of Advertisers on Page 451

*Morse Code in color.*

# CC CQ

*Michael J. Chuck*
*648 Longmeadow Drive*
*Severna Park, MD 21146*

It was only a matter of time before I decided to merge my two hobbies—computers and amateur radio. The Radio Shack 16K Extended Basic Color Computer, well shielded against radio frequency interference, makes a very good candidate for a Morse code terminal.

My first step was defining the features I wanted in a Morse terminal. After studying many ads for Model I terminals, I came up with the following wish list for my system:

- Code speeds of 5-60 words per minute
- Automatic speed adjustment when receiving
- Split-screen operation
- Reasonably low cost
- Ability to compose messages while receiving
- Menu-driven transmit for pre-made messages
- Transmit buffer to allow typing ahead of the keyer
- Automatic file handling of the QSO information
- Ability to update the current QSO file while receiving
- Morse code practice feature

After many program edits, the wish list finally became a reality and my Color Computer was operating on the air. I was so pleased with its performance that I decided to pass my ideas along to all you hams with Color Computers.

**The Key Box**

**Color Computer**
**Extended Color Basic**
**16K RAM**
**Amateur radio receiver and transmitter**

### System Description

Both the transmit and receive routines are written in Assembly language to take advantage of the higher speed of execution. The transmit portion of the program uses a look-up table to convert ASCII code to the dots and dashes of Morse code.

Since Morse consists of only dots and dashes, a binary number can represent the code sequence of a letter. The program uses one to represent a dash and zero to represent a dot. All code sequences must begin with one, however, to distinguish between letters that start with dots in Morse. For example, the letter S can be represented in a table by 00001000 and Z by 00011100. By performing successive left shifts on the binary value, the dots and dashes can be peeled off after locating the leading one.

The transmit table begins at hex 3B40, so by substituting the ASCII value of the letter to be transmitted for the least-significant byte (40) the decode address is formed. For example, the letter C is hex 43 in ASCII. Therefore, at address hex 3B43 the value 26 is stored (binary 00011010). After removing the leading one the code 1010 remains, so the letter C (dash dot dash dot) would be sent.

The actual keying is accomplished with the cassette motor control relay. By setting bit 3 at address FF21 (65313 decimal) the relay is energized. After waiting the appropriate time for a dot or dash, bit 3 is reset to deenergize the relay. The computer relay output at the cassette motor control plug is capable of using my Heathkit electronic keyer directly. If you attempt to key a transmitter directly, however, use a low-voltage isolation relay to protect the computer relay.

The receive portion of the program uses the serial I/O jack on the Color Computer to input Morse signals. If bit 1 at address 65314 is set, data is present on the data input line (pin 2 of the connector). However, the computer requires a signal greater than +3 volts. An inexpensive two-transistor amplifier boosts the receiver phone jack signal to a higher level. Since the amplifier actually supplies a pulsating dc voltage, a digital filter (time delay) was incorporated into the receive routine.

The purpose of the filter is to ensure that the computer disregards the high-frequency changes in the signal. A filter value of 30 is loaded initially. If on high-

| | |
|---|---|
| 40 | Entry point of the transmit mode |
| 90 | Entry point of the message menu |
| 110-300 | Decode of the menu selection |
| 310-400 | Code practice generator section |
| 410 | Entry point of the receive mode |
| 430-560 | Store the transmit buffer and scanned file data when returning to the transmit mode |
| 570-660 | Calculate transmit speed factor and store |
| 670-740 | Preview canned messages prior to transmission |
| 750-830 | File update menu |
| 840 | Beginning of file-handling routine |

*Table 1. HAMSHACK functional-block line numbers*

**During transmit**

↑—Call the message menu
↓—Call the current QSO file update menu
#—Send AR
@—Send a string of dots (error)
&—Send SK
Enter—Switch to the receive mode

**During receive**

↑—Start scanning his name
↓—Start scanning his call sign
→—Start scanning his QTH
Clear—Stop the scanning process
Enter—Switch to the transmit mode

*Table 3. Key functions*

speed receiving some letters are lost, you can relax the filter. Enter Delay from the menu, and then a new lower value. The filter should be as high as possible (less than 256) and yet receive properly.

Decoding data-present signals from Morse to ASCII uses the same technique as the transmit portion. After loading a leading one and shifting left, a zero is loaded if the data is present less than two dot times, and a one is loaded if the data is present for more than two dot times. A left shift is performed after each successive digit is loaded. During receive the dot time is dynamically adjusted. If data is not present for greater than two dot times, the letter is decoded and displayed on the upper portion of the screen.

The table for decoding received signals is located at address hex 3F00. For example, the letter A is loaded as 00000101 during receive (hex 05). By substituting into the table address, 3F05 becomes the decoder address. A PEEK to the decoder address will return the number 65, which is the value to be POKEd to video RAM to display the letter A.

The Basic calling program is completely menu driven. The enter key is used as the transmit/receive switch and can be used to default to transmit mode from any menu. There are five message text buffers that can be pre-loaded and called up to transmit via the message menu. The up arrow key is used to enter the message menu from the transmit mode. Any message selected from the menu is displayed on the bottom portion of the screen for review before being transmitted. During review, if RST is to be sent, enter three numbers instead of a normal review reply to change the RST report from 599 to the new numbers.

Also callable from the message menu are three utility functions:

- Speed allows entering a new transmit code speed
- Delay allows entering a new receive filter delay value
- Test sends random code for practice

While in the transmit mode, the down arrow calls up the current QSO log menu. You can make changes in any field from this menu. After a QSO is completed it can be saved to page 1 of video RAM from this menu also. Depending on the amount of data in each field of the file, 15 to 30 files can be saved to RAM before taping and clearing the storage area.

From the file update menu, reviews, tape saves, verifies and more can be done. A verify must always be performed after a tape save to clear the file storage area. (Be sure to replace the cassette motor control plug into the cassette and remove this program before tape saves.)

File data is saved to tape as a string with field separators as follows:

```
STRINGS$ = "<" + T0$ + "%" + T1$ + "%" + FQ$ + "%"
           HC$ + "%" + HN$ + "%" + QT$ + ">"
     T0$ = QSO start time
     T1$ = QSO finish time
     FQ$ = Frequency used
     HC$ = His call sign (station worked)
     HN$ = His name
     QT$ = His QTH and other remarks
```

By performing string searches for the separators, the file data can be recovered. (How about a program to check for duplicate QSOs during a contest, or to print the log? The log data is all available on tape, so let your imagination go and program your computer to do a little more of your work.)

To ease data entry into the log and the canned messages, three arrow keys can be pressed during receive to scan and file data as it is displayed. The other ham's call, name and remarks can be scanned by pressing ↓, ↑, → respectively. The received video will be displayed in the inverse mode to verify what is scanned. Pressing clear stops the scan. Scanning erases any data that was in the scanned field of the log from a previous receive mode, but it is cumulative during a given receive mode until returning to the transmit mode. For example, the QTH could be scanned and the scanning stopped. Later during the same receive mode, his rig description may be scanned into the same file field.

While in the receive mode, a message can be composed on the bottom of the screen. Since this message is POKEd directly into M6$ (line 7 of the program), you must keep the text in the correct area. Backspacing beyond the start position (print position 288) or entering characters beyond the end of screen can destroy the Basic program. The technique of POKEing directly to a string constant proved to be the fastest

| Address | Usage |
|---|---|
| 1536-3071 | QSO file storage |
| 3098-3327 | M6$ (line 7) used to store the transmit text composed during the receive mode |
| 14653-15167 | Transmit routine |
| 15168-15231 | Transmit conversion table (ASCII to Morse) |
| 15232 | Value to POKE to 65313 to turn on cassette relay |
| 15233-15234 | Two times the transmit speed dot time |
| 15235-15236 | Seven times the transmit speed dot time (word spacing) |
| 15237-15737 | Receive routine |
| 15738-15978 | Keyboard scan and character-handling subroutine |
| 15981-15982 | M6$ pointer (points to the next POKE position) |
| 15983-15984 | "No data present" timer count on receive. Transmit pointer on transmit (points to position of next letter position on the bottom of screen to be transmitted) |
| 15985-15986 | Transmit buffer cursor counter (position on bottom of screen to display next letter from the keyboard) |
| 15987-15988 | Top of screen cursor counter |
| 15989 | Value to POKE 65313 to turn off cassette relay |
| 15990-15991 | Six times the receive dot time (used for word space) |
| 15992-15993 | Four times the receive dot time (used as dash compare for automatic speed adjustment) |
| 15994-15995 | Two times the received dot time (used for dot/dash selection and for element/character selection) |
| 15996-15997 | Receive dot time (used for automatic speed adjustment) |
| 15998-15999 | Timing counter storage |
| 16000 | Received letter decoder value (value to display) |
| 16001 | ASCII value of last key input |
| 16002-16003 | Storage position pointer for scan of his name |
| 16004-16013 | Name scan storage area |
| 16014-16015 | Storage position pointer for scan of his call |
| 16016-16025 | Call scan storage area |
| 16026-16027 | Storage position pointer for scan of remarks |
| 16028-16126 | Remarks scan storage area |
| 16127-16383 | Receive Morse to ASCII decoder table |

*Table 2. Memory use*

method of handling the message when returning to the transmit mode.

One of the five message buffer numbers may be selected for the text when returning to transmit. If none is selected (you press only enter) buffer 1 is used. Because of this default, message 1 should not be filled when starting the program if you plan to prepare text while receiving. You may, however, change any message during program execution by typing during receive and specifying the appropriate buffer number to store the text in.

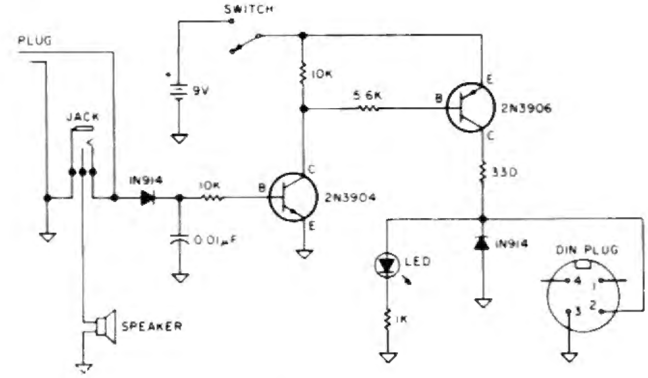



*Fig. 1. Schematic diagram (see Table 4)*

| | |
|---|---|
| Transistor 2N3906 | (RS #276-1604) |
| Transistor 2N3904 | (RS #276-1603) |
| Diode 1N914 | (RS #276-1620) |
| LED and holder | (RS #276-068) |
| Capacitor .01uf | (RS #272-131) |
| Resistor 10K ohms | (RS #271-034) |
| Resistor 5.6K ohms | (RS #271-031) |
| Resistor 330 ohms | (RS #271-017) |
| Resistor 1K ohms | (RS #271-023) |
| Power switch | (RS #275-602) |
| Speaker 2" | (RS #40-245) |
| DIN Plug | (RS #26-3020) |
| Phone plug ¼" | (RS #42-2377) |
| Phone jack ⅛" | (RS #274-255) |
| Mini box | (RS #270-251) |
| Component board | (RS #276-160) |
| 9 volt battery | (RS #23-464) |
| Battery holder | (RS #270-326) |
| Battery clip | (RS #270-325) |

*Table 4. Parts list*

If the radio receiver is inactive, press enter to return to the transmit mode. After a minute or two in the receive mode with no activity, the automatic speed compensation software may not recover properly. This is not a problem because the transmit mode is the desired standby mode. From transmit the menu calls are available and pressing enter immediately switches to the receive mode.

During transmit a wraparound buffer (the bottom portion of the screen) is used for typing. As the program sends a character it clears it from the bottom buffer and displays it on the top of the screen. This allows the typing to be more than 200 characters ahead of the transmitter. The buffer turned out to be a real plus for a moderate typist working at five-words-per-minute code speed!

### The Interface Amplifier

Since the audio signal of my radio receiver was too low to be detected by the RS-232 input, a simple amplifier was required. I decided on a very simple circuit us-

ing parts available from the local Radio Shack store.

The two-transistor circuit rectifies and produces a positive output voltage whenever a tone is present at the radio receiver earphone jack. An LED gives a visual indication that the proper level of output voltage is being supplied to the computer. In operation the receiver gain is adjusted to a level to activate the LED as no gain control was placed in the interface amplifier circuit.

Figure 1 shows the schematic for the circuit. Table 4 lists the components required. I used a general-purpose circuit board and just started soldering. Layout is not critical. If you can read a simple schematic and do a little soldering, you will have no trouble with the interface.

**Program Entry**

Since QSO logs are saved in page 1 of the video RAM, always perform a PCLEAR1 when turning on your computer. Hamload (Program Listing 1) is run prior to Hamshack (Program Listing 2) to load the transmit and receive routines. Since Hamload contains all of the Assembly-language data, it is a bit tedious to key in. Be patient! Errors in data statements are difficult to find and have disastrous results.

I wrote Hamshack in a packed form to conserve memory, so it is also a bit tedious. When keying it in, enter lines 5 and 7 exactly as listed! The receive routine POKEs directly to M6$ and adding a space or changing a line number to two digits will cause problems.

Some user-specific information must be inserted in lines 130, 140 and 160. Also, there is room for two additional messages in the message menu. By studying lines 110-190, the addition should be self-explanatory.

If some day you hear KA9EQD calling CQ, please give me a call. My Color Computer has been dying to talk to one of his own kind! ■

*Michael Chuck, a mechanical engineer, enjoys amateur radio and model railroading.*

*Program Listing 1*

```
10 CLS:PRINT"ASSEMBLY LANGUAGE LOADER":CLEAR10,14650:FORK=15237T
O15978:READ A:POKE K,A:PRINT@256,K,A:NEXTK
20 FORK=15979TO16127:POKE K,0:PRINT@256,K,0:NEXTK
30 FORK=16157TO16382:POKE K,0:PRINT@256,K,0:NEXTK
40 FORK=16128TO16157:READ A:POKEK,A:PRINT@256,K,A:NEXTK:POKE1616
0,53:POKE16161,52:POKE16163,51:POKE16167,50:POKE16168,43:POKE161
70,35:POKE16175,49:POKE16176,54:POKE16177,45:POKE16178,47:POKE16
184,55:POKE16188,56:POKE16190,57:POKE16191,48:POKE16197,38
50 FORK=14653TO15231:READA:POKEK,A:PRINT@256,K,A:NEXTK
60 POKE16204,63:POKE16213,46:POKE16234,59:POKE16237,40:POKE16243
,44:POKE16248,58:POKE16383,0:CLS:PRINT"LOAD AND RUN hamshack":EN
D
70 DATA252,62,111,131,0,30,46,18,252,62,111,243,62,126,253,62,12
6,204,0,0,253,62,111,126,59,247,182,62,128,72,183,62,128,126,60,
60,204,0,0,253,62,111,126,60,188
80 DATA204,12,26,253,62,109,204,5,32,253,62,113,204,4,0,253,62,1
15,134,0,183,62,129,204,62,132,253,62,130,204,62,144,253,62,142,
204,62,156,253
90 DATA62,154,182,255,34,129,6,38,42,134,1,183,62,128,204,0,0,25
3,62,111,253,62,126,182,255,34,129,6,38,35,252,62,126,195,0,1,25
3,62,126
```

*Listing continues*

*Listing continued*

```
100 DATA189,61.122,182,62,128,182,62,128,126,59,240,189,61,122,1
82,62,129,129,13,38,1,57,126,59,219,182,255,34,129,7,38,23,252,6
2,111,195,0,1
110 DATA253,62,111,189,61,122,182,62,129,129,13,38,1,57,126,60,2
6,126,59,133,18,252,62,126,179,62,122,45,8,182,C2,128,139,1,183,
62,128,252,62,126
120 DATA179,62,124,46,24,252,62,126,253,62,124,243,62,126,253,62
,122,243,62,122,253,62,120,243,62,122,253,62,118,252,62,126,179,
62,120,45,43,252,62,126
130 DATA253,62,120,182,62,120,68,183,62,122,182,62,121,70,183,62
,123,182,62,122.68,183,62,124,182,62,123,70,183,62,125,252,62,12
0,243,62,122,253,62,118
140 DATA252,62,111.179,62,122,45,3,126,60,182,204,0,0,253,62,126
,253,62,111.126,59,240,204,0,0,253,62,126,182,62,128,183,60,196,
182,63,0
150 DATA183,62,128,182,62,129,129,94,39,26,129,10,39,49,129,9,39
,72,182,62,128,128,64,46,8,182,62,128,139,64,183,62,128,126,61,6
5,252,62,130
160 DATA253,60,255,195,0,1,16,131,62,142,39,3,253,62,130,182,62,
128,183,0,0.126,61,55,252,62,142,253,61,26,195,0,1,16,131,62,154
,39,3
170 DATA253,62,142,182,62,128,183,0,0,126,61,55,252,62,154,253,6
1,53,195,0,1,16,131,63,0,39,3,253,62,154,182,62,128,183,0,0,182,
62,128
180 DATA128,64,45,3,183,62,128,252,62,115,253,61,90,195,0,1,16,1
31,5,0,38,3,204,4,0,253,62,115,182,62,128,183,0,0,252,62,111
190 DATA179,62,118,46,14,204,0,0,253,62,111,134,1,183,62,128,126
,59,240,134,1,183,62,128,126,59,169
200 DATA189,161.193,129,0,39,8,183,62,129,18,18,126,61,189,182,6
2,129,129,0,39,42.129,94,39,19,129,10,39,17,129,9,39,15,129,13,3
9,41
210 DATA129,12,39,14,126,61,232,30,137,30,137,30,137,30,137,126,
61,212,134,0,183,62,129,126,61,212.204,0,1,30,137.30,137,30,137,
30,137,18
220 DATA126,61,212,252,62,109,253,61,210,134,62,183,0,0,30,137,3
0,137,30,137.30,137,30,137,30,137,30,137,204,0,1,30,137,57,129,8
,39,79,128,64
230 DATA45,37,252,62,109,253,62,12,195,0,1,253,62,109,252,62,113
,253,62,15,195,0,1,253.62,113,182,62,129,183,0,0,183,0,0,18,126,
62,101
240 DATA252,62,109,253,62,49,195,0,1,253,62,109,252,62,113,253,6
2,54,195,0,1,253,62,113,182,62,129,183,0,0,139,64,183,0,0,126,62
,101
250 DATA252,62,109,131.0,1,253,62,77,195,0,0,253,62,109,134,32,1
83,0,0,252,62,113.131,0,1,253,62,97,195,0,0,253,62,113,134,96,18
3,0,0
260 DATA18,18,134,0,183,62,129,57
270 DATA0.32,69,84,73,65,78,77,83,85,82,87,68,75,71,79,72,86,70,
0,76,0,80,74,66,88,67,89,90,81
280 DATA204,12,126,253,62,109,182,62,114,177,62,112,39,57,182,62
,112,183,57,105,183,57,97,139,1,129,255,38,2,134,32,183,62,112,1
82,5,0,183,62,130.134,96,183,5,0
290 DATA182,62,116,183,57,128,139,1,129,255,38,2,134,0,183,62,11
6,182,62,130,183,4,0,126,57,166.189,61,122,182,62,129,129,13,39,
23,129,10
300 DATA39,19,129,94,39,15,182,62,114,129,255,38,5,134,32,183,62
,114,126,57,61,57,182,62,130,183,57,174,182,59,0,183,62,130,120,
62,130,37,76,120,62,130,37,84,120,62,130,37,92
310 DATA120,62,130,37,100,120,62,130,37,108,120,62,130,37,116,12
0,62,130,37,124,204,0,0,195,0,1,253,62,126,189,61,122,182,62,129
,129,94,39,14,182,62,114,129,255,38,5,134,32,183,62,114,32,1,57,
252,62,126,16,179,59,131,38,216,126,57,61
320 DATA120,62,130,37,5,189,58,205,32,3,189,58,148,120,62,130,37
,5,189,58,205,32,3,189,58,148,120,C2,130,37,5,189,58,205,32,3,18
9,58,148
330 DATA120,62,130,37,5,189,58,205,32,3,189,58,148,120,62,130,37
,5,189,58,205,32,3,189,58,148,120,62,130,37,5,189,58,205,32,3,18
9,58,148
340 DATA120,62,130,37,5,189,58,205,32,3,189,58,148,204,0,0,195,0
,1,253,62,126,189,61,122,182,62,129,129,13,39,22,129,10,39,18,12
9,94,39,14,182,62,114,129,255,38,5,134,32,183,62,114,32,1,57,252
,62,126,16,179,59,129,38,208,126,57,61,182,59,128,183,255,33
350 DATA204,0,0,195,0,1,253,62,126,189,61,122,182,62,129,129,13,
39,22,129,10.39,18,129,94,39,14,182,62,114,129,255,38,5,134,32,1
83,62,114,32,1,57,252,62,126,16,179,59,129,38,208,182,59,128,183
,255,33,204,0,0,195,0,1,253,62,126,189,61,122,182,62,129,129
360 DATA13,39,22,129,10,39,18,129,94,39,14,182,62,114,129,255,38
,5,134,32,183,62,114,32,1,57,252,62,126,16,179,62,107,38,208,182
,62,117,183,255,33,204,0,0
370 DATA195,0,1,253,62,126,189,61,122,182,62,129,129,13,39,22,12
9,10,39,18,129,94,39,14,182,62,114,129,255,38,5,134,32,183,62,11
4,32,1,57,252,62,126,16,179,62,107,38,208,57
380 DATA128,5,24,26,12,2,18,14,16,4,23,13,20,7,6,15,22,29,10,8,3
,9,17,11,25,27,28,0,0,0,0,0,0,85,0,42,0,0,69,0,109,109,0,40,115,
49,85,50,63,47,39,35,33,32,48,56,60,62,120,106,0,0,0,76
390 END
```



✓ *See List of Advertisers on Page 451*

*Program Listing 2*

```
5 CLEAR2000,14650
7 M6$="THIS STRING IS USED TO PASS THE TRANSMIT BUFFER INFORMATI
ON FROM ASSEMBLY LANGUAGE TO THE BASIC MENU PROGRAM -- IT IS ENT
ERED AS 230 DUMMY CHARACTERS ! ! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
10 CLS:INPUT"WPM";S:GOSUB570:CLS:K=PEEK(65313):POKE15989,K:S=K O
R 8:POKE15232,S:RS$="599":R4=1536:DEFUSR1=15282:DEFUSR2=14653
20 INPUT"YOUR CALL";CS$:CLS:GOSUB1220:PRINT"MESSAGE #1";:PRINT@6
6,"";:INPUTM1$:CLS:GOSUB1220:PRINT"#2";:PRINT@66,"";:INPUTM2$:CL
S:GOSUB1220:PRINT"#3";:PRINT@66,"";:INPUTM3$:CLS:GOSUB1220:PRINT
"#4";:PRINT@66,"";:INPUTM4$:CLS:GOSUB1220:PRINT"#5";
30 PRINT@66,"";:INPUTM5$:CLS:POKE15985,5:POKE15986,32
40 POKE16001,0:PRINT@256,"TRANSMIT MODE";:GOSUB680:POKE15987,4:P
OKE15988,0:POKE15983,5:POKE15984,32
50 K=USR2(0)
60 K=PEEK(15989):POKE65313,K
70 IF PEEK(16001)=10 GOTO750
80 IF PEEK(16001)=13 GOTO410
90 GOSUB670:POKE15985,5:POKE15986,32
100 PRINT@265,"MESSAGE MENU":PRINT:PRINT"CQ     CALL  QTH    RIG
 OVER     NAME  RST    73     QRM    ANS      1      2      3      4
 5                    TEST   DELAY SPEED":INPUT"SELECT";A$
110 IFA$="CQ" THEN X$="CQ CQ CQ DE "+CS$+" K ":GOTO690
120 IFA$="CALL" THEN X$=HC$+" "+HC$+" DE "+CS$+" KN ":GOTO690
130 IFA$="QTH" THEN X$="QTH QTH ENTER YOUR QTH HERE !":GOTO690
140 IFA$="RIG" THEN X$="RIG HR IS ENTER YOUR RIG HERE -":GOTO690
150 IFA$="OVER" THEN X$="BK TO U "+HC$+" DE "+CS$+" KN ":GOTO690
160 IFA$="NAME" THEN X$="NAME IS ENTER YOUR NAME -":GOTO690
170 IFA$="RST" THEN X$="RST RST "+RS$+" "+RS$+" - ":GOTO690
180 IFA$="73" THEN X$="73 TO U AND URS "+HN$+". HP TO WK U AGN S
OON. DE "+CS$+" K ":GOTO690
190 IFA$="QRM" THEN X$="QRM QRM PSE TRY AGN. DE "+CS$+" KN ":GOT
O690
200 IFA$="1" THEN X$=M1$:GOTO690
210 IFA$="2" THEN X$=M2$:GOTO690
220 IFA$="3" THEN X$=M3$:GOTO690
230 IFA$="4" THEN X$=M4$:GOTO690
240 IFA$="5" THEN X$=M5$:GOTO690
250 IFA$="ANS" THEN X$=HC$+" DE "+CS$+" R R - OK ON LAST XMIT "+
HN$+" - ":GOTO690
270 IFA$="SPEED" THEN GOSUB670:PRINT@256,"NEW WPM";:INPUTS:GOSUB
570:GOSUB670:GOTO40
280 IFA$="DELAY" THEN GOSUB670:PRINT@256,"NEW FILTER DELAY";:INP
UTK:IFK<256 THEN POKE15242,K:GOSUB670:GOTO40 ELSE PRINT"VALUE TO
O LARGE":FORK=1 TO 100:NEXTK:GOTO280
290 IFA$="TEST" GOTO310
300 GOSUB670:GOTO40
310 X$="":FORKK=1TO30:FORJJ=1TO5
320 K=RND(47)+40:IFK=41 THEN K=35
330 IFK=42 THEN K=38
340 IFK=43 THEN K=40
350 IFK>59 AND K<63 THEN GOTO320
360 X$=X$+CHR$(K)
370 NEXTJJ
380 X$=X$+" "
390 NEXTKK
400 A$="S":GOTO710
410 CLS:PRINT@256,"RECEIVE MODE":PRINT@0,"";:POKE15996,0
420 K=USR1(0):CLS:K=PEEK(15982)-26
430 M7$=LEFT$(M6$,K)
440 INPUT"BUFFER NUMBER (1-5)";K:ON K GOTO450,460,470,480,490
450 M1$=M7$:GOTO500
460 M2$=M7$:GOTO500
470 M3$=M7$:GOTO500
480 M4$=M7$:GOTO500
490 M5$=M7$
500 IFPEEK(16003)=132 GOTO520
510 HN$="":FORK=132TO PEEK(16003):L=PEEK(15872+K):HN$=HN$+CHR$(L
):NEXTK:IF RIGHT$(HN$,1)=" " THEN K=LEN(HN$)-1:HN$=LEFT$(HN$,K)
520 IFPEEK(16015)=144 GOTO540
530 HC$="":FORK=144TO PEEK(16015):L=PEEK(15872+K):HC$=HC$+CHR$(L
):NEXTK:IF RIGHT$(HC$,1)=" " THEN K=LEN(HC$)-1:HC$=LEFT$(HC$,K)
540 IFPEEK(16027)=156 GOTO560
550 QT$="":FORK=156TO PEEK(16027):L=PEEK(15872+K):QT$=QT$+CHR$(L
):NEXTK:IF RIGHT$(QT$,1)=" " THEN K=LEN(QT$)-1:QT$=LEFT$(QT$,K)
560 CLS:POKE15985,5:POKE15986,32:GOTO40
570 IFS<7.4 THEN S=150+31.25*(7.4-S):GOTO660
580 IFS<9.6 THEN S=115+15.9091*(9.6-S):GOTO660
590 IFS<13 THEN S=85+8.8235*(13-S):GOTO660
600 IFS<16.8 THEN S=65+5.2632*(16.8-S):GOTO660
610 IFS<22 THEN S=50+2.8846*(22-S):GOTO660
```

*Listing continues*



See List of Advertisers on Page 451

*Listing continued*

```
620 IFS<27.2 THEN S=40+1.9231*(27.2-S):GOTO660
630 IFS<36 THEN S=30+1.1364*(36-S):GOTO660
640 IFS<54 THEN S=20+.5556*(54-S):GOTO660
650 S=53:GOTO640
660 JJ=INT(S/256):KK=S-JJ*256:POKE15979,JJ:POKE15980,KK:K=2*S:JJ
=INT(K/256):KK=K-JJ*256:POKE15233,JJ:POKE15234,KK:K=6.5*S:JJ=INT
(K/256):KK=K-JJ*256:POKE15235,JJ:POKE15236,KK:RETURN
670 PRINT@256,STRING$(255," "):RETURN
680 PRINT@0,STRING$(255," "):PRINT@0,"";:RETURN
690 GOSUB670:PRINT@256,X$:PRINT@480,"S=SEND MESSAGE  ";CHR$(94);
"=MENU";:INPUTA$:GOSUB680
700 IFA$=CHR$(94) GOTO90
710 IFA$="S" THEN GOSUB670:PRINT@288,X$;:K=LEN(X$)+32:POKE15986,
K:GOTO40
730 IF LEFT$(X$,3)="RST" THEN RS$=A$:X$="RST RST "+RS$+" "+RS$+"
 - ":GOTO690
740 GOSUB670:GOTO40
750 POKE15985,5:POKE15986,32:GOSUB670:PRINT@265,"UPDATE MENU":PR
INT"C= ";HC$;:PRINT TAB(15)"N= ";HN$:PRINT"F= ";FQ$;:PRINT TAB(1
5)"X=START= ";T0$:PRINT"?= ";QT$:PRINT"S=SAVE":INPUT"SELECT";X$
760 IFX$="C" THEN GOSUB670:PRINT@256,"CALL=";HC$:INPUT"ENTER HIS
 CALL";HC$:GOTO750
770 IFX$="N" THEN GOSUB670:PRINT@256,"NAME=";HN$:INPUT"ENTER HIS
 NAME";HN$:GOTO750
780 IFX$="F" THEN GOSUB670:PRINT@256,"FREQ=";FQ$:INPUT"ENTER THE
 FREQ";FQ$:GOTO750
790 IFX$="X" THEN GOSUB670:PRINT@256,"START TIME=";T0$:INPUT"ENT
ER THE START TIME";T0$:GOTO750
800 IFX$="/" OR X$="?" THEN GOSUB670:PRINT@256,"REMARKS=";QT$:IN
PUT"ENTER REMARKS (QTH)";QT$:GOTO750
810 IFX$=CHR$(94) GOTO90
830 IFX$<>"S" THEN GOSUB670:GOTO40
840 GOSUB670:PRINT@256,"ENTER THE STOP TIME";:INPUTT1$:GOSUB670
850 PRINT@265,"FILE MENU":PRINT"T=TAPE ALL FILES":PRINT"S=SAVE C
URRENT FILE TO RAM":PRINT"V=VERIFY TAPE DATA":PRINT"R=REVIEW FIL
ES":PRINT"Q=RETURN TO FILE DATA":INPUT"SELECT";X$
860 IFX$="Q" GOTO750
870 IFX$="T" GOTO920
880 IFX$="S" GOTO950
890 IFX$="V" GOTO980
900 IFX$="R" GOTO1050
910 GOSUB670:GOTO40
920 GOSUB670:PRINT@256,"SET CASSETTE TO RECORD":INPUT"PRESS ENTE
R";A$:OPEN "O",#-1,"FILE"
930 FORKK=1536TO3071:PRINT #-1,PEEK(KK):NEXT
940 CLOSE #-1:GOTO750
950 X$="<"+T0$+"%"+T1$+"%"+FQ$+"%"+HC$+"%"+HN$+"%"+QT$+">":AL=30
71-R4:LS=LEN(X$):IFLS>AL THEN GOSUB670:PRINT@256,"NO ROOM LEFT!"
:FORKK=1TO3000:NEXTKK:GOTO750
960 FORKK=1 TO LS:A$=MID$(X$,KK,1):JJ=ASC(A$):POKE R4-1+KK,JJ:NE
XTKK
970 R4=R4+LS:HC$="":HN$="":QT$="":T0$=T1$:RS$="599":GOTO750
980 ON ERROR GOTO1030:GOSUB670:PRINT@256,"SET CASSETTE TO PLAY":
INPUT"PRESS ENTER";A$:OPEN "I",#-1,"FILE"
990 ON ERROR GOTO1040:FORKK=1536TO3071:INPUT #-1,K:IFK<>PEEK(KK)
 THEN PRINT"ERROR"
1000 NEXTKK:INPUT"CHECK COMPLETE PRESS ENTER";A$
1010 INPUT"CLEAR FILE (Y/N)";A$:IF A$="Y" THEN R4=1536:FORKK=153
6 TO 3071:POKE KK,0:NEXTKK
1020 CLOSE #-1:GOTO750
1030 PRINT"ERROR":FORKK=1TO3000:NEXT:GOTO750
1040 PRINT"ERROR":NEXTKK
1050 W2=1536
1060 KK=0:F0$="":F1$="":F2$="":F3$="":F4$="":F5$=""
1070 JJ=PEEK(W2):IFJJ=62 GOTO1180
1080 IFJJ=37 THEN KK=KK+1:GOTO1160
1090 IFJJ=60 GOTO1160
1100 IFKK=0 THEN F0$=F0$+CHR$(JJ)
1110 IFKK=1 THEN F1$=F1$+CHR$(JJ)
1120 IFKK=2 THEN F2$=F2$+CHR$(JJ)
1130 IFKK=3 THEN F3$=F3$+CHR$(JJ)
1140 IFKK=4 THEN F4$=F4$+CHR$(JJ)
1150 IFKK=5 THEN F5$=F5$+CHR$(JJ)
1160 W2=W2+1:IFW2>R4 GOTO750
1170 GOTO1070
1180 W2=W2+1:GOSUB670:PRINT@266,"FILE REVIEW":PRINT"C=";F3$;:PRI
NT TAB(15) "N=";F4$:PRINT"F=";F2$;:PRINT TAB(15) "X=";F0$;" ";F1
$:PRINT"?=";F5$:INPUT"Q=QUIT  P=PROCEED";X$
1190 IFX$="Q" GOTO750
1200 IFX$="P" GOTO1060
1210 GOTO1180
1220 PRINT@32,STRING$(32,"-"):PRINT@256,STRING$(32,"-"):PRINT@0,
"";:RETURN
1230 END
```

# Pascal-80

by J. B. Harrell, III

**Pascal-80 provides a serious version of the Pascal language to TRS-80 users. You might find its structured programming rewarding.**

★ ★ ★ ★

**Pascal-80**
**New Classics Software**
**239 Fox Hill Road**
**Denville, NJ 07834**
**Models I and III, 48K RAM, 1 disk drive**
**$99 disk**

Pascal-80 is not "just another compiler" but an honest attempt to provide TRS-80 users with a meaningful system for this language. It is a monitor system that provides an interface to the disk-file structure, a text editor, the Pascal compiler, and the Pascal interpreter. The system comes on a TDOS (tiny version of DOSPLUS) system disk in single-density, 35-track format for Model I users.

Users are not limited to this operating system; they can use Pascal-80 with any of the second-generation operating systems, such as NEWDOS80.

### Operation

The system is easy to operate; you type the command Pascal from the DOS-ready prompt and follow the displayed command menu.

All the system functions of Pascal-80 are available from the monitor, making it easy to save and load files, compile, execute, and edit source text from the monitor.

The compiler/editor system uses an in-memory text buffer with 23K of memory space that holds about 1,100 lines of code. The amount of source code that can be stored increases with use of text-compression techniques. Mr. Blank (one of the program's modifiers) has provided an Include compile-time option that lets you assemble a library of routines on the disk and compile only those routines you need by including them in your source program. This lets you create source programs of a size limited only by your system's memory capacity.

The in-memory compile feature also accounts for the compiler's incredible speed. When timed on a Model I running at standard speed (1.77 MHz), I got compile rates averaging 1,000 lines per minute when listing to the video screen and 1,500 lines per minute using the NOLIST option.

The system has some nice features available from the monitor to aid the user: The W command lets you write the compiled P-code to disk in a named file, and the X command lets you execute this compiled code directly from disk. This allows you to compile lengthy programs once in the final version and then run them directly without compilation delays. In addition, when executing the compiled code from the disk, you gain an additional 9K of memory, as the whole system, except the P-code interpreter, is overlayed by the compiled object code.

### Pascal-80 Editor

The Pascal-80 system has a full-screen editor that you can invoke from the monitor. Text displays on the video screen in a moving window of 15-line groups. By pressing the break key, the editor command menu displays at the bottom of the screen.

The editor uses a blinking underline character (the normal cursor character) that moves by pressing the arrow keys. Unlike many text editors, each line is a separate entity and cannot be extended past 64 characters. The text lines are stored in the text workspace only after pressing enter. This includes updating any changes made to the line after it is originally typed. As with all the keys, the arrow keys auto-repeat to move the cursor as far as you want. If you try to pass the upper or lower boundaries of the video screen, it will scroll in that direction for as long as you press the respective arrow key.

Cursor movement using the arrow keys is usually nondestructive; however, you can correct errors by backspacing over them using shift, left arrow. Or, you can use the text-overwrite mode and just type over the line. Remember, you must press enter to save any line changes.

The editor command menu gives a listing of the commands for editing the source program. Invoke each command by using the control key, shift, down arrow. This is a change from the original Pascal-80 that used break to preface all editor commands. Break now writes an editor command menu on the bottom line of the screen where it remains until you exit the editor or press clear.

The workspace stores the text using a text-compression algorithm that encodes one or more blank spaces into one byte. This produces a significant file-size reduction, but this file format is not compatible with any text editor currently in use. The logical length of each line, including encoded blanks, is always 64 bytes and the individual lines are not separated by an end-of-line character (0DH). Even though the revised editor available with Pascal-80 includes many of the necessary functions to efficiently edit text, it is still convenient to use a full text editor in many cases. Mr. Blank has provided conversion programs to convert text files to ASCII files and vice-versa.

**Pascal-80 Compiler**

The next major component in the Pascal-80 system is the compiler. This compiler generates a pseudo-code (commonly called a P-code) output in a one-pass compilation process. The P-code must be interpreted in order to be meaningful. The concept is based on compilers being able to operate on different machines and produce code that was transportable from one installation to the next. The P-code generated by Pascal-80 is not true P-code, but a compressed output requiring less space.

There are six compiler options to divert compiler output from the video screen to the line printer and to control information regarding the stack space, symbol table space, and actual code generated. The compiler output can be suppressed except for error messages. Two options allow all variables to be preset to zero upon allocation and perform a verify operation after disk file writes.

The compiler implements a TRS-80 version of the standard Pascal with some limitations and extensions. The reference manual with Pascal-80 does not describe the Pascal language and I suggest you buy *Programming in Pascal* by Peter Grogono (Revised Edition, Reading, MA Addison-Wesley, 1980) as a tutorial on the standard language. The following standard Pascal functions are *not* currently implemented in Pascal-80:

- Variant records, which record definitions that use the Case. . .Of variant structure to define multiple structures on the same data record.
- The With. . .Do statement.
- Pointer variables, including the New and Dispose statements.
- File window (buffer) variables, in-



See List of Advertisers on Page 451

| Test No. | Benchmark Test | Pascal-80 | Disk Basic | Fortran-IV | Tiny Pascal P-Code | Tiny Pascal Z80 |
|---|---|---|---|---|---|---|
| 1 | FOR/DO LOOP | 0.503ms | 1.872ms | 0.071ms | 0.942ms | 0.444ms |
| 2 | A = 1.0 | 0.885ms | 10.582ms<br>10.214msx | 0.549ms<br>0.289msx | ——— | ——— |
| 3 | IX = 1 | 0.460ms | 3.192ms | 0.019ms | 0.249ms | 0.107ms |
| 4 | A = A + 1.0 | 4.084ms | 14.191ms<br>12.454msx | 3.194ms<br>0.904msx | ——— | ——— |
| 5 | IX = IX + 1 | 0.956ms | 4.723ms | 0.039ms | 0.557ms | 0.162ms |
| 6 | IARRAY(I) = 0 | 2.743ms | 11.142ms | 0.066ms | 0.456ms | 0.213ms |
| 7 | J = K x L | 1.937ms | 6.041ms | 0.582ms | 0.804ms | 0.517ms |
| 8 | A = B x C | 9.979ms | 11.206ms<br>6.438msx | 5.869ms<br>1.686msx | ——— | ——— |
| 9 | POKE White Out | 1.24sec | 7.24sec | 0.122sec + | 1.33sec | 0.70sec |
| 10 | SET White Out | 9.47sec | 44.53sec | ——— | 17.75sec | 6.97sec |

*Table 1. Benchmark Comparisons. The For. . . Do loop benchmark test was timed for 10,000 executions of the loop and the times displayed are the average values for one execution of the loop. Each of the benchmark times are listed in milliseconds, unless otherwise noted. For each of tests 2–8, the statement listed was timed for 10,000 executions in a For. . . Do loop and the average execution time for the statement is calculated from the difference in execution times for the For. . . Do loop with and without the statement. The tests were run in single- and double-precision floating-point arithmetic using Basic and Fortran to demonstrate the speed difference in the two arithmetic modes. Pascal-80 uses only the double-precision (14 digits) mode.*

x The time listed is for execution of the statement using single-precision floating point values.

\+ This method executed so fast that it was necessary to time the execution of the program for an average over 100 executions.

cluding the Get and Put procedures.

- The procedures Pack and Unpack (all structures are already based on allocation on byte boundaries).
- The Page procedure (but Write (LP,CHR(12)) is a good substitute).
- File structures (i.e., Array of File).

Sets are limited to 256 members, and if numeric, the set members must be in the range of 0–255. The identifier of a procedure or a function cannot be passed as a parameter to another procedure or function. No expression passed as a value parameter can exceed 510 bytes unless passed as a VAR parameter. Integer variables used to reference array elements in a record must be global variables. Local variables will not work.

The character [ can be replaced with (. and ] replaced with .) on the Model I. The Model III has these characters and generates them on the display. Even though the Model I user will see the characters as up arrow and left arrow respectively, these characters will still compile and print correctly. Spaces are significant in some isolated Pascal-80 situations where they would not be significant in standard Pascal. Mr. Blank is currently working on a release to implement these features.

Many extensions have been added to this newer version of Pascal-80 and are listed below.

- Arrays of characters can now be printed with a single statement.
- The Read and Write procedures can now be used with nontext files in place of the Get and Put procedures.
- When using string constants in assignment statements and in comparisons with character arrays, the constant on the right can be shorter than the item on the left and will be padded with blanks as necessary. If String is type ARRAY[1..10] OF CHAR then STRING: = 'NAME' is valid and STRING>'NA' is True. (The right argument must have at least two characters to be a string and not a CHAR.)
- PROC and FUNC are abbreviations for the keywords procedure and function.
- Predefined constants are MININT (−32,768), MAXINT (32,767), PI, true, and false.
- Real variables have 14 digits of precision and occupy eight bytes. The data type REAL6 is provided if this much significance is not needed and occupies four bytes. You save no time as all calculations are done in double precision. REAL6 variables cannot be passed to a procedure or function as a value parameter unless they are members of an array or record.
- No files need be declared in the program header. In addition, the program name is optional. There are three default files (INPUT, OUTPUT, LP) that are declared as File of Text.

The following procedures have been added to the standard Pascal procedures/functions:

- Call (address,value)—a type integer function that places the eight-bit value in the A register and calls the routine located at address. The eight-bit value in the A register on return is the function value.
- Close—closes disk files
- CLS—clears the screen
- EX (real-expression)—a type integer function that returns the exponent value of the real-expression.
- FP (real-expression)—a type real function that returns the mantissa of the real-expression as its function value.
- INKEY—a function returning the



See List of Advertisers on Page 451

value of the keyboard at the call in CHAR format. This function is identical to the Basic INKEY$ function.

- MEM—an integer function that returns a value corresponding to the amount of memory remaining.
- PEEK (address)—an integer function that returns the eight-bit value from memory at *address*.
- POKE (address,value)—normally, this is a procedure to place the eight-bit value at the address specified; however, the procedure has been extended. The address and value parameters are interpreted as the x and y coordinates of the graphic pixel, and if address is in the range of 0–127, then a Set operation is performed. If address ranges from 128–255, then a Reset is performed, and if address is in the range of 256 to 383, then a Point is performed with the result left in location 21458 that can be accessed with a PEEK (0 if the point is off and nonzero otherwise).
- Seek (expression,file name)—positions the specified file name to the record in the file pointed to by the integer value of expression.

A compile-time function is recognized to let you include source code from a disk file in the program currently being compiled. This is done by placing the compiler command (*$ file name *) in the source program where the included code is to be inserted from the disk. This is a powerful addition to the original Pascal-80. The Case statement has two extensions from standard Pascal. An optional Else clause will be executed if no other case condition is satisfied, and if no satisfied case condition and no Else clause exists, execution passes to the next statement without generating an error message.

Both integer and real values are displayed using the same format descriptors, for example, Write (expression: fieldwidth:digits). A field width of zero produces a display in the default format for the displayed expression type. Similarly, a field width of −1 produces the display of the expression in scientific notation.

In addition to the functions specified as part of the standard Pascal language and the extensions just covered, Mr. Blank has added the functions for random numbers. The Basic functions of RND(0) and RND(number) are coded in Pascal to provide linkage to the run-time package and are stored on the system disk so they can be included in any program. The POKE command accesses the graphic capability of the TRS-80.

Pascal-80 provides an additional source-code procedure (GOTOXY) on disk to let you position the video system cursor to any screen location. The call GOTOXY (horizontal-pos,vertical-pos) specifies the cursor's location.

**Additional Features**

The Pascal-80 master disk contains the compiler system and several other files. These files contain the Pascal source code for graphics, random-number, and GOTOXY linkages, plus several demonstration Pascal programs. Each program is designed to illustrate a feature of the Pascal language.

In addition to the demonstration programs, there are two utilities to convert Pascal-80 text files to ASCII character files (ASCII/CMD) and to convert from character format files to text format (TEXT/CMD). The ASCII files can be used with almost any word processor. I have had no problem using Newscript to read the files, but Electric Pencil needs a 00H byte at the current end-of-file prior to reading it with Pencil.

The two remaining files (AUTHOR/SRC and AUTHCODE/CMD) allow you to write an application program in Pascal, compile it and sell it as a CMD file that executes directly from the disk operating system command level. New Classics Software grants a license to original registered owners of Pascal-80 to distribute programs compiled by Pascal-80, provided you meet the requirements specified in the user's manual. A minimal program (BEGIN END.) compiled and converted to a CMD file requires nine granules of disk space due to the P-code's interpreter.

**Pascal-80's Performance**

I performed benchmark comparisons using similar programs written in Pascal-80, Disk Basic, and Fortran. Table 1 reports the results of the comparisons. The first eight tests were conducted using a simple For-loop executed 10,000 times. After determining the time for the For-loop, the times for each of the remaining tests were calculated. Test number 9 is a simple For-loop to white out the video screen by POKEing the value 191 (BFH) from locations 15360 to 16383. Test number 10 is a nested For-loop to white out the video screen by use of the Set(x,y) command.

Pascal-80's performance is outstanding considering the interpretive nature of its execution. From observing the data in test 8 (and similar data for the computation A = B + C), it is obvious that the 14-digit precision of Pascal-80's floating-point arithmetic can make the program almost as slow as an identical Basic program if using extensive floating-point arithmetic. However, the converse statement that single-precision Basic can run faster is not necessarily true and you'll have to use many floating-point calculations to overcome the strength of the compiled source program.

The data in tests two and four show that the Basic times are only slightly different and both single- and double-precision calculations are significantly slower than the corresponding double-precision statement executed in Pascal-80. This is because Basic must interpret each component of each statement every time it is executed and each constant must be converted from character format to binary every time encountered. Basic is spending most of its time in the conversion routines.

The initial benchmark comparison yielded a surprise when test 10 was timed. Pascal-80 was slower than Basic for the screen white-out using the Set command. This was due to the addition of the graphics commands as an afterthought and the use of Pascal-80 as a software linkage to these high-performance routines. I immediately received an upgraded system implementing these commands as a special option of the POKE statement.

The comparison to Tiny Pascal was included to demonstrate that the far more complex implementation of Pascal-80 does not suffer from a slowdown caused by size growth. Remember, Tiny Pascal is only a very limited subset of the total language.

### Conclusion

I have tried to portray Pascal-80 as the first real attempt to provide useful implementation of TRS-80 Pascal for a reasonable price. You should be using a Z80 compiler such as Fortran if speed is your major objective. But if ease in programming using this well-structured language is your objective, then Pascal-80 is perfect.

The ability to write, edit, compile, and debug your program in a friendly, interactive environment is a great instructional tool. ■

*John Harrell can be reached at 30 Keklico Court West, Charleston, SC 29408.*

✓ See List of Advertisers on Page 451

# TRS-80 Tongues

by Robert Nicholas and Philip Martel

**Can't decide which language to use? Here's a simple temperature conversion program in six: Basic, Fortran, Cobol, Pascal, APL, and Forth.**

Why buy another language when you already have Basic in your TRS-80? It all depends on what you want your programs to do. Let's consider six languages from the perspective of one simple problem—a Fahrenheit-to-Celsius conversion. The examples will give you a feel for the languages' similarities and differences.

### Basic

Nearly every home computer comes with Basic (beginners' all-purpose symbolic instruction code) built-in. The average newcomer to computing can sit down with an introductory manual and begin writing simple programs in Basic within a few hours. Power!

Basic is rather forgiving, and more or less conversational. While the more powerful commands of a modern extended Basic appear strange, the core commands of the language are everyday words: GOTO, Return, If, Then, Else, And, Or, Let, Input, Print, Random, Read, Write, Run, Stop, New, +, −, * (multiply), and / (divide).

For instance, look at Program Listing 1. This routine accepts a temperature in Fahrenheit degrees and converts it to Celsius via the formula C = 5/9 (F −32).

The first three program lines are remark statements, indicated by the Basic command REM. The programmer uses these lines as notes—they don't mean anything to the computer.

The CLS in line 40 tells the computer to clear the screen.

Line 50 tells the computer to print the phrase "Degrees Fahrenheit" at position 256 on the screen. (The TRS-80 has 1,024 positions on the screen, broken down into 16 lines of 64 characters.)

Line 60 waits for you to enter the degrees in Fahrenheit you want converted to Celsius. Your entry will be stored in the computer as the variable F.

Line 70 provides a means of escaping the program. If you enter the number -9999.99, the program ends. If the value of F is anything other than -9999.99, the program continues with line 80, which clears the screen again.

```
10 REM BASIC PROGRAM TO CONVERT FAHRENHEIT TO CELSIUS
20 REM           BOB NICHOLAS
30 REM ENTERING A VALUE OF -9999.99 ENDS THE PROGRAM
40 CLS
50 PRINT@ 256, "DEGREES FAHRENHEIT ";
60 INPUT F
70 IF F = -9999.99 THEN END
80 CLS
90 C = (F - 32) * 5/9
100 PRINT@ 384, F;  " DEGREES FAHRENHEIT = ";  C; "  DEGREES
CELSIUS"
110 GOTO 50
```

*Program Listing 1. Basic Version*

**The Key Box**

**Model I or III**
**Programming Languages**

Line 90 calculates the Celsius equivalent to F.

Now line 100 prints your value and the Celsius value on the screen at position 384. Line 110 tells the computer to go to line 50, where the program asks you for another Fahrenheit value to convert.

You can enter whole numbers, decimal numbers, and positive and negative numbers into this program. All we said was Input F—we didn't have to specify that F could be either positive or negative, or how many decimal places we were going to use. Well, that's not true of all computer languages.

**Fortran**

Fortran, an acronym for formula translation, is similar to Basic. However, the input and output from Fortran generally require Format statements (something like Basic's Print Using). In addition, while Basic is usually interpreted, Fortran is almost always compiled. A compiled program runs faster than one that is interpreted.

Lines 100 and 200 (see Listing 2) start with a C. They are comments—remarks.

Lines 300 and 400 display "Type in Fahrenheit Value" on the screen. The first column holds Editor line numbers, not used by the Fortran program. To the right are Fortran line numbers, and they may be referred to in the program. Line 300 says write to the screen (logical unit number—LUN1) what's in line 15. Line 400 gives directions to skip a line (/), write the text in single quotes, and skip another line.

Lines 500 and 700 cause the program to read the value F from the keyboard (which is also referred to by the LUN 1) in a format (F15.0) that allows, but does not require, a decimal point. Line 600 goes to line 40 if F is less than (.LT.)-500. It could have been put after line 700, but we wanted to show the format statement separated from the I/O statement.

One style of Fortran programming places all format statements at the end of the program. This is useful if a particular format is used many times; it can be written once and found easily.

Line 800 is the actual computation. Notice the decimal points. The distinction between integers and real numbers is much stronger in Fortran than in Basic. If you want a constant treated as a real number, you have to put in the decimal point.

Real numbers, as a computer thinks of them, are numbers that can have fractional parts. Integers do not have fractional parts. In Basic, if you want to



See List of Advertisers on Page 451

```
00100  C  PROGRAM TO CONVERT FAHRENHEIT TO CELSIUS
00200  C  BY PHIL MARTEL
00300  10        WRITE(1,15)
00400  15        FORMAT(/' TYPE IN THE FAHRENHEIT VALUE'/)
00500            READ(1,20)F
00600            IF (F .LT. -500.) GOTO 200
00700  20        FORMAT(F15.0)
00800            C=5./9.*(F-32.)
00900            WRITE(1,30)F,C
01000  30        FORMAT(1X,G10.3,'  DEGREES FAHRENHEIT IS '
01100          1 ,G10.3,' DEGREES CELSIUS')
01200            GOTO 10
01300  200       STOP DONE
01400            END
```

*Program Listing 2. Fortran Version*

```
000100 IDENTIFICATION DIVISION.
000110 PROGRAM-ID.      FAHRENHEIT.
000120 AUTHOR.          BOB NICHOLAS.
000130*****************************************************
000140*  THIS PROGRAM ACCEPTS A TEMPERATURE IN FAHRENHEIT *
000150*  DEGREES FROM -9999.98 TO 9999.99 AND CONVERTS IT *
000160*  TO CELSIUS DEGREES.                              *
000170*  A VALUE OF -9999.99 ENDS THE PROGRAM.            *
000180*****************************************************
000190
000200 ENVIRONMENT DIVISION.
000210 CONFIGURATION SECTION.
000220 SOURCE-COMPUTER.    TRS80-I.
000230 OBJECT-COMPUTER.    TRS80-I.
000240
000250 DATA DIVISION.
000260 WORKING-STORAGE SECTION.
000270 77   FAHRENHEIT   PIC S9(4)V99.
000280 77   CELSIUS      PIC S9(4)V99.
000290 77   QUESTION     PIC ZZZ9.99-.
000300 77   ANSWER       PIC ZZZ9.99-.
000310
000320 PROCEDURE DIVISION.
000330 10-MAINLINE.
000340      DISPLAY " " ERASE.
000350        PERFORM  20-CONVERSION   UNTIL   FAHRENHEIT   =
-9999.99.
000360      STOP RUN.
000370
000380 20-CONVERSION.
000390      DISPLAY "DEGREES FAHRENHEIT" LINE 5 POSITION 2.
000400      ACCEPT FAHRENHEIT LINE 5 POSITION 28
000410          SIZE 8 PROMPT CONVERT.
000420      COMPUTE CELSIUS = FAHRENHEIT - 32) * 5 / 9.
000430      MOVE FAHRENHEIT TO QUESTION.
000440      MOVE CELSIUS TO ANSWER.
000450      DISPLAY QUESTION LINE 07 POSITION 02 ERASE,
000460          " DEGREES FAHRENHEIT = ",
000470          ANSWER, " DEGREES CELSIUS".
000480 END PROGRAM.
```

*Program Listing 3. Cobol Version*

use integers you have to use DEFINT or place a % after the variable name. In Fortran, a variable that starts with the letters I through N or a number without a decimal point is assumed to be an integer.

Line 900 outputs the Fahrenheit and Celsius values. Line 1000 is the Format statement for line 900. The 1X specifies that the first character is a blank. This is needed because the first character output is a printer control character. Certain nonblanks can produce odd results. The G10.3 outputs a number with 10 spaces allocated, up to three of which are after the decimal point.

Line 1100 is a continuation of line 1000. The & in column 6 indicates a continued line. Anything but a zero or blank will do.

Line 1300 stops execution of the program, prints Stop Done and returns to DOS. Line 1400 tells the compiler to stop. Unlike Basic, where an End statement can occur anywhere, in Fortran an End must only be the last statement in the program.

## Cobol

Cobol (common business-oriented language) programs tend to be lengthy. The language demands a certain amount of structure—every Cobol program (see Listing 3) consists of four blocks referred to as divisions. These are: Identification division (lines 100–180); Environment division (lines 200–230); Data division (lines 250–300); and Procedure division (lines 320–480).

The Identification division indicates the name of the program (line 110) and the author (line 120). You can also indicate the date the program was written and the date it was compiled. Generally, the only thing required by Cobol is the Program ID. Each of these lines is referred to as a "paragraph."

Lines 130–180 are remarks, indicated to the editor by an asterisk in the first column following the line number. As in Fortran, the line numbers are for editing purposes only. Notice also that lines (sentences) end in periods.

The Environment division is broken down into sections. The Configuration section is broken down into two paragraphs that record the source computer (the computer on which the program is to be compiled) and the object computer (the computer on which the program will be run). In addition, if the program contains any files to be processed, they must be spelled out in the Input/Output section.

The Data division is broken down into three possible sections: File section, Working-storage section and Linkage section. Only the Working-storage section appears in this program. Every variable used in the execution of the program must be specifically mentioned in one of these three sections. If the variable is one of the fields in a file, it would appear in the File section. If it is a non-file variable used in the process of executing the program, it would be in the Working-storage section. And if this were a subroutine called from another computer program, the variables being passed back and forth would have to be in the Linkage section.

This may seem tedious compared to the convenience of Basic, but it provides the programmer far greater control over the information being processed. Additionally, whereas Basic variables can be two characters long and Fortran variables can be six, Cobol variables can be up to 30 characters long. This makes reading and editing Cobol programs easier.

In this program, we have four variables defined in the Working-storage

See List of Advertisers on Page 451

```
(* 10*)(* FAHRENHEIT TO CELSIUS CONVERSION IN TINY PASCAL
*)
(* 20*)(* BY PHIL MARTEL *)
(* 30*)VAR F,C,CINT,CFRAC,FLAG : INTEGER;
(* 40*)BEGIN
(* 50*)    FLAG := 1;
(* 60*)    WHILE FLAG =1 DO
(* 70*)    BEGIN
(* 80*)       WRITE('TEMPERATURE IN FAHRENHEIT');
(* 90*)       READ(F#);
(* 100*)      IF F<= -500 THEN FLAG := 0
(* 110*)      ELSE BEGIN
(* 120*)        C := (F-32)*5;
(* 130*)        CINT := C DIV 9;
(* 140*)        CFRAC :=(ABS(C) MOD 9)*10 DIV 9;
(* 150*)        WRITE(F#,'DEGREES FAHRENHEIT IS
',CINT#,'.',CFRAC#,
(* 160*)              ' DEGREES CELSIUS',13);
(* 170*)      END  (* FOR THE CASE WHERE F > -500 *)
(* 180*)    END (* OF THE WHILE LOOP *)
(* 190*)END.
```

*Program Listing 4. Pascal Version*

```
0:     CLS
1:     # 28 31


0:     FTOC
1:     CLS
2:     BEGIN  :'ENTER DEGREES FAHRENHEIT ';
3:     F←q
4:     CLS
5:     C←(F - 32) x 5 % 9
6:     F ;'DEGREES FAHRENHEIT = ';C ;'DEGREES CELSIUS'
7:     →BEGIN
```

*Program Listing 5. APL Version*

section: Fahrenheit, Celsius, Question, and Answer. The 77 in front of them is a level number. The 77 level can appear in either the Working-storage or Linkage sections. It is used for variables that are not subdivided. Subdivision is one of the main advantages of Cobol. It provides the programmer with easy reference to a variable and to any predefined subsections of it. A subdivision is shown by a 01 level number with sublevels. For instance:

```
01 NAME.
   03 LAST-NAME      PIC X(15).
   03 FIRST-NAME     PIC X(12).
   03 INITIAL        PIC X(1).
```

This gives us a 28-character field called Name, a 15-character field called Last Name, a 12-character First Name and a one-character middle Initial. The X(15) specifies a field that will accept all keyboard characters. A( ) is used to specify alphabetic characters only, while 9( ) specifies numeric only.

Fahrenheit has a picture of S9(4)V99. The S means the value can be negative. If the S is left off, -25 and 25 both become positive 25. The 9(4) means that up to four digits may follow. V is an assumed decimal point. And the 99 allows up to two digits following the decimal point. The result is that Fahrenheit must be between -9999.99 and 9999.99. If you enter 25, the V assumed decimal point will convert it to 25.00.

Celsius has a similar picture. Fahrenheit and Celsius must have assumed decimal points (the V) because they are used in calculations. However, when we want to show these values on the screen or on paper, we have to include the decimal point in the picture. Hence, we have two new variables.

Question will be used in printing the value entered for Fahrenheit. The picture clause is ZZZ9.99-. The Zs suppress leading zeros in the answer so that 0003.45 appears as 3.45. The dash means that we will have any negative sign printed after the number (3.45-).

Answer has a similar picture and is used in printing the value of Celsius on the screen. All these pictures are the equivalent of the Format statements in Fortran.

It is not at all unusual for the Data division to be the longest part of a Cobol program. Every variable used has to be carefully and explicitly defined.

Suppose we want to express the equivalent of our Cobol line 270 FAHRENHEIT PIC S9(4)V99 in Basic. This would require the following additions to our Basic program following line 60 INPUT F:

```
61 IF F < -9999.99 OR
F > 9999.99 THEN GOTO 60
62 LET X = INT (X * 100) / 100
```

Line 61 assures us that F is within the proper range. Lines 62 and 63 detect if there are more than two digits following the decimal point. If there are, they get knocked off. And even this doesn't finish the task. Remember that if we enter 25 for the value of Fahrenheit, we will get 25.00. The Basic program would still print plain old 25. If we wanted the decimal format, we would have to use a Print Using statement in Basic. So if you are looking for this type of control in a business or commercial application, Cobol is by far preferable to Basic.

And finally, on to the Procedure division—the part of the program that actually does something. The Procedure division is broken down into paragraphs. This program contains two paragraphs in this division: 10-MAINLINE and 20-CONVERSION. All lines in a paragraph must be indented at least four spaces. Further indentation (such as lines 410, 460, and 470) is optional and strongly recommended.

While Cobol is a fairly standardized language, those commands relating to screen output tend to vary from computer to computer and compiler to compiler. As a result, some references within the Procedure division detract from the normal high degree of readability of the language. For instance, line 340 DISPLAY "" ERASE is the equivalent of CLS in Basic—it clears the screen.

If you glance through the rest of the Procedure division, you will notice there are no GOTOs even though they are available in Cobol. Good Cobol tends to be GOTOless programming. The program is written as a series of modules organized into paragraphs or sections made up of paragraphs. When you need to use a given module, you perform it, as in line 350.

```
PERFORM 20-CONVERSION
UNTIL FAHRENHEIT = -9999.99.
```

Paragraph 20-Conversion will run continually (like a subroutine) until we enter -9999.99 as the value of Fahrenheit. Then execution drops to line 360, and the program stops. Note that in Cobol we perform or GOTO paragraphs rather than line numbers as in Basic. Big

★ **FREE SHIPPING** ★
WITHIN CONTINENTAL 48 STATES

MICRO IMAGES

# ARE YOU A DREAMER?

## IMAGINE THIS . . .

*You place your order and it arrives when expected. "What do you know, they did ship that day!" You open your package and SURPRISE, it's what you ordered, not last year's version six times removed. "Wait a minute, I must have paid full retail to get this kind of service. H'mm, that's not it. These prices are among the lowest. What's the catch? I've got it! They charged me large shipping and handling charges. No, not that either. It says here 'Free shipping within the Continental 48 States via UPS ground.' Only the differential is charged for UPS Blue or 1st Class. Now I've got it, it's only a dream!!!"*

*TRY US!*

At Micro Images your dreams become reality! How do we do it? Simple, we tell the truth. If the item you order is not in stock — we tell you. If we can't ship that day — we tell you. Which version? We tell you. Why are we telling you this? The answer is easy. This is what we do best and what separates us from the competition, AND WE WANT YOU TO KNOW.

**LAZYWRITER**
Latest Version Mod I/III $159.95
See Below For New Options

**MAXI MANAGER**
Manager with Utility - $119.95
Maxi Utility Only - $44.95

**SUPERUTILITY +**
Mod I/III - Limited Time Only - $46.95
SEE NEW BOOK BELOW

**MZAL** - Ver. 2
Mdl I or III — $134.95

**MAXI CRAS**
Mdl I/III — $84.95
MAIL $84.95 STAT $179.95

**LDOS** 5.1
MDL I or III — $114.95

**GEAP** - $42.95
with Dot Writer - $69.95

**NEW SCRIPT** - 7.0 - $114.95
with Mailing Label Opt. - $124.95
Mailing Label Opt. Only - $27.95

**New Random House Version**
Proofreader by Aspen ..... Mod I/III $49.00
Proof-Edit by Aspen ..... Mod I/III $25.00
Grammatik by Aspen ..... Mod I/III $49.00
*SPECIAL . . . All 3 FOR ONLY $110.00*

**MULTIDOS** - $74.95
Specify Mdl III or Mdl I
Single or Double Density

## DOSPLUS
Version 3.4S/3.4D/3.4III ....... $119.95
Version II For Model II ....... $199.95

**UNITERM** Mod I/III ....... $74.95
**UNITERM/80** Mod I/III ....... $84.95

**MICROSOFT**
Fortran 80 or A.L.D.S. ......... $89.95
Basic Compiler ............... $179.95
Editor/Assembler + Tape $27.95 Disk $45.95

**HEXSPELL 2** - Mod I/III $94.95

**POSTMAN MASS MAILING SYS.**
Standard Ver. Mod I/III ......... $118.95
with Postwriter Mod I/III ........ $159.95

**DATA-WRITER** Mod I/III $114.95

**SFINKS 3.0 Disk** Mod I/III .. $36.95
**SFINKS** Chess Tutor Mod I/III $17.95

**EDAS** Mod II 4.0 ............ $74.95
Mod I/III 3.5 ............ $69.95

**LAZYWRITER OPTIONS**
LAZYDOC .................... $54.95
LAZY TAB .................... $13.95
LAZYCALC .................... $27.95
LAZYDRAW + DO ............ $18.95
Tab. Calc. Draw + Do *All 4* $37.95

## BOOKS
Disk + Other Mysteries — IJG .... $20.95
Basic Decoded — IJG ............ $27.95
Custom TRS-80 — IJG ............ $27.95
Basic Faster & Better — IJG ...... $27.95
Mdl II/16 Visicalc — W C Brown ..... $16.95
Mdl I/III Visicalc — W C Brown ...... $16.95
Copyright Kit .................... $11.95
Inside Superutility Plus .......... $17.95

**LNW-Doubler 5/8** $205.95
Includes Dosplus 3.4D

**SOOPER SPOOLER** - by Compulink
16K Parallel I/O Unit ..................... $334.95
46K Memory Opt - $149.95 Serial I/O Opt. - $89.95
Complete Unit - All Options ............... $549.95

**LYNX** auto dial/answer Mod I/III $239.95

★ ★ **PRINTER STANDS** ★ ★

**SPACE AGE**
MX 80 Clear ..... $27.50
MX 80 Bronze .... $29.95

**NO FRILLS**
MX 80 Clear ..... $13.50
MX 80 Bronze .... $15.95

Microline 82A Space Age Bronze ........... $29.95

★ **VISION FRIENDLY** ★
Vision Friendly is an eye exercise package for the Model III which comes with a Green, Blue, and Amber screen and a short graphic program. The package combines color & graphics to relax your eyes and ease eye strain. Each screen is a full grade quality screen which needs no glue, tape, etc. to attach and can be removed without a trace. The three plexiglass screens are more than worth the price. We are so sure you will like this package we are offering a **10 day money back guarantee**. $29.95

**FREEDOM TECHNOLOGY INT'L**
Freedom Option ................ $199.95
Freedom Plus ................. $399.95

**MICROBUFFER** - Practical Peripherals
Parallel or Serial (Epson) .......... $149.95

## RIBBONS

| ZIP BOX RELOADS | ½ Dz. | Dz. |
|---|---|---|
| Epson MX 70/80-20 Yds | 24.00 | 42.00 |
| Epson MX 100-30 Yds | 30.00 | 52.00 |
| LP-I/II/IV-16 Yds | 18.00 | 32.00 |
| Centronics 730/737/739 and 779-16 Yds | 18.00 | 32.00 |

All ZIP BOXES are individually sealed black nylon and require no rewinding

| CARTRIDGES | Each | Dozen |
|---|---|---|
| Epson MX70/80 | 8.95 | 90.00 |
| LP-III/V | 6.50 | 70.00 |
| Centronic 702/03/04/53 | 11.00 | 120.00 |
| RS DSY WHL II Multi Strike | 6.50 | 70.00 |
| Diablo Hytype II Multi Strike | 6.50 | 70.00 |
| Qume-300,000 chr Multi Strike | 6.50 | 70.00 |
| Nec SPIN H-Yield Multi Strike | 7.00 | 75.00 |
| MICRLNE 80/82A/83A Spl | N/A | 24.00 |

Minimum order 3 cartridges - any mix. For smaller quantities add $1.50 per order. All our reloads and cartridges are manufactured by one of the oldest and most reputable ribbon Mfg's in the country

★ ★ ★ ★ ★ **QUALITY GUARANTEED** ★ ★ ★ ★ ★

## GAMES

| | Tape | Disk |
|---|---|---|
| Sea Dragon | 17.95 | 22.45 |
| Penetrator | 22.45 | 22.45 |
| Defiance | | 22.45 |
| Panik | 17.95 | 22.45 |
| Armored Patrol | 17.95 | 22.45 |
| Eliminator | 17.95 | 22.45 |
| Starfighter | 22.45 | 26.95 |
| Forbidden Planet or City | | 34.95 |
| Strike Force | 14.50 | 17.95 |
| Voyage of Valkyrie | 31.95 | 34.95 |
| Bounceoids or Sp. Castle | 14.50 | 17.95 |
| The Institute | 17.95 | 20.95 |
| Fortress or Alien Def | 14.50 | 17.95 |
| Any Big 5 Program | 14.50 | 17.95 |

TO ORDER CALL OR WRITE TO:

*MICRO IMAGES*
146-03 25th Road
Flushing, N.Y. 11354
(212) 445-7124 ✓157

PHONE HOURS:
Mon.-Fri.
10 AM-9 PM
Sat.
10 AM-5 PM

**ORDERING INFORMATION**
No credit cards at these low prices. Add $2.00 on all COD orders. Certified Ck/MO/COD shipped immediately. Please allow 2 weeks for personal checks. For extra fast service phone in your COD order. Free shipping within Continental 48 states via UPS ground. For Canada, Hawaii, Alaska, applicable shipping and insurance charges apply. Prices subject to change without notice. New York State residents please add appropriate sales tax.

The items listed above are a cross-section of our product line. We carry the full line of most companies listed in this ad, plus much more. **SEND FOR YOUR FREE CATALOG.**

✓See List of Advertisers on Page 451

deal? You bet! How many times have you gone crazy trying to find the line number for your Basic subroutine or GOTO after renumbering a program? This problem doesn't exist in Cobol.

Now for the conversion and display portion of the program—line 390 prints Degrees Fahrenheit on the screen. Note that you get to state the screen position as a line number and position (column) number—much easier to locate than the Basic PRINT@ value.

Lines 400 and 410 are the equivalent of Input F. Note again that we get to state the screen position for the input. We set a size of eight, because our picture for Fahrenheit in Data division allows eight positions: one for the sign, four digits, a decimal point, and two more digits. The word Prompt tells the computer to put hyphens on the screen showing how many places are available for data entry (--------). Finally, Convert tells the computer to store whatever is entered on the screen (string information) in Fahrenheit as a numeric value.

Line 420 does the actual conversion from Fahrenheit to Celsius. We Compute the value of Celsius.

Line 430 moves the value stored in Fahrenheit into Question for printout on the screen with the assumed decimal point in Fahrenheit (line 270) shown as indicated in the picture of Question (line 290). Line 440 does the same thing for Celsius.

Lines 450–470 format the screen printout of the input and the answer. Notice again that we get to specify the line and the position. The Erase statement clears the screen before the response is printed.

Line 480 (End Program) is required by the Cobol compiler, like Fortran.

**Pascal**

Radio Shack's Tiny Pascal lacks many features of a full Pascal, especially in the variety of data types—files, records, pointers, real numbers, and user-defined types. These data types are not needed, but they are a convenience and make programs more self-documenting. We used the method described in John Harrell's article "Modifying Tiny Pascal for Disk" (*80 Micro*, June 1981, p. 154) to put the Tiny Pascal system and source programs onto disk.

Pascal, unlike Basic or Radio Shack's Fortran, does not use line numbers. As a convenience, we have put numbers in comments (*like this*) at the beginning of each line.

Tiny Pascal compiles into an intermediate code called P-code, which is then interpreted. But for all practical purposes it is a compiler like Fortran.

Lines 10 and 20 (see Program Listing 4) are comments. The variables are defined in line 30. In Pascal, all variables must be defined at the start of the program or subroutine where they are used. This makes you more aware of the variables you are using and also allows the compiler to run faster.

Line 40 starts a block. All Pascal programs are composed of blocks, each of which begins with Begin and ends with (you guessed it) End. The main block ends at line 150. The period after End at line 150 tells the compiler it is the end of the program.

Line 50 sets the variable Flag for the While loop in line 60. While Flag equals one, it repeats the block in lines 70–180.

Line 80 asks for the Fahrenheit temperature and line 90 reads it in. The # in line 80 tells the system the input will be a decimal integer.

Line 100 tests the Fahrenheit value to see if it is less than or equal to -500. If it is, the flag is set to zero to end the While loop. If not, then the block in lines 110–170 is entered.

Lines 120–140 calculate CINT, the integer part of the temperature, and CFRAC, one digit after the decimal point. Lines 150 and 160 print the results. Again, # indicates that a decimal integer should be printed. The 13 at the end does not have a #, so it is treated as a character, in fact a carriage return.

```
( FAHRENHEIT TO CELSIUS CONVERSION PROGRAMS)
: FC 32 - 50 * 9 / ; ( REPLACES F WITH 10 * C )
: FTOC DUP . " DEGREES FAHRENHEIT IS "
   FC DUP ABS <# # 46 HOLD #S SIGN #> TYPE
   "  DEGREES CELSIUS " ;
: FCPROG BEGIN CR " TYPE IN FAHRENHEIT VALUE " CR #IN
   FTOC 0 END ;
```

*Program Listing 6. Forth Version*

**APL**

APL (a programming language) is quite different from all the other languages discussed in this article. For one thing, it is essentially mathematical. Where most languages work with one-dimensional values (scalars), APL can deal with multidimensional blocks of data (matrices) just as handily.

APL does not use standard algebraic notation for its calculations—there is no hierarchy of operations as in algebra. A mathematical expression is simply evaluated from right to left. You can force order if you wish by using parentheses. The reason for this unusual feature is that APL has many operators besides the usual +, −, *, and / of Basic or Fortran. Defining a reasonable hierarchy and keeping track of it would be very difficult, so Dr. Iverson, the developer of APL, avoided the issue.

Another major difference lies in the structure of a program. APL deals with a workspace rather than a program. Each workspace is composed of labeled functions that call one another as required. You execute one function from another merely by naming it.

For instance, in the APL workspace shown in Program Listing 5, we have two functions: CLS and FTOC. We initially enter the workspace through FTOC. The first thing FTOC does is execute the CLS function.

The CLS function has one command: # 28 31. The # is the equivalent of CHR$ in Basic, so this line performs CHR$(28) and then CHR$(31)—it homes the cursor to the upper left corner of the screen and clears the screen from the cursor to the end. The result is a completely cleared video screen.

This brings us back to line 2 of FTOC. We label this line as Begin (we will want to come back to this spot later in the program) and then print "Enter Degrees Fahrenheit." (Note that you don't have to say Print. All output goes to the screen by default.)

Look at line 3. The small letter q stands for the APL command Quad. This equivalent of the Basic command Input is for numeric input only. Quote Quad is used for entering string information. In APL, line 3 would be phrased as ASSIGN F QUAD. (Assign—shown by the back arrow—is the equivalent of the equals sign in Basic.) You aren't limited to just typing in a number in APL. You could enter a mathematical expression (such as 5 + 3) or even a variable name (C, for example).

# WE WILL NOT BE UNDERSOLD

## SOFTWARE

| | |
|---|---|
| MICROPRO | |
| Wordstar | $239.00 |
| Mailmerge | $ 89.00 |
| Customization Notes | $359.00 |
| Spellstar | $239.00 |
| Datastar | $199.00 |
| Calcstar | $199.00 |
| MICROSOFT | |
| Basic Interpreter | $349.00 |
| Basic Compiler | $389.00 |
| Fortran 80 | $499.00 |
| Cobol 80 | $695.00 |
| DATA BASE | |
| FMS80 | $595.00 |
| dBase II | $599.00 |

## NEW! IBM PC SOFTWARE NEW!

| | |
|---|---|
| INFORMATION UNLIMITED | |
| Easy Writer | $289.00 |
| Easy Speller | $149.00 |
| Easy Filer | $319.00 |
| VISICORP | |
| Visicalc 256K | $199.00 |
| VisiDex | $199.00 |
| VisiFile | $229.00 |
| VisiTrend/VisiPlot | $229.00 |
| MICROPRO | |
| Wordstar | $239.00 |
| MISCELLANEOUS | |
| Supercalc by Sorcim | $229.00 |
| Superwriter by Sorcim | $289.00 |
| Home Accountant Plus | $139.00 |
| ENTERTAINMENT | |
| Deadline | $39.00 |
| Temple of Apshai | $29.00 |
| Curse of Ra | $15.99 |
| Call For More IBM Software And Accessories | |

## DISK DRIVES

| | |
|---|---|
| CCI 100 for the TRS-80 Model 1 | |
| 5¼ 40 track | $299.00 |
| CCI 189 for the Zenith Z-89 | |
| 5¼ 40 track | $379.00 |
| CORVUS 5M with Mirror | $3089.00 |
| CORVUS 10M with Mirror | $4489.00 |
| CORVUS 20M with Mirror | $5389.00 |
| CORVUS Interfaces | CALL |
| RANA SYSTEMS add-on Disc Drive for the Apple II | |
| Elite One 40 Track | CALL |
| Elite Controller | CALL |
| Elite Two 80 Track | CALL |
| Elite Three 80 Track double-sided | CALL |

## DISKETTES

| | |
|---|---|
| Maxell 5¼ single side | $39.00 |
| Maxell 8 single side | $49.00 |
| Maxell 5¼ double side | $45.00 |
| Maxell 8 double side | $55.00 |
| BASF 5¼ | $26.95 |
| BASF 8 | $36.00 |
| Verbatim 5¼ | $26.95 |
| Verbatim 8 | $36.00 |
| Wabash 5¼ | $21.95 |

## IBM PC ACCESSORIES

| | |
|---|---|
| 64K Card by Microsoft | $435.00 |
| Joystick by T & G | $49.00 |
| 128K Card | $579.00 |
| 192K Card | $629.00 |
| 256K Card | $699.00 |
| Combo Card by Apparat | $249.00 |
| Call for more IBM PC add-ons | |

For fast delivery, send certified checks, money orders, or call to arrange direct bank wire transfers. Personal or company checks require one to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

## APPLE SOFTWARE

| | |
|---|---|
| MICROPRO | |
| Wordstar | $199.00 |
| Mailmerge | $89.00 |
| Spellstar | $149.00 |
| DataStar | $199.00 |
| CalcStar | $189.00 |
| VISICORP | |
| VisiCalc | $199.00 |
| VisiTerm | $79.00 |
| VisiDex | $199.00 |
| VisiPlot | $169.00 |
| VisiFile | $199.00 |
| VisiSchedule | $259.00 |
| VisiTrend/Plot | $239.00 |
| MISCELLANEOUS | |
| Micro Courier | $219.00 |
| Screen Director | $129.00 |
| Executive Briefing System | $169.00 |
| Supercalc | $199.00 |
| Personal Filing System | $115.00 |
| PFS Report Writer | $75.00 |
| Word Handler | $169.00 |
| ENTERTAINMENT | |
| Beer Run | $28.00 |
| Zork I, II | $26.95 |
| Deadline | $39.00 |
| Chop Lifter | $24.95 |
| Cannonball Blitz | $29.95 |
| Knights of Diamonds | $29.95 |
| Midnight Magic | $32.00 |
| Wizardry | $45.00 |
| Tuesday Morning Quarterback | $24.95 |
| Crush, Crumble & Chomp | $24.95 |
| Datestones of Ryn | $15.99 |
| Morloc's Tower | $15.99 |
| Snack Attack | $23.95 |
| Star Blazer | $24.95 |

## APPLE ACCESORIES

| | |
|---|---|
| ADVANCED LOGIC | |
| Add-Ram 16K Card | $99.00 |
| Z-Card CP/M for the Apple II | $225.00 |
| Smarterm 80 Column Board w/Softswitch | $249.00 |
| Z-80 Card by Microsoft | $319.00 |
| 16K Card by Microsoft | $159.00 |
| 32K Card by Saturn | $199.00 |
| Keyboard Enhancer II by Videx | $125.00 |
| Videoterm by Videx | $259.00 |
| Game Paddles by TG | $49.00 |
| Joystick by TG | $49.00 |
| Numeric Keypad by Keyboard Co. | $139.00 |
| ALF 9 Voice Board | $159.00 |
| ALF 3 Voice Board | $229.00 |
| System Saver by Kensington | $75.00 |
| Versacard by Prometheus | $229.00 |
| Microbuffer II 16K w/graphics | $259.00 |
| Microbuffer II 32K w/graphics | $299.00 |
| APPLE INTERFACE CARDS BY CCS | |
| Serial Asynch. #7710 | $139.00 |
| Centronics #7729 | $149.00 |
| Call for other CCS cards | |

## RAM

| | |
|---|---|
| 16K Ram Kit for Apple II; TRS80 | |
| 200 nano seconds, 4116 chips | $17.50 |

Dealer (National/International) Inquiries Invited.
Send for FREE catalogue

## COMPUTERS

| | |
|---|---|
| CALIFORNIA COMPUTER SYSTEM | |
| Mainframe 2200a | $485.00 |
| Z-80 CPU 2810a | $265.00 |
| 64K RAM 2065 | $569.00 |
| Floppy Controller 2422a | $359.00 |
| ZENITH | |
| Z-89 48K | CALL |
| Z-90 64K | CALL |
| Z-100 | CALL |
| Call For Prices On The Complete Zenith Line | |
| CASIO FX702P Pocket Computer | $179.00 |
| Sanyo MBC 1000 64K | CALL |
| Call For Prices On Complete Sanyo Computer Line | |

## PRINTERS

| | |
|---|---|
| NEC 7710 Serial | $2395.00 |
| NEC 7720 KSR | $2749.00 |
| NEC 7730 Parallel | $2395.00 |
| NEC 3510 Serial | $1850.00 |
| NEC 3520 | $2099.00 |
| NEC 3530 Parallel | $1850.00 |
| NEC 8023 Dot Matrix 100cps | $539.00 |
| Epson MX-80 | CALL |
| Epson MX-80FT | CALL |
| Epson MX-100 | CALL |
| IDS Micro Prism | CALL |
| IDS Prism 80 | CALL |
| IDS Prism 132 | CALL |
| Okidata Microline 80 | CALL |
| Okidata Microline 82A | CALL |
| Okidata Microline 83A | CALL |
| Okidata Microline 84 | CALL |
| Datasouth 180 cps | CALL |

## MONITORS

| | |
|---|---|
| Amdek 13" Color | $329.00 |
| Sanyo 9" B & W | $135.00 |
| Sanyo 9" Green | $140.00 |
| Sanyo 12" B & W | $179.00 |
| Sanyo 12" Green | $189.00 |
| Sanyo 13" Color | $359.00 |
| Zenith 12" Green | $105.00 |
| Zenith 13" Color | $339.00 |
| Electrohome 13" HI-RES Color Monitor | $829.00 |
| Electrohome 13" Color | $349.00 |
| Electrohome 12" B&W | $179.00 |
| Electrohome 12" Green | $189.00 |
| Electrohome 9" B&W | $149.00 |
| Electrohome 9" Green | $159.00 |

**SPECIAL OF THE MONTH**
**IDS MICRO PRISM**
**Dot Matrix – correspondence quality 110 CPS**
**Dot Plot Graphics CALL**

## TERMINALS

| | |
|---|---|
| ADDS Viewpoint | $495.00 |
| Zenith Z-19 | $679.00 |
| Televideo 910 | $595.00 |
| Televideo 925 | $779.00 |
| Televideo 950 | $969.00 |

## TELECOMMUNICATIONS

| | |
|---|---|
| Prentice Star Modem | $129.00 |
| Novation Cat Modem | $139.00 |
| Novation D-CAT | $149.00 |
| Novation AUTO-CAT | $199.00 |
| Novation APPLE CAT | $299.00 |
| Hayes Smart Modem | $249.00 |
| Smart Modem 1200 | CALL |
| Hayes Micro-Modem | $295.00 |
| Hayes Chronograph | $225.00 |
| Signalman Mark I | $85.00 |

# The CPU Shop

✓ 100

master charge

VISA

TO ORDER CALL TOLL FREE 1-800-343-6522
420-438 Rutherford Ave., Dept. MI 12, Charlestown, Massachusetts 02129

Hours 9 AM – 9 PM (EST) Mon.-Fri. (Sat. till 6)
Technical information call 617/242-3361

TWX- 710-348-1796

Massachusetts Residents call 617/242-3361
Massachusetts Residents add 5% Sales Tax

✓ See List of Advertisers on Page 451

Line 4 calls the CLS function again. Line 5 computes the value of C, and line 6 prints the answer. Line 7 branches to Begin, our label for line 2. (The right arrow is used for the Branch function.)

### Forth

Forth is an abbreviation for Fourth-generation language. However, the second-generation system on which it was originally written only allowed five-character program names. Hence, Fourth became Forth!

Forth is different. On the positive side, you are almost forced to write structured code in Forth. Forth is extensible. If you need a feature (say the arctangent of the ratio of two numbers) you can add it. The feature is then as much a part of the language as any other word.

On the negative side, writing a program in Forth is different than writing a program in Basic, Fortran, Cobol or Pascal, and is more like APL. You must learn the jargon.

As it comes out of the box, Forth does not support floating-point numbers, although you can add that capability.

Forth is a stack-oriented language. This means arithmetic statements are in Reverse Polish Notation (RPN)—the system used by Hewlett-Packard calculators and internally by almost all computers. Numbers are entered onto a stack (the HP calculators have an enter button; Forth just requires a space) and then operations are performed on the numbers on the stack. For instance, 1 + 2 * 3 would be written 1 2 3 * + in Forth.

In Program Listing 6 line 0 is a comment. Comments in Forth are enclosed in parentheses. Note the word programs—there are three programs in this block: FC, FTOC, and FCPROG. (A block in Forth is very much like a workspace in APL.)

Line 1 is the definition of FC. Colons start a definition, and semicolons end it. FC starts with a Fahrenheit value on the operand stack, pushes 32 onto the stack, subtracts the top of stack (TOS) from the next value down, pushes 50 onto the stack, and so on. As the comment says, 10 times the Celsius value is left on the stack.

As I mentioned before, Forth uses integer arithmetic. Unless you use a floating-point or multiple-precision package, numbers are limited to the range 32767 to -32768. As the program is written, the Fahrenheit value must be less than 656 degrees. With a larger value, the multiplication by 50 will overflow.

However, a built-in operator (*/) performs a multiply followed by a divide. The results of the multiplication are double precision. Using this operator would allow temperatures up to 5,930 degrees Fahrenheit to be converted.

Lines 2, 3 and 4 define a program, FTOC, that prints "Degrees Fahrenheit Is," takes the number on the stack and converts it using FC, and prints the result and then "Degrees Celsius." The code in line 3 between the words DUP and TYPE outputs the number as a signed number with one decimal place.

Lines 5 and 6 define the third program, FCPROG. This is the outer program—it runs an endless loop with the BEGIN and O END, asks "Type in Fahrenheit Value," inputs the value, and calls FTOC, which does the conversion and outputs the result. ■

*Robert Nicholas (2-B Lennox Heights, Lenox, MA 01240) runs a data-processing center. Philip Martel, an electrical engineer, can be reached at 748 Tyler St., Pittsfield, MA 01201.*

*A software solution.*

# Grandpa "Spooled" His Computer

*Tom Shield*
*40 Lake Park Road*
*Champaign, IL 61820*

Most large computers operate on a time sharing basis; the computer does a little of each of the many tasks it has to do at a time so that it can process all the tasks at the same time. Many peripherals are slower than the CPU itself; items such as terminals, printers and card readers run much slower than the CPU can drive them. The CPU can service each peripheral and complete many other tasks before servicing the peripheral again. There is no need for time sharing to allow many users on a TRS-80, since most of us have only one terminal, but most of us do have a printer and need a spooler.

Many spooling methods store the data quickly and then print the data while the CPU is doing other tasks. One method involves an extra CPU and memory. A second method, which I will present, involves using the computer's memory together with an interrupt routine to send characters to the printer. Both methods have advantages and disadvantages. The hardware method is invisible to the user and leaves all the machine's memory free, but at a prohibitive price. The second choice, a software spooler, is not invisible to the user and, depending on the buffer size, can drastically decrease available memory. On the other hand, it is less expensive. Thus the software solution is the choice of most small system users.

### Operation

Spooling requires two separate actions—storing the data and then sending it to the printer intermittently so the computer can do other tasks. The information must be intercepted before it is sent to the printer. The Model III PRCHAR (003BH) routine jumps out into RAM at 4025H and finds a JP 03C2H instruction returning it to ROM. It is simple to change this to the address of my storage routine. At the point the routine jumps to RAM, the character to be printed is in the C register and the registers are saved on the stack in the order HL,DE,IX,HL,BC (the first is an extra to balance the stack). I save the contents of the C register, POP the saved registers off the stack, and return to the calling program.

The Z80's first maskable interrupt causes a jump to 38H in the ROM and disassembly shows a subsequent jump to RAM at

*Program Listing 1*

```
ROUTINE TO DUMP BUFFER TO PRINTER BY A USR CALL

FD70      F3           DI                    'DISABLE INTERRUPTS
FD71      21 C2 03     LD      HL,03C2H      'REPLACE JUMP TO STORE-
FD74      22 26 40     LD      (4026H),HL    ' ROUTINE TO DISABLE SPOOLER
FD77      ED 4B FC FD  LD      BC,(FDFCH)    'GET END POINTER
FD7B      2A FE FD     LD      HL,(FDFEH)    'GET BEGINNING POINTER
FD7E      7E           LD      A,(HL)        'GET CHARACTER FROM BUFFER
FD7F      CD 3B 00     CALL    003BH         'PRINT IT
FD82      23           INC     HL            'POINT TO NEXT ONE
FD83      CD 8D 02     CALL    028DH         'CHECK FOR BREAK
FD86      C2 60 FE     JP      NZ,FE60H      'IF BREAK EXIT
FD89      E5           PUSH    HL            'SAVE CURRENT LOCATION
FD8A      ED 42        SBC     HL,BC         'CHECK FOR END
FD8C      E1           POP     HL            'RESTORE CURRENT LOCATION
FD8D      C2 7E FD     JP      NZ,FD7EH      'LOOP IF NOT DONE
FD90      CD 50 FE     CALL    FE50H         'RESET POINTERS
FD93      21 A0 FD     LD      HL,FDA0H      'STORE ROUTINE ADDRESS
FD96      22 26 40     LD      (4026H),HL    'CHANGE PRCHAR JUMP
FD99      FB           EI                    'ENABLE INTERRUPTS
FD9A      C9           RET                   'RETURN TO CALLER
FD9B-FD9F      00           NOP
```

*Listing continues*

4012H. There a JP 3018H instruction takes the program back into ROM. I inserted a routine with a simple change in this jump address. Since the interrupt routine must continue after the inserted routine is done, I saved and restored all registers. A JP 3018H instruction ends the routine.

The programs in Listing 1 accomplish what I have just described for a 48K Model III with or without disks. There are three main programs, two subroutines used by the main programs and two short programs to enable and disable the spooler. The program to store all characters sent to the printer resides at FDA0H; the program inserted in the interrupt sequence is at FE00H. The program at FD70H accessed by a USR call from Basic dumps the buffer to the printer. The enable spooler routine at location FE40H puts in the addresses of my routines instead of the ROM addresses; the routine at FE70H reverses this process, disabling the spooler. With the routine at FE40H set as the starting point for execution after loading from disk, the spooler is on immediately after loading. The disable spooler routine is called from Basic via a USR call.

These programs use a predefined buffer area and communicate the current buffer end and beginning by using four bytes of memory as pointers. Addresses FDFCH and FDFDH hold the end of buffer address, FDFEH and FDFFH hold the beginning address. Initializing the pointers to the starting value of F000H sets the limits on the buffer; you can replace F000H at address FE51H with any starting address of the buffer. The next time the buffer is empty the pointers will be reset to this new value. The end of the buffer is set at FD60H so the program is not written over.

The programs also check memory bytes used as switches to control the spooler's operation. The byte at FDFAH controls whether the characters are stored or sent directly to the printer; a zero value here selects storage and anything else reverts to normal operation. Address FDFBH controls the interrupt routine. If it is zero, the cur-

*Listing 1 continued*

```
ROUTINE TO STORE ALL CHARACTERS SENT TO PRINTER

FDA0      F3             DI                      'DISABLE INTERRUPTS
FDA1      F5             PUSH    AF              'SAVE AF
FDA2      3A FA FD       LD      A,(FDFAH)       'GET SWITCH VALUE
FDA5      FE 00          CP      00H             'IS IT ZERO?
FDA7      28 05          JR      Z,FDAEH         'CONTINUE IF IT IS
FDA9      F1             POP     AF              'RESTORE AF
FDAA      FB             EI                      'ENABLE INTERRUPTS
FDAB      C3 C2 03       JP      03C2H           'RETURN TO PRCHAR IN ROM
FDAE      79             LD      A,C             'TRANSFER CHARACTER TO A
FDAF      ED 5B FC FD LD         DE,(FDFCH)      'GET END POINTER
FDB3      12             LD      (DE),A          'SAVE CHARACTER
FDB4      13             INC     DE              'NEXT LOCATION
FDB5      21 60 FD       LD      HL,FD60H        'END OF BUFFER
FDB8      ED 52          SBC     HL,DE           'CHECK FOR OVERFLOW
FDBA      CA CA FD       JP      Z,FDCAH         'IF IT IS DUMP BUFFER
FDBD      ED 53 FC FD LD         (FDFCH),DE      'REPLACE POINTER
FDC1      F1             POP     AF              'BALANCE STACK
FDC2      E1             POP     HL              'BALANCE STACK
FDC3      D1             POP     DE              'RESTORE REGISTERS
FDC4      DD E1          POP     IX
FDC6      E1             POP     HL
FDC7      C1             POP     BC
FDC8      FB             EI                      'ENABLE INTERRUPTS
FDC9      C9             RET                     'RETURN TO CALLER
FDCA      21 C2 03       LD      HL,03C2H        'ORIGINAL ADDRESS
FDCD      22 26 40       LD      (4026H),HL      'REPLACE IT
FDD0      2A FE FD       LD      HL,(FDFEH)      'GET BEGINNING POINTER
FDD3      7E             LD      A,(HL)          'GET CHARACTER
FDD4      D5             PUSH    DE              'SAVE END POINTER
FDD5      CD 3B 00       CALL    003BH           'PRINT CHARACTER
FDD8      D1             POP     DE              'RESTORE POINTER
FDD9      23             INC     HL              'NEXT CHARACTER
FDDA      CD 8D 02       CALL    028DH           'CHECK FOR BREAK
FDDD      C2 60 FE       JP      NZ,FE60H        'ON BREAK EXIT
FDE0      E5             PUSH    HL              'SAVE CURRENT LOCATION
FDE1      ED 52          SBC     HL,DE           'CHECK FOR END
FDE3      E1             POP     HL              'RESTORE CURRENT LOCATION
FDE4      C2 D3 FD       JP      NZ,FDD3H        'LOOP IF NOT DONE
FDE7      CD 50 FE       CALL    FE50H           'RESET POINTERS
FDEA      21 A0 FD       LD      HL,FDA0H        'STORE ROUTINE ADDRESS
FDED      22 26 40       LD      (4026H),HL      'CHANGE PRCHAR JUMP
FDF0      C3 C1 FD       JP      FDC1H           'GOTO EXIT
FDF3-FDF9     00               NOP

SWITCHES AND POINTERS

FDFA      00                                     'SWITCH FOR STORAGE
FDFB      00                                     'SWITCH FOR INTERRUPT
FDFC      00                                     'END OF BUFFER POINTER
FDFD      F0
FDFE      00                                     'START OF BUFFER POINTER
FDFF      F0

ROUTINE INSERTED IN INTERRUPT SEQUENCE

FE00      F5             PUSH    AF              'SAVE AF
FE01      3A FB FD       LD      A,(FDFBH)       'GET SWITCH VALUE
FE04      FE 00          CP      00H             'IS IT ZERO?
FE06      28 04          JR      Z,FE0CH         'IF IT IS CONTINUE
FE08      F1             POP     AF              'RESTORE AF
FE09      C3 18 30       JP      3018H           'RETURN TO INTERRUPT
FE0C      DB F8          IN      A,(F8H)         'GET PRINTER STATUS
FE0E      E6 F0          AND     F0H             'CHECK IT
FE10      FE 30          CP      30H             'IS IT 48?
FE12      C2 08 FE       JP      NZ,FE08H        'EXIT IF NOT READY
FE15      E5             PUSH    HL              'SAVE HL,DE
FE16      D5             PUSH    DE
FE17      2A FE FD       LD      HL,(FDFEH)      'GET BEGINNING POINTER
FE1A      ED 5B FC FD LD         DE,(FDFCH)      'GET END POINTER
FE1E      E5             PUSH    HL              'SAVE CURRENT LOCATION
FE1F      ED 52          SBC     HL,DE           'CHECK IF BUFFER IS EMPTY
FE21      E1             POP     HL              'RESTORE CURRENT LOCATION
FE22      CA 2E FE       JP      Z,FE2EH         'IF EMPTY GOTO EXIT
FE25      7E             LD      A,(HL)          'GET CURRENT CHARACTER
FE26      D3 F8          OUT     (F8H),A         'SENT IT TO PRINTER
FE28      23             INC     HL              'POINT TO NEXT CHARACTER
FE29      22 FE FD       LD      (FDFEH),HL      'REPLACE POINTER
FE2C      18 03          JR      FE31H           'DON'T RESET POINTERS
FE2E      CD 50 FE       CALL    FE50H           'RESET POINTERS
FE31      D1             POP     DE              'RESTORE REGISTERS
FE32      E1             POP     HL
FE33      F1             POP     AF
FE34      C3 18 30       JP      3018H           'CONTINUE WITH INTERRUPT SEQUENCE
FE37-FE3F     00               NOP

ROUTINE TO ENABLE THE SPOOLER

FE40      21 A0 FD       LD      HL,FDA0H        'LOCATION OF STORE ROUTINE
FE43      22 26 40       LD      (4026H),HL      'CHANGE PRCHAR JUMP
FE46      21 00 FE       LD      HL,FE00H        'LOCATION OF INTERRUPT ROUTINE
FE49      22 13 40       LD      (4013H),HL      'CHANGE INTERRUPT JUMP
FE4C      C3 2D 40       JP      402DH           'RETURN TO DOS
FE4F      00             NOP

SUBROUTINE TO RESET POINTERS TO START OF BUFFER

FE50      21 00 F0       LD      HL,F000H        'START OF BUFFER
```

*Listing 1 continues*

✓See List of Advertisers on Page 451

rent byte in the buffer is sent to the printer every interrupt cycle, otherwise nothing is done. This allows all the data printed by a program to be stored and dumped at once, using the routine at FD70H after program execution is complete. If the buffer is filled, the contents will be sent to the printer regardless of the status of the byte at FDFBH. This ensures no lost data on buffer overflow.

*Listing 1 continued*

```
FE53      22 FE FD     LD        (FDFEH),HL       'RESET START POINTER
FE56      22 FC FD     LD        (FDFCH),HL       'RESET END POINTER
FE59      C9           RET                        'RETURN
FE5A-FE5F      00             NOP

EXIT ROUTINE ON BREAK

FE60      CD 50 FE     CALL      FE50H            'RESET POINTERS
FE63      21 A0 FD     LD        HL,FDA0H         'LOCATION OF STORE ROUTINE
FE66      22 26 40     LD        (4026H),HL       'REPLACE JUMP ADDRESS
FE69      FB           EI                         'ENABLE INTERRUPTS
FE6A      C3 19 1A     JP        1A19H            'JUMP TO BASIC READY
FE6D-FE6F      00             NOP

ROUTINE TO DISABLE THE SPOOLER BY A USR CALL

FE70      21 C2 03     LD        HL,03C2H         'ORIGINAL PRCHAR ADDRESS
FE73      22 26 40     LD        (4026H),HL       'REPLACE IT
FE76      21 18 30     LD        HL,3018H         'ORIGINAL INTERRUPT ADDRESS
FE79      22 13 40     LD        (4013H),HL       'REPLACE IT
FE7C      C9           RET                        'RETURN TO CALLER
```

Listing 1: Commented disassembly of spooler as it should be loaded into memory.

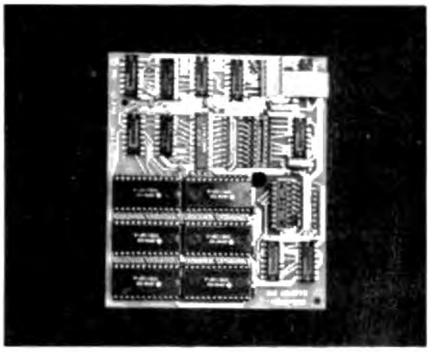
### Use

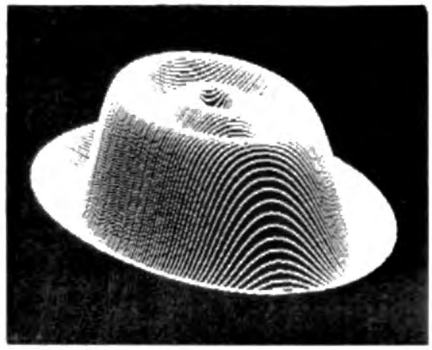
The spooler program given here is invisible; when in place the only sign of its operation is the Ready prompt under Basic or DOS while the printer is running. The program can be entered with a disk system by using Debug to POKE the values into memory and then using DUMP to save it to disk. When dumping the program to disk, set TRA = 0FE40 so the program is enabled immediately when it is called under DOS. Make sure the pointers are included in the dump and are both set to F000H; or the program will try to send the empty buffer to the printer. Set the switch values to the status desired when the program is loaded. Without disks, use Listing 2 to POKE the program into memory. At FE4CH the jump to DOS (402DH) is replaced with a return (C9) instruction allowing the spooler to be enabled by the Basic USR call in the program.

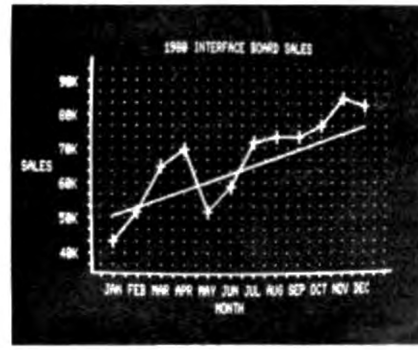
*Program Listing 2*

```
10 CLS:DEFINT I
20 SUM=0
30 PRINT"PROGRAM TO LOAD SPOOLER FOR THE"
40 PRINT"MODEL III, BY TOM SHIELD, AUGUST 1981"
50 PRINT"MEMORY SIZE SHOULD BE SET AT 61400"
60 FORI=-656 TO -388   'LOAD FROM FD70H TO FE7CH
70 READ BYTE
80 POKE I,BYTE
90 SUM=SUM+BYTE
100 NEXT I
110 IF SUM<>34143 THEN PRINT"CHECK SUM ERROR, CHECK 'DATA' STATE
```

*Listing 2 continues*



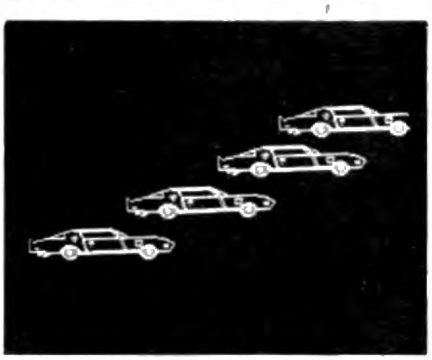





## Mod III — GRAFYX SOLUTION™ — $299.95 ✓464

- 512 × 192 Dot Graphics increases resolution 16× (better than Apple)
- Extensive Business, Personal, Educational, and Scientific applications
- 14 BASIC commands set and reset Points, Lines, Circles, Boxes, etc.
- Allows Display and Printout of detailed Line, Bar, and Pie graphs
- 80 Character/line display compatible with BASIC programs and DOS
- Price includes 98K bits memory, 30 programs, 52 page manual
- Plug-in, clip-on board eliminates soldering for easy installation

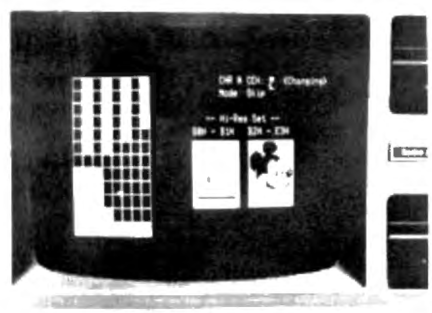
---



REBATE $40 REBATE

VALID 11-1 TO 12-31-82

After purchasing Grafyx Soln. or 80-Grafix, return this coupon with registration form to receive rebate.



## Mod I, III — 80-GRAFIX™ — $169.95 ✓465

- Create and Save 128 programmable hi-resolution characters/shapes
- Characters are generated within an 8 × 12 matrix (Mod I: 6 × 12)
- Great for creating game figures, symbols, and alternate character sets
- Excellent software support (20 programs) & extensive documentation
- Module easily installs within the computer with no soldering

Manual only $15
Products guaranteed
Dealers welcome

**MICRO-LABS, INC.** 214-235-0915
902 Pinecrest, Richardson, Texas 75080

FREE shipping
Prepaid or COD
Tx. res add 5%

✓ See List of Advertisers on Page 451

Under Basic answer the memory size question with 61400 to protect the program and its buffer. Different buffer sizes require a different setting. Under DOS, the program needs to be protected from being overwritten by commands that use the top 180H of memory. Executing a Clear (MEM = 0F000) under DOS allows the spooler to execute under DOS as well as Basic. Failure to do this causes the machine to hang up when the program is erased and the interrupt sequence branches to a nonexistent routine.

The spooler stores data and sends it to the printer when it is available; since the buffer size is fixed, only a limited amount of data can be stored. If the buffer is filled faster than it is printed, it will eventually be full. When this happens the whole buffer is dumped to the printer so that no data are lost. Since this is done by the character storing routine, the Ready prompt will not appear until the buffer is empty and normal operation has resumed. If you do not want the buffer contents or simply need to stop sending data to the printer, Break stops execution and clears the buffer. This is also true for the routine to dump the buffer via a USR call. Since the interrupt routine only sends data to the printer when it is ready you can put your printer off-line to change or adjust the paper while a program is running without stopping its execution.

A spooler is a very useful utility but you must understand what it does to maximize its use. If a program producing output is slower than the printer you do not need the spooler: The printer will be ready when the program needs it. If you use programs that need large amounts of memory or do word-processing with big documents, using memory as a buffer is detrimental to normal operation. On the other hand, if you have a short program to LLIST, after which you would like to look at a disk directory or run the program, it is great to get the Ready prompt almost before the printer starts. Or if you want to print 10 each of various mailing labels you can enter the next address while the previous one is still printing! The advantages are endless. The sound of most printers is not very musical; with a spooler I

*Listing 2 continued*

```
MENTS":END  ELSE PRINT"LOAD COMPLETE, AND SPOOLER RUNNING"
120 POKE 16526,64:POKE 16527,254 'WITH DISKS DEFUSR0=&HFE40
130 X=USR(0)  'OR X=USR0(0) WITH DISKS
140 PRINT"SOME IMPORTANT ADDRESSES ARE:"
150 PRINT" -448 (&HFE40)=LOCATION OF ENABLE SPOOLER ROUTINE"
160 PRINT" -400 (&HFE70)=LOCATION OF DISABLE SPOOLER ROUTINE"
170 PRINT" -656 (&HFD70)=LOCATION OF PRINT BUFFER ROUTINE"
180 PRINT" -518 (&HFDFA)=LOCATION OF STORE SWITCH"
190 PRINT" -517 (&HFDFB)=LOCATION OF INTERRUPT SWITCH"
200 END
210 DATA 243, 33, 194, 3, 34, 38, 64, 237
220 DATA 75, 252, 253, 42, 254, 253, 126, 205
230 DATA 59, 0, 35, 205, 141, 2, 194, 96
240 DATA 254, 229, 237, 66, 225, 194, 126, 253
250 DATA 205, 80, 254, 33, 160, 253, 34, 38
260 DATA 64, 251, 201, 0, 0, 0, 0, 0
270 DATA 243, 245, 58, 250, 253, 254, 0, 40
280 DATA 5, 241, 251, 195, 194, 3, 121, 237
290 DATA 91, 252, 253, 18, 19, 33, 96, 253
300 DATA 237, 82, 202, 202, 253, 237, 83, 252
310 DATA 253, 241, 225, 209, 221, 225, 225, 193
320 DATA 251, 201, 33, 194, 3, 34, 38, 64
330 DATA 42, 254, 253, 126, 213, 205, 59, 0
340 DATA 209, 35, 205, 141, 2, 194, 96, 254
350 DATA 229, 237, 82, 225, 194, 211, 253, 205
360 DATA 80, 254, 33, 160, 253, 34, 38, 64
370 DATA 195, 193, 253, 0, 0, 0, 0, 0
380 DATA 0, 0, 0, 0, 0, 240, 0, 240
390 DATA 245, 58, 251, 253, 254, 0, 40, 4
400 DATA 241, 195, 24, 48, 219, 248, 230, 240
410 DATA 254, 48, 194, 8, 254, 229, 213, 42
420 DATA 254, 253, 237, 91, 252, 253, 229, 237
430 DATA 82, 225, 202, 46, 254, 126, 211, 248
440 DATA 35, 34, 254, 253, 24, 3, 205, 80
450 DATA 254, 209, 225, 241, 195, 24, 48, 0
460 DATA 0, 0, 0, 0, 0, 0, 0, 0
470 DATA 33, 160, 253, 34, 38, 64, 33, 0
480 'IN LINE 490 FOR DISKS REPLACE 201->195,0->45,0->64
490 DATA 254, 34, 19, 64, 201, 0, 0, 0
500 DATA 33, 0, 240, 34, 254, 253, 34, 252
510 DATA 253, 201, 0, 0, 0, 0, 0, 0
520 DATA 205, 80, 254, 33, 160, 253, 34, 38
530 DATA 64, 251, 195, 25, 26, 0, 0, 0
540 DATA 33, 194, 3, 34, 38, 64, 33, 24
550 DATA 48, 34, 19, 64, 201
```

*Program Listing 3*

```
FE80    7E          LD      A,(HL)      'GET 1ST CHARACTER AFTER FILESPEC
FE81    FE 0D       CP      0DH         'IS THERE NONE?
FE83    CA DC FE    JP      Z,FEDCH     'THEN JUST ENABLE SPOOLER
FE86    CD 00 FF    CALL    FF00H       'CONVERT TO HEX
FE89    E5          PUSH    HL          'SAVE POSITION
FE8A    21 00 10    LD      HL,1000H    'MULTIPLIER FOR 1ST DIGIT
FE8D    D5          PUSH    DE          'SAVE DE
FE8E    CD 4E 44    CALL    444EH       'CALL $DMULT
FE91    D1          POP     DE          'RESTORE DE
FE92    65          LD      H,L         'MOVE MIDDLE BYTE
```

*Listing 3 continues*

*Listing 3 continued*

```
FE93    6F          LD     L,A           'MOVE LOWER BYTE
FE94    EB          EX     DE,HL         'SAVE RESULT
FE95    E1          POP    HL            'RESTORE POSITION OF BUFFER
FE96    23          INC    HL            'NEXT CHARACTER
FE97    CD 00 FF    CALL   FF00H         'CONVERT TO HEX
FE9A    E5          PUSH   HL            'SAVE POSITION
FE9B    21 00 01    LD     HL,0100H      'MULTIPLIER FOR 2ND DIGIT
FE9E    D5          PUSH   DE            'SAVE DE
FE9F    CD 4E 44    CALL   444EH         'CALL $DMULT
FEA2    D1          POP    DE            'RESTORE DE
FEA3    65          LD     H,L           'MOVE MIDDLE BYTE
FEA4    6F          LD     L,A           'MOVE LOWER BYTE
FEA5    ED 5A       ADC    HL,DE         'ADD IN EARLIER RESULT
FEA7    EB          EX     DE,HL         'SAVE IT
FEA8    E1          POP    HL            'RESTORE BUFFER POSITION
FEA9    23          INC    HL            'NEXT CHARACTER
FEAA    CD 00 FF    CALL   FF00H         'CONVERT TO HEX
FEAD    E5          PUSH   HL            'SAVE POSITION
FEAE    21 10 00    LD     HL,0010H      'MULT FOR 3RD DIGIT
FEB1    D5          PUSH   DE            'SAVE RESULT
FEB2    CD 4E 44    CALL   444EH         'CALL $DMULT
FEB5    D1          POP    DE            'RESTORE TOTAL
FEB6    65          LD     H,L           'MOVE MIDDLE BYTE
FEB7    6F          LD     L,A           'MOVE LOWER BYTE
FEB8    ED 5A       ADC    HL,DE         'GET TOTAL
FEBA    EB          EX     DE,HL         'SAVE IT
FEBB    E1          POP    HL            'RESTORE BUFFER POSITION
FEBC    23          INC    HL            'NEXT ONE
FEBD    CD 00 FF    CALL   FF00H         'CONVERT TO HEX
FEC0    6F          LD     L,A           'LOW ORDER BYTE
FEC1    26 00       LD     H,00H         'HIGH ORDER BYTE
FEC3    ED 5A       ADC    HL,DE         'ADD TO TOTAL
FEC5    11 00 80    LD     DE,8000H      'LOWER LIMIT ON BUFFER
FEC8    E5          PUSH   HL            'SAVE TOTAL
FEC9    ED 52       SBC    HL,DE         'CHECK IT
FECB    E1          POP    HL            'GET TOTAL
FECC    FA DC FE    JP     M,FEDCH       'EXIT IF TOO LOW
FECF    E5          PUSH   HL            'SAVE TOTAL
FED0    11 60 FD    LD     DE,FD60H      'UPPER LIMIT ON BUFFER
FED3    ED 52       SBC    HL,DE         'CHECK IT
FED5    E1          POP    HL            'GET TOTAL
FED6    F2 DC FE    JP     P,FEDCH       'EXIT IF TOO HIGH
FED9    22 51 FE    LD     (FE51H),HL    'CHANGE BUFFER POINTER
FEDC    21 A0 FD    LD     HL,FDA0H      'STORE ROUTINE ADDRESS
FEDF    22 26 40    LD     (4026H),HL    'CHANGE LPRINT JUMP
FEE2    21 00 FE    LD     HL,FE00H      'INTERRUPT ROUTINE ADDRESS
FEE5    22 13 40    LD     (4013H),HL    'CHANGE INTERRUPT JUMP
FEE8    C3 2D 40    JP     402DH         'JUMP TO DOS

FEEB-FEFF    00          NOP             'OR ANYTHING

FF00    7E          LD     A,(HL)        'GET CHARACTER
FF01    FE 40       CP     40H           'COMPARE TO  40H
FF03    F2 0A FF    JP     P,FF0AH       'IF LETTER JUMP
FF06    DE F2       SBC    A,2FH         'CORRECTION FOR NUMBERS
FF08    18 02       JR     FF0CH         'GOTO RETURN
FF0A    DE 37       SBC    A,37H         'CORRECTION FOR LETTERS
FF0C    C9          RET                  'RETURN
```

Listing 3: Routine to set the buffer size when the spooler is called from DOS. It is simply added to the rest of the program with TRA=FE80 on dump to disk.

spend less time listening and staring at a lifeless screen.

**Program Refinements**

I have made two additions to the program for DOS users only. You can solve the problem of executing a CLEAR (MEM = xxxxx) by applying the following patch to the spooler once it is on disk:

```
PATCH SPOOL/CMD (ADD = 0FE59,FIND
  = C9000000,CHG = 221144C9)
```

This patch automatically protects the spooler from DOS. The program in Listing 3 sets the start of the buffer when the spooler is called from DOS. Add this segment to the end of the spooler program and DUMP from FD70H to FF0CH to disk setting TRA = 0FE80. The program reads the first four characters after the filespec (the normal TRSDOS syntax) and uses that address as the new beginning for the buffer. You must enter the address in hexadecimal format with no leading characters. To set the start of the buffer to D000H execute under DOS: SPOOL D000. If no address is entered the default value of F000H is used as before. Note you must still protect the spooler under Basic. Another useful trick is to DUMP the code at FE70H to FE7CH with TRA = 0FE70 under the name of SPOOLOFF. Entering SPOOLOFF from DOS turns the spooler off. ■



✓ See List of Advertisers on Page 451

*Now where did I put these files?*

# Diskmap

*LCDR John B. Harrell, III*
*RR 1, Box 841, Tatnic Rd.*
*Wells, ME 04090*

I have wondered many times just how the Disk Operating System places files on the disk. I have had to worry about the placement of files when moving from one disk system to the other especially if the track count for the drives is different. One may easily find file locations with a utility such as Apparat's Superzap and a reference such as the *TRS-80 Disk and Other Mysteries* by H.C. Pennington. This is time-consuming and requires constant reference to the book to remember what the information presented means.

Diskmap uses the information in Pennington's book to present file structure concisely. Figure 1 shows the type of display that Diskmap constructs, repeated four times to map a 40-track disk. I have found it useful to look at the directories of disks with no system on them. Diskmap was written to be independent of the major operating systems primarily by the inclusion of disk read and manipulation routines.

### Floppy Disk Formatter/Controller

At the heart of this utility program are the disk control and input routines. To understand these routines, you must understand how the disk is interfaced with the expansion interface and the Western Digital FD1771-01 floppy disk formatter/controller chip.

The Floppy Disk Controller (FDC) takes commands from the computer and uses them to position the proper disk head and to read data from or write data to the storage medium (the disk). The FDC interface to the computer is an 8-bit bidirectional bus for data, status, and command transfers. On the TRS-80 this interface consists of memory-mapped addresses for each FDC register. Data flow to or from the FDC is controlled by one of the addresses in Table 1 in a memory-store or memory-load operation.

There are eleven commands used with the FDC: These commands are divided into four groups. Refer to Table 2 for the structure of the commands. Each command is issued to the FDC by a store operation to the command register (X'37EC') using the appropriate command. Obtain the status of the command just issued from a load from the Status register (X'37EC'); interpret it by reference to Table 3. A Read Sector command is performed by first seeking the correct track and looking at status to determine the state of the Busy bit, while waiting for the command to terminate. The appropriate sector number is then stored in the Sector register and the Read command is issued. The program then polls the DRQ and BUSY bits to determine the state of the Read command.

The Restore command repositions the disk head over track zero and the FDC Track register is set to zero at the termination of the operation. The h field, the Head Load flag, causes no action on the TRS-80 as the head is loaded when the disk is selected and the drive motor is turned on. The v flag causes the FDC to verify that the track number on the track last read is the same as the contents of the Track register. The last field, rr, controls the rate at which the FDC issues stepping pulses to the disk. The stepping rates for the disk head given in Table 2 are for an FDC using a one Megahertz clock; a 2 MHz clock halves these times. These three fields are the same for the remainder of the Type I commands.

```
                           *DISK GRANULE MAP*
DRIVE:0   NEWDOS80   02/20/81   GRANS FREE:3   TRKS LOCKED OUT:0
                   .   1ST GRANULE    .   2ND GRANULE    .
                   ..............................................
TRACK NUMBER: 0    <--- BOOT/SYS    --->.<--- SYS0/SYS    --->.
TRACK NUMBER: 1    <--- SYS0/SYS    --->.<--- SYS0/SYS    --->.
TRACK NUMBER: 2    .<-- LMOFFSET/CMD -->.<-- LMOFFSET/CMD -->.
TRACK NUMBER: 3    .<-- LEVEL1/CMD   -->.<--- LEVEL1/CMD --->.
TRACK NUMBER: 4    .<-- LEVEL1/CMD   -->.<--- LEVEL1/CMD --->.
TRACK NUMBER: 5    .<--- BUILDER    --->.<--- BUILDER     --->.
TRACK NUMBER: 6    .<--- VERIFY/JCL --->.<-- SPOOLER/CMD  -->.
TRACK NUMBER: 7    .<-- SPOOLER/CMD  -->.< ----------------- >.
TRACK NUMBER: 8    .< ----------------- >.<-- SAMPLE01/BAS  -->.
TRACK NUMBER: 9    .<-- SAMPLE01/BAS -->.<-- SAMPLE01/BAS  -->.
                   ..............................................
             PRESS ANY KEY FOR NEXT 10 TRACK DISPLAY
```

*Fig. 1. Disk Granule Map*

### *The Key Box*

**Basic Level II**
**Model I**
**16K RAM**
**TRSDOS**
**One to four disk drives**

The Seek command tells the FDC to compare the track number in the Data register (a previous instruction must load the Data register) with the current contents of the Track register. The FDC then issues the appropriate number of stepping pulses in the correct direction to cause the head to travel to the desired track.

The Step command causes the FDC to issue one step pulse to the selected drive's head. The direction of the Step is the same as the direction of the last Step issued. The u field updates the Track register to the proper value for the corresponding track. If the u flag is not used, the user must keep track of the disk head location.

The two commands Step-In and Step-Out are similar to Step except that the direction of head travel is explicitly defined in the command. All fields in these commands operate as in the Step command.

The second group of FDC commands are data movement commands for reading and writing data to or from the disk. The adaption used to transfer data in the TRS-80 is called Programmed input/output. The Z80 Central Processing Unit (CPU) must continually poll the FDC and be ready to accept or provide data. This concept is opposed by the Direct Memory Access (DMA) method. With DMA the FDC actually demands and receives full control to the address and data busses of the computer; data transfers directly to the memory store without the CPU. This allows overlapped execution with data transfer and alleviates the I/O bound central processor.

There are several requirements before the performance of Read or Write commands. First, the drive must be selected and the motor must be up to normal operational speed. TRSDOS and NEWDOS 2.1 assume that this time delay is approximately one second. NEWDOS80 sets this delay at one-half second. Second, a Seek instruction must position the head over the appropriate track. Third, the Sector register in the FDC must be set to the number of the sector to be read or written.

The format of the Read and Write commands is in Table 2. Note that they have similar fields of m, b, and e. The m field enables a multiple sector transfer of information. The FDC transfers information and updates the sector register until the sector register contains a number larger than the highest sector number on the track or until a Force Interrupt command is loaded into the Command register. The b field identifies the format type as standard IBM or non-standard. This changes the FDC's interpretation of the sector length code and is part of the record identification. The e field is not used in the Model I. On the Write command, the aa field specifies the type of Data Address Mark (DAM) for the sector transferred. The standard DAM is an X'FB' for all sectors other than the directory (DIR/SYS), which uses DAM X'FA' to show

| TRS-80 ADDRESS | MEMORY LOAD | MEMORY STORE |
|---|---|---|
| X'37E1' | N/A | Select drive & start |
| X'37EC' | Read STATUS reg | Load COMMAND reg |
| X'37ED' | Read TRACK reg | Load TRACK reg |
| X'37EE' | Read SECTOR reg | Load SECTOR reg |
| X'37EF' | Read DATA reg | Load DATA reg |

*Table 1. Memory Map*

| TYPE | COMMAND | FORMAT | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| I | RESTORE | 0 | 0 | 0 | 0 | h | v | r | r |
| I | SEEK | 0 | 0 | 0 | 1 | h | v | r | r |
| I | STEP | 0 | 0 | 1 | u | h | v | r | r |
| I | STEP IN | 0 | 1 | 0 | u | h | v | r | r |
| I | STEP OUT | 0 | 1 | 1 | u | h | v | r | r |
| II | READ COMMAND | 1 | 0 | 0 | m | b | e | 0 | 0 |
| II | WRITE COMMAND | 1 | 0 | 1 | m | b | e | a | a |
| III | READ ADDRESS | 1 | 1 | 0 | 0 | 0 | e | 0 | 0 |
| III | READ TRACK | 1 | 1 | 1 | 0 | 0 | 1 | 0 | s |
| III | WRITE TRACK | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| IV | FORCE INTERRUPT | 1 | 1 | 0 | 1 | i | i | i | i |

**FLAG BIT SUMMARY**

h = 1 Load head at beginning (TRS-80 default loads the head when the motor is started and the drive selected)
h = 0 Do not load head at beginning

v = 1 Verify on last track
v = 0 No verify

rr = Stepping motor rate (bits 0-1) (step rate is for 1 MHz clock in the FDC)
0 0 = 12 msec per track
0 1 = 12 msec per track
1 0 = 20 msec per track
1 1 = 40 msec per track

u = 1 Update TRACK register
u = 0 No update

m = 1 Read or write multiple records
m = 0 Read or write single records

b = 1 IBM standard format (128, 256, 512, or 1024 bytes)
b = 0 Non-standard (length is multiple of 16 up to 4096)

aa = Data Address Mark
0 0 = FB data mark (normal data mark)
0 1 = FA data mark (user defined)
1 0 = F9 data mark (user defined)
1 1 = F8 data mark (deleted data mark)

s = 1 Synchronize to address mark
s = 0 No synchronize

iiii = Interrupt Condition Flags
0000 = Terminate and force not busy with no interrupt
bit 0 = interrupt on not ready to ready transition
bit 1 = interrupt on ready to not ready transition
bit 2 = interrupt on index pulse
bit 3 = interrupt immediately

e = 1 Enable HLT, HLT, and the 10 msec delay
e = 0 Head is assumed engaged and there is no delay

*Table 2. FDC Command Summary*



✓See List of Advertisers on Page 451

read protection. This is why you get error code six if you attempt to read the sector in Basic. You can read the directory from a Basic program if you trap this error and if you change byte 01 in sector 1 (the HIT sector) from 2CH to C4H.

Group III commands include the following: Read Address, Read Track, and Write Track. These commands are used to format a disk and to verify that it can be read. The Read Address command reads the six identification bytes from the disk in the next ID field. These six bytes are:

- Track Address (zero to number of track minus one)
- Side Number (usually reflects the side of the disk; in TRSDOS disks this value is zero)
- Sector Address (usually zero through nine)
- Sector Length
- Two CRC bytes

Read Track reads the entire track of information: the data, the ID bytes and all the gaps. Information is transferred similar to the Read command. The Data Request Bit (DRQ) signals when data has been assembled by the FDC and is ready for transfer.

Write Track is used principally to format the disk by writing appropriate control information to the disk. This command is not normally used to write data to the disk as several bytes have special meaning to the FDC (bytes X'F7' through X'FE'). The use of these special bytes is described in Table 4.

The last command group has only one command. Force Interrupt stops the FDC and forces it to generate an interrupt. If any conditions in the command are set (see Table 2), the command forces the FDC to terminate, drop the BUSY status bit, and interrupt. If none of these conditions are selected, the FDC terminates the current operation and resets BUSY without interrupt.

Disk structure is generally standard IBM format; the disk is formatted using sector multiples of 128 bytes (128,256,512, or 1024 bytes). The non-standard disk is formatted to use a sector having a length that is a multiple of 16 bytes; this length varies from 16 to 4,906 bytes if the track is large enough.

The type of sector to be read or written is determined by the b flag in the command; the length is determined from the sector length field stored in the ID field (remember the information read using a Read Address command?).

**Status for Type I commands**

| BIT | NAME | MEANING |
|---|---|---|
| S7 | Not Ready | The selected drive is not ready. |
| S6 | Protected | The Write Protect tab is on the disk. |
| S5 | Head Loaded | Indicates the head is loaded and engaged. |
| S4 | Seek Error | The desired track was not verified. |
| S3 | CRC Error | There was at least one CRC error on a track verification operation. |
| S2 | Track Zero | The current position of the head is over track zero. |
| S1 | Index | The index mark from the drive has been detected by the FDC. |
| S0 | Busy | The FDC has a command in progress. |

**Status for Type II/III commands**

| BIT | NAME | MEANING |
|---|---|---|
| S7 | Not Ready | The selected drive is not ready. Type II/III commands will not execute if the drive is not ready. |
| S6 | Record Type Write Prot | On a Read record, this bit indicates part of a two bit representation which is indicative of the type of Data Address Mark (DAM) encountered. On a Write record, this bit reflects write protect status. |
| S5 | Record Type Write Fault | On a Read Sector command, this is the second of the bits for type of DAM found. On a Write Track command, it indicates a write fault. |
| S4 | Rec Not Fnd | The sector and/or the track were not found at the current head location. |
| S3 | CRC Error | A CRC error was detected during a read command. |
| S2 | Lost Data | The program did not respond to the DRQ within the time frame of being able to read or write two bytes to the disk. |
| S1 | Data Req | The Data register has been filled when executing a Read command or has been emptied while performing a Write command. |
| S0 | Busy | The FDC is busy with the current command. |

*Table 3. Status Register Summary*

### Diskmap

Diskmap uses little sophisticated coding and no trickery to provide the disk user with a visual display of every file on the disk. Using the NEWDOS80 information, I could have written this utility to default to the parameters identifying each disk at a system boot. This would have made Diskmap incompatible with the other disk operating systems so the user defines the directory track and the number of tracks at assembly time. As Program Listing 1 shows, this version was assembled with the directory default set to track 17 and the number of tracks set at 40.

Diskmap performs a disk map on a systemless disk. If you have a single disk drive system, it becomes very difficult to do even a directory (particularly in NEWDOS80 where a new overlay is loaded for each command entered). Hold the break key down after entering the command name. The program will pause; place the appropriate disk in drive 0 and then press Enter to map the disk.

Clock interrupts must be turned off when attempting any data transfer to or from the disk. Failure to turn off the 25 msec interrupt terminates the disk read or write with the Lost Data bit posted. This means that the FDC attempted to access the Data register twice without an

| Byte | Description |
|---|---|
| F7H | Write CRC bytes |
| F8H-FBH | Write Data Address Marks |
| FCH | Write Index Address Mark |
| FDH | Spare |
| FEH | Write ID Address Mark |

*Table 4. Special Bytes*

# AARDVARK

## TRS-80 COLOR OSI VIC-64 VIC-20 SINCLAIR TIMEX




**TUBE FRENZY**
(by Dave Edson)
This is an almost indescribably fast action arcade game. It has fast action, an all new concept in play, simple rules, and 63 levels of difficulty. All machine code, requires Joysticks. Another great game by Dave Edson. TRS 80 COLOR ONLY. 16k and Joysticks required. $19.95.

**QUEST** – A NEW IDEA IN ADVENTURE GAMES! Different from all the others. Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. 16k TRS-80, TRS-80 Color, and Sinclair. 13K VIC-20. $14.95 each.


**ADVENTURES!!!**
These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are interactive fantasies. It's like reading a book except that you are the main character as you give the computer commands like "Look in the Coffin" and "Light the torch.")

Adventures require 16k on TRS80, TRS80 color, and Sinclair. They require 8k on OSI and 13k on Vic-20. Derelict takes 12k on OSI. $14.95 each.

**CATERPILLAR**
O.K., the Caterpillar does look a lot like a Centipede. We have spiders, falling fleas, monsters traipsing across the screen, poison mushrooms, and a lot of other familiar stuff. COLOR 80 requires 16k and Joysticks. This is Edson's best game to date. $19.95 for TRS 80 COLOR.

**PROGRAMMERS!**
**SEE YOUR PROGRAM IN THIS SPACE!!**
Aardvark traditionally pays the highest commissions in the industry and gives programs the widest possible coverage. Quality is the keyword. If your program is good and you want it presented by the best, send it to Aardvark.

**ESCAPE FROM MARS**
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

**PYRAMID** (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

**HAUNTED HOUSE** (by Bob Anderson)
It's a real adventure – with ghosts and ghouls and goblins and treasures and problems – but it is for kids. Designed for the 8 to 12 year old population and those who haven't tried Adventure before and want to start out real easy.

**DERELICT**
(by Rodger Olsen & Bob Anderson)
New winner in the toughest adventure from Aardvark sweepstakes. This one takes place on an alien ship that has been deserted for a thousand years – and is still dangerous!


**CATCH'EM**
(by Dave Edson)
One of our simplest, fastest, funnest, all machine code arcade games. Raindrops and an incredibe variety of other things come falling down on your head. Use the Joysticks to Catch'em. It's a BALL! – and a flying saucer! – and a Flying Y! – and so on. TRS 80 COLOR. $19.95.

**BASIC THAT ZOOOMMS!!**
**AT LAST AN AFFORDABLE COMPILER!**
The compiler allows you to write your programs in easy BASIC and then automatically generates a machine code equivalent that runs 50 to 150 times faster.

It does have some limitations. It takes at least 8k of RAM to run the compiler and it does only support a subset of BASIC – about 20 commands including FOR, NEXT, END, GOSUB, GOTO, IF, THEN, RETURN, END, PRINT, STOP, USR (X), PEEK, POKE, *, /, +, -, >, <, =, VARIABLE NAMES A-Z, SUBSCRIPTED VARIABLES, and INTEGER NUMBERS FORM 0-64K.

TINY COMPILER is written in BASIC. It generates native, relocatable 6502 or 6809 code. It comes with a 20-page manual and can be modified or augmented by the user. $24.95 on tape or disk for OSI, TRS-80 Color, or VIC.

*Please specify system on all orders*

**ALSO FROM AARDVARK – This is only a partial list of what we carry. We have a lot of other games (particularly for the TRS-80 Color and OSI), business programs, blank tapes and disks and hardware. Send $1.00 for our complete catalog.**

## AARDVARK - 80
## 2352 S. Commerce, Walled Lake, MI 48088
## (313) 669-3110

***Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.*** 107

See List of Advertisers on Page 451

*"I turn interrupts off at the beginning and re-enable interrupts at the end. . ."*

intervening access to the Data register by the computer. I turn interrupts off at the beginning and re-enable interrupts at the end as I do not use the display clock. Disable interrupts before each disk I/O operation and re-enable interrupts before each exit or the exit from the read/write routine.

Next, determine the drive number (drive 0 always for the break key pause) and from this the proper select bit. The select bit is always 2**N (N = 0,1,2,3). The DOS command buffer is scanned for a X'OD' following Diskmap; if none is there, it checks for a blank. The drive number is then checked for validity.

After the proper drive has been determined, the FDC is issued a Reset to Not Busy command. The drive is selected and a loop is entered waiting for a change in the Index pulse as the drive rotates the disk. If approximately two full revolutions pass without a change in the Index pulse, the drive is declared not ready and the utility terminates in error.

Next, the user must reselect the drive and establish the head position. The FDC remembers the head location (Track register) for only one drive at a time; this is only valid if the update flag was used when moving the head last. The program performs a Restore to Track Zero command using the 20 msec/track stepping time. This moves the head to track zero and resets the Track register to zero. Next, the Data register is loaded with the number corresponding to the Directory Track number (default value is 17 decimal) and a Seek command moves the head to track 17 on the selected drive. In lines 116–134 (see the Listing) the Push/Pop sequence is a delay between use of the X'37EC' port to address the Command register and then address the Status register. The Bit instruction tests the Busy bit to determine when the operation is complete.

Lines 135-138 establish conditions necessary to perform the sector read loop. Line 139 issues the Read Record command (see Table 2); the heart of the read loop is lines 147-153. The DRQ bit is tested to see if a data byte has been assembled, ready for the computer to store (DRQ = 1). If DRQ is not true ("true" refers to single bits equaling one), then the Busy bit is checked. If Busy is still true, continue to status DRQ until a byte is ready. When DRQ is true, a byte has been assembled and may be stored in the buffer.

When DRQ and Busy are both false, one sector has been read and the FDC is no longer busy. The Status word is examined for any errors detected on the read. No attempt is made to examine the status word for information concerning the Data Address Marks just read. I have assumed that the directory track was read and did not check for the appropriate DAM (a Directory Track is read-protected by writing the sectors with a DAM of X'FA' rather than the IBM normal of X'FB').

The disk sector read routine is repeated nine more times to copy the entire directory track to the buffer. Beginning at the address specified by Buffer is the GAT sector of the directory. The GAT sector contains the Granule Allocation Table beginning at offset byte 0; the Track Locked Out Table beginning at offset byte 60H; and the disk name and creation date starting at offset byte D0H and D8H respectively. This information is used in lines 170–232 to build the subtitle line for the video screen.

The string array MAP contains 80 entries corresponding to each granule; each entry is 13 bytes long. This allows the map to be built prior to display using filenames in the form of FILESPEC/EXT. This array must be

# SCREEN KLEEN™

## PUTS THE POWER IN YOUR HAND TO REMOVE COMPUTER DUST

- Printer paper dust
- Dust data processing equipment
- Dust your computer room "Clean Room" clean

**2 CLOTHS FOR $2.95 PPD**

VISA • MASTER • CHECK

**MWB INDUSTRIES, INC.**
**2013 FRANKLIN STREET**
**DETROIT, MICHIGAN 48207**
**(313) 259-1104**

DEALERS INQUIRIES INVITED ✓90

---

Tired of just PEEKing around???
Try **DISaDATa**, The Disassembler That Even Tracks Down DATA!!!

For **your** disassembler, wouldn't you like one that --

- Outputs "
```
H37F7   DEFM   'Cass? '
        DEFB   03H          "
```
instead of the meaningless "equivalent",
```
"  AB25   LD    B,E
          LD    H,C
          LD    (HL),E
          LD    (HL),E
          CCF
          JR    NZ,$+5       ", etc.?
```
- **Automatically** identifies such data areas?
- Outputs fully labeled EDTASM*-format code to display, printer, cassette or disk?
- Relocates itself to any desired RAM area, up/down one byte or as many as required?
- Runs on TRS*-80 Model I (Level II) or Model III, 16K or more, cassette or disk?

To order **DISaDATa** phone (513) 435-4480 (M-F, 9 a.m. - 5 p.m. EST), or send check or money order for:

#1354-10, (Cassette) .................... $24.95 + 1.50 shipping
#1354-20, (Model III diskette) ............ $29.95 + 1.50 shipping

to the address below.
(Ohio residents please add 6% sales tax.)

*Registered Trademarks, Tandy Corporation

Pro/Am Software

Professional Software for both Novice and Expert

**Pro/Am Software**
220 Cardigan Road
Centerville, Ohio 45459 ✓260

We accept Visa/Mastercard. When ordering via card include card number, merchant number, expiration date, your name, address and phone number.

---

# HUMBUG!

(to high prices)

## TRS 80™* Model III 48 K 2 Disk Drive

- ☑ Tandon drives SS/DD
- ☑ Astec Power Supply
- ☑ 40/80 track support on controller board
- ☑ TRS DOS
- ☑ 120 day warranty

**$1695 complete**

*Trade Mark Tandy Corp

ASHLAND COMPUTER
1716 Wilshire Blvd.
Ashland, KY 41101
(606) 325-2210
Call 10-6 M-F

✓514

---

# We've done it again!

*We said we had the best system, and now... 8" double density is here for Model III's.*

## Check out these Features -

✓568K of storage on a single sided 8 inch drive, when used with our controller & speedup MOD and DOS plus 3.4. ✓The maximum storage capacity of your Model 3 will be 4.544 Megabytes - without a hard disk. ✓One 5¼ single sided and one 8 inch single sided drive will store 676K, with a full blown DOS plus 3.4. ✓One 5¼ single sided and one 8 inch double sided drive will store 1.244 Megabytes, with a full blown DOS plus 3.4. ✓Mix and match 5¼ and 8 inch drives, in any density, single or double sided. - All under software control.

**We will also be able to provide DOS plus 3.4 on 8 inch diskettes. (Final price being negotiated).**

1. Speedup for 8 inch drives only **$89.00** 2. General purpose speed up, which also works with 8 inch drives **$129.00**

## Here's 3 good reasons to buy our system -

**1. VERSATILITY**

Our own proven, dependable drive controller will give you the hardware to run almost any drive configuration you can imagine.

- 5¼ and 8 inch drives in any combination
- Internal/external drive combinations. Up to all 4 drives external
- **Single - and dual - headed drive combinations**

**2. QUALITY**

- We use quality drives - MPI and now new generation Siemens.
- We construct our own disk drive installation kits of heavy steel - no shielding Problems
- We are now using a field proven switching supply

**3. PRICE**

| | |
|---|---|
| • Drive controller kit with **first** drive | $679 |
| • Second (internal drive) | $279 |
| • Third drive (5¼ floppy with cable) | $359 |
| • Fourth drive (5¼ floppy) | $329 |

**If these sound like good deals, but you don't have the Model III yet - we've got the answer:**

**TRS-80 Model III 48K TRS DOS, 2 drives** $1899

**DEALERS, WE HAVE DEALS FOR YOU TOO.... WRITE OR PHONE -**

master charge VISA

Process Control Technology ✓288
8030 Lorraine Ave., Suite 328 • Stockton, California 95210 • (209) 952-6576

zeroed before proceeding.

To build the array Map, lines 260-306 read all non-null entries in the directory and reconstruct the filespec for each in the locations in MAP corresponding to the extent information in each File Primary Directory Entry (FPDE).

Interspersed with the FPDEs in the directory may be File Extension Directory Entries (FXDE). Table 5 explains the byte/bit structure for FPDEs and FXDEs found in the directory. FXDEs are an additional area for the disk operating system to keep information about the FPDEs. The last entry in the chain of information concerning the file's disk location must start with an X'FF' byte. If the DOS must have more than four entries to contain the information about a file, the last entry in the FPDE starts with X'FE' and the second byte is a Directory Entry Code (DEC) describing the location of the corresponding FXDE in the directory. When an FXDE is detected in line 298, the IX register is set to the appropriate value in the buffer corresponding to the location of the FXDE. This is per-

| | | |
|---|---|---|
| Byte 0 | | Bit record for file attributes |
| | Bit 7— | 1 = FXDE, 0 = FPDE |
| | Bit 6— | 1 = System File, 0 = Not System File |
| | Bit 5— | Not used |
| | Bit 4— | 1 = Directory entry is assigned to a file<br>0 = Directory entry is free |
| | Bit 3— | 1 = invisible file, 0 = Visible file |
| | Bits 2-0— | Protection attributes for each file<br>111—No access is allowed<br>110—Execution access only<br>101—Read/Execute access<br>100—Write/Read/Execute access<br>011—Not used<br>010—Rename/Write/Read/Execute access<br>001—Kill/Rename/Write/Read/Execute access<br>000—Full access |
| Byte 1 | | Not used in an FPDE; in an FXDE, this byte is used as a DEC to point back to its FPDE. |
| Byte 2 | | Not used in an FPDE or FXDE. |
| Byte 3 | | Used as the end of file byte pointing to the relative byte in the sector that is the physical end of this file. |
| Byte 4 | | Logical record length (0 = 256 bytes). |
| Bytes 5-C | | File Name, left-justified blank-filled format. |
| Bytes D-F | | File Name Extension in same format. |
| Bytes 10-11 | | Two byte hash-code value for the update password. |
| Bytes 12-13 | | Two byte hash-code value for the access password. |
| Bytes 14-15 | | This is the EOF sector or the number of the sectors in the file. If Byte 3 is 00H then this is the actual sector count. If Byte 3 is not 00H then the actual sector count is this value plus 1. |
| Bytes 16-17<br>18-19<br>1A-1B<br>1C-1D<br>1E-1F | | These last five pairs of bytes form extent information which locate the file on the disk. Each 2-byte extent has the track number for the file, the starting granule, and number of contiguous granules for this extent. If more extents are needed to describe a file than are available in the FPDE, the last extent is coded to point to an FXDE for additional extent storage.<br>Consider the following Extent:<br>1C 24<br>The first byte of this extent is the track number of the start of the file in this extent, e.g. track 1CH or 28. The second byte is two packed values which are the granule number (0 or 1) and the number of contiguous granules in this extent less one. In this example 24H is 0010 0100 binary. The left three bits represent the granule number and the right five bits represent the count of contiguous granules; 001 = granule 1 or sector 5 and 00100 = 4 + 1 or five contiguous grans in this extent. If the first byte is FFH, this signifies the end of the file extents. A first byte of FEH signifies a pointer to an FXDE follows in the second byte and may be interpreted as follows: The left three bits of the second byte multiplied by 32 is the offset of the FXDE in the directory sector described by the right three bits. These three bits are the offset from the HIT sector; add two to them to reflect the right sector on the directory track. The extent FE 63 is interpreted as follows: 63H equals 0110 0011 binary. The right three bits plus two tell you that the FXDE is in sector 5 on the directory track and the FXDE is in entry 011 binary*20H or offset address 60H. The only bytes used in constructing an FXDE are bytes 0, 1, and 16 through 1F; all other bytes are zeroed. |

*Table 5. FPDC and FXDE directories*

formed by the code in lines 464-474.

Having constructed the Map array, the screen is cleared and the prototype structure is displayed by the code in lines 312-343. Once this is completed, the PAGES and TRK counters step through the Map array and display the disk map 10 tracks at once (see lines 348-447).

### Diskmap On Disk

If you have an editor/assembler capable of reading and writing disk source files, enter the source code and assemble the program to disk.

If you are not so fortunate, enter the code into the tape version of the editor/assembler and assemble and write the object code to tape. When you have created an object tape, use LMOFFSET or Tapedisk to load the tape to disk. The disk Dump command for Tapedisk is "F DISKMAP/CMD:d 5300 5846 5300".

### Compatibility

This utility runs under NEWDOS80, NEWDOS 2.1, and TRSDOS. Diskmap uses a few ROM routines that are relatively independent of the TRS-80 ROM configuration.

This useful program now resides on my disk along with DIRCHECK and SUPERZAP as one of my most-used utilities.■

*Program Listing*

```
                00001 ;***********************************************************
                00002 ;*                                                         *
                00003 ;*                    >> DISKMAP <<                        *
                00004 ;*                                                         *
                00005 ;*                                                         *
                00006 ;* <DISKMAP> IS A UTILITY WHICH WILL READ THE DIRECTORY    *
                00007 ;* OF ANY DISK AND DISPLAY THE CONTENTS OF THE DISK IN     *
                00008 ;* A MAP BY GRANULES.  ALSO DISPLAYED IS INFORMATION       *
                00009 ;* SUCH AS DISK NAME, CREATION DATE, DRIVE NUMBER, THE     *
                00010 ;* NUMBER OF GRANULES FREE AND THE NUMBER OF TRACKS        *
                00011 ;* LOCKED OUT.  THIS UTILITY MAY BE USED WITH A SINGLE     *
                00012 ;* DISK SYSTEM TO MAP A DISKETTE HAVING NO SYSTEM FILES    *
                00013 ;* USING THE "PAUSE" OPTION SINCE <DISKMAP> HAS ITS OWN    *
                00014 ;* DISK I/O ROUTINES.                                      *
                00015 ;*                                                         *
                00016 ;*            WRITTEN BY J.B. HARRELL, 03/07/81            *
                00017 ;*                 30 KEKLICO COURT WEST                   *
                00018 ;*            CHARLESTON, SOUTH CAROLINA, 29408            *
                00019 ;*                                                         *
                00020 ;***********************************************************
                00021
                00022
5300            00023         ORG     5300H
                00024
                00025
5300 F3         00026 DSKMAP  DI                      ;TURN OFF INTERRUPTS
5301 3A4038     00027         LD      A,(3840H)       ;LOAD BREAK KEY ROW TO
5304 E604       00028         AND     4               ;TEST FOR "PAUSE" OPTION
5306 2822       00029         JR      Z,DSKMP1        ;NO PAUSE
5308 CDC901     00030         CALL    CLS             ;CLEAR THE SCREEN
530B 21163C     00031         LD      HL,VIDEO+22
530E 222040     00032         LD      (CURSOR),HL
5311 217B56     00033         LD      HL,TITLE        ;WRITE TITLE HEADER
5314 CD3956     00034         CALL    OUTSTR
5317 213857     00035         LD      HL,PRMPT0       ;WRITE PROMPT REQUESTING
531A CD3956     00036         CALL    OUTSTR          ;DISKETTE MOUNT
531D CD4900     00037 DSKMP0  CALL    GETKEY          ;WAIT FOR ANY KEY
5320 FE0D       00038         CP      CR              ;TEST FOR <ENTER>
5322 20F9       00039         JR      NZ,DSKMP0       ;WAIT FOR <ENTER>
5324 1601       00040         LD      D,1             ;SET DEFAULT DRIVE TO
5326 1E30       00041         LD      E,'0'           ;DRIVE 0 AND GO
5328 1832       00042         JR      NOPARS
532A 1601       00043 DSKMP1  LD      D,1             ;DEFAULT DRIVE SELECT BIT
532C 1E30       00044         LD      E,'0'           ;DEFAULT DRIVE NUMBER
532E 3A1F43     00045         LD      A,(CMDBUF+7)    ;SEE IF PARS OR NOT
5331 FE0D       00046         CP      CR              ;BY TEST FOR A CR
5333 2827       00047         JR      Z,NOPARS
5335 FE20       00048         CP      ' '             ;HAS TO HAVE ONE SPACE
5337 201A       00049         JR      NZ,ERROR
5339 3A2043     00050         LD      A,(CMDBUF+8)    ;GET DRIVE PARAMETER
533C 5F         00051         LD      E,A             ;SET IN OUTPUT REG
533D FE30       00052         CP      '0'             ;TEST FOR 0, 1, 2, 3
533F 281B       00053         JR      Z,NOPARS
5341 CB22       00054         SLA     D
5343 FE31       00055         CP      '1'             ;(D) HAS DRIVE SELECT BIT
5345 2815       00056         JR      Z,NOPARS        ;WHICH IS 2**N WHERE N
5347 CB22       00057         SLA     D               ;IS THE DRIVE NUMBER
5349 FE32       00058         CP      '2'
534B 280F       00059         JR      Z,NOPARS
534D CB22       00060         SLA     D
534F FE33       00061         CP      '3'
5351 2809       00062         JR      Z,NOPARS
```

*Listing continues*

*Listing continued*

```
5353 219156   ERROR   LD    HL,ERR1        ;ILLEGAL PARAMETER FOUND
5356 CD3956           CALL  OUTSTR
5359 C3EE55           JP    DOSEXT         ;REENABLE INTERRPUT - RET

535C 7A       NOPARS  LD    A,D            ;SAVE SELECT BIT
535D 321857           LD    (SELBIT),A
5360 7B               LD    A,E            ;SAVE DRIVE NUMBER IN
5361 32BA56           LD    (DRNR),A       ;VARIOUS MESSAGES
5364 32D156           LD    (DRVNR),A

              ; NOW THAT THE PARAMETER HAS BEEN DECODED, THE NEXT TASK
              ; IS TO SELECT THE DRIVE AND TO SEE IF IT IS READY BY
              ; READING THE INDEX PULSE BIT AND MAKING SURE THAT IT
              ; CHANGES PRIOR TO TWO FULL DISKETTE REVOLUTIONS.  IF
              ; IT DOES - OK, IF NOT - THEN DISKETTE IS NOT READY FOR
              ; A VARIETY OF REASONS.

5367 3ED0             LD    A,0D0H         ;FDC COMMAND - QUIT OPS
5369 32EC37           LD    (CMDREG),A     ;TERMINATE, RESET "BUSY"
536C 3A1857           LD    A,(SELBIT)     ;GET DRIVE SELECTED
536F 32E137           LD    (SELECT),A     ;SELECT IT AND START MTR
5372 010080           LD    BC,8000H
5375 CD6000           CALL  DELAY          ;WAIT ONE-HALF SECOND
5378 3A1857           LD    A,(SELBIT)     ;RESELECT DRIVE
537B 32E137           LD    (SELECT),A
537E E5               PUSH  HL
537F E1               POP   HL             ;WASTE TIME
5380 E5               PUSH  HL
5381 E1               POP   HL
5382 3AEC37           LD    A,(STATUS)     ;READ STATUS REGISTER
5385 E602             AND   2              ;INDEX PULSE BIT
5387 5F               LD    E,A            ;SAVE IT
5388 011027           LD    BC,10000       ;COUNT FOR APPRX 2 REVS
538B 3A1857           LD    A,(SELBIT)
538E 32E137           LD    (SELECT),A
5391 3AEC37   TEST    LD    A,(STATUS)     ;GET STATUS WORD
5394 E602             AND   2
5396 BB               CP    E              ;SEE IF INDX BITS ARE =
5397 200E             JR    NZ,READY       ;NO - IT MUST BE OK
5399 0B               DEC   BC             ;DECR COUNT
539A 78               LD    A,B
539B B1               OR    C
539C 20F3             JR    NZ,TEST
539E 21AE56           LD    HL,ERR2        ;DRIVE X IS NOT READY
53A1 CD3956           CALL  OUTSTR
53A4 C3EE55           JP    DOSEXT

              ; NOW THAT THE DRIVE HAS BEEN DETERMINED TO BE IN AN
              ; OPERATIONAL STATUS, RESTORE THE DRIVE TO TRACK 0 TO
              ; LET THE FDC KNOW WHERE THE HEAD IS AND THEN SEEK THE
              ; "DIRTRK".  READ 10 SECTORS FROM "DIRTRK" AND STORE IN
              ; BUFFER FOR LATER USE.

53A7 3A1857   READY   LD    A,(SELBIT)     ;RESELECT DRIVE
53AA 32E137           LD    (SELECT),A
53AD 21EC37           LD    HL,CMDREG      ;SET UP ROUTINE
53B0 360E             LD    (HL),0EH       ;RESTORE - 20MS/TRACK
53B2 E5               PUSH  HL
53B3 E1               POP   HL
53B4 E5               PUSH  HL
53B5 E1               POP   HL
53B6 CB46     READ1   BIT   0,(HL)         ;TEST FOR BUSY BIT
53B8 20FC             JR    NZ,READ1       ;WAIT TILL NOT BUSY
53BA 3E11             LD    A,DIRTRK       ;SET UP SEEK
53BC 32EF37           LD    (DATARG),A
53BF 361E             LD    (HL),1EH       ;SEEK - 20MS/TRACK
53C1 E5               PUSH  HL
53C2 E1               POP   HL
53C3 E5               PUSH  HL
53C4 E1               POP   HL
53C5 CB46     READ2   BIT   0,(HL)
53C7 20FC             JR    NZ,READ2       ;WAIT FOR NOT BUSY
53C9 010060           LD    BC,BUFFER      ;BC HAS BUFFER ADDR
53CC 11EF37           LD    DE,DATARG
53CF AF               XOR   A
53D0 32EE37           LD    (SECTOR),A     ;START AT SECTOR 0
53D3 368C     READ3   LD    (HL),8CH       ;READ SECTOR
53D5 E5               PUSH  HL             ;WASTE TIME
53D6 E1               POP   HL
53D7 E5               PUSH  HL
53D8 E1               POP   HL
53D9 1803             JR    READ5
53DB 0F       READ4   RRCA                 ;TEST BUSY BIT
53DC 300A             JR    NC,READ6       ;NOT BUSY - DONE OR ERROR
53DE 7E       READ5   LD    A,(HL)         ;GET STATUS WORD
53DF CB4F             BIT   1,A            ;TEST FOR "DRQ"-DATA REQ
53E1 28F8             JR    Z,READ4        ;NO DRQ - CHECK BUSY
53E3 1A               LD    A,(DE)         ;DRQ - GET NEXT BYTE
53E4 02               LD    (BC),A         ;STORE BYTE
53E5 03               INC   BC
53E6 18F6             JR    READ5          ;GO FOR MORE
53E8 7E       READ6   LD    A,(HL)         ;FDC STATUS WORD
53E9 E61C             AND   1CH            ;TEST FOR REC NOT FND,
53EB 2809             JR    Z,READ7        ;CRC ERR, LOST DATA
53ED 211957           LD    HL,DIRERR      ;DIRECTORY READ ERROR
53F0 CD3956           CALL  OUTSTR
53F3 C3EE55           JP    DOSEXT
53F6 3AEE37   READ7   LD    A,(SECTOR)     ;GET CURRENT SECTOR NR
53F9 3C               INC   A
53FA 32EE37           LD    (SECTOR),A     ;UPDATE IT
```

*Listing continues*

✓See List of Advertisers on Page 451

*Listing continued*

```
53FD FE0A           CP      10          ;TEST FOR FULL TRACK READ
53FF 20D2           JR      NZ,READ3

             ; THE DIRECTORY HAS BEEN READ TO THE BUFFER AND NOW
             ; READ THROUGH THE "GAT" SECTOR (SECTOR 0) AND SET UP THE
             ; TITLE LINE FOR THE SCREEN.

5401 010800         LD      BC,8        ;BYTE COUNT FOR NAME
5404 11D556         LD      DE,DKNAME
5407 21D060         LD      HL,GAT+0D0H ;START OF DISK NAME
540A EDB0           LDIR
540C 010800         LD      BC,8        ;LENGTH OF DATE
540F 21D860         LD      HL,GAT+0D8H ;START OF CREATION DATE
5412 11DF56         LD      DE,DATE
5415 EDB0           LDIR
5417 0E00           LD      C,0         ;COUNT TRACKS LOCKED OUT
5419 0628           LD      B,NRTRKS
541B 216060         LD      HL,GAT+60H  ;START OF TRK LOCKED OUT
541E 7E      TRKS   LD      A,(HL)
541F FEFC           CP      0FCH        ;CODE FOR NOT LOCKED OUT
5421 2801           JR      Z,TRKS2
5423 0C             INC     C
5424 23      TRKS2  INC     HL
5425 10F7           DJNZ    TRKS        ;LOOP FOR FULL DISK WORTH
5427 69             LD      L,C         ;SET UP CONVERSION
5428 60             LD      H,B         ;H=0
5429 CD9A0A         CALL    HLACC       ;(HL)-->4121H, 2-->40AFH
542C CDBD0F         CALL    CVTASC      ;CONVERT NUMBER TO STRING
542F 23             INC     HL          ;SKIP LEADING BLANK
5430 7E             LD      A,(HL)      ;GET FIRST DIGIT
5431 110857         LD      DE,TLO
5434 12             LD      (DE),A      ;STORE IN HEADING
5435 23             INC     HL
5436 13             INC     DE
5437 7E             LD      A,(HL)      ;GET NEXT DIGIT
5438 B7             OR      A           ;TEST FOR TERMINATION
5439 2801           JR      Z,TRKS3
543B 12             LD      (DE),A
543C 0E00    TRKS3  LD      C,0         ;COUNT OF FREE GRANULES
543E 0628           LD      B,NRTRKS
5440 210060         LD      HL,GAT      ;START OF GRAN ALLOC TBL
5443 7E      GRANS0 LD      A,(HL)      ;GET NEXT BYTE
5444 FEFC           CP      0FCH        ;BOTH FREE GRANS
5446 2004           JR      NZ,GRANS1
5448 0C             INC     C
5449 0C             INC     C
544A 180C           JR      GRANS3
544C FEFD    GRANS1 CP      0FDH        ;CODE FOR ONE FREE
544E 2003           JR      NZ,GRANS2
5450 0C             INC     C
5451 1805           JR      GRANS3
5453 FEFE    GRANS2 CP      0FEH        ;CODE FOR THE OTHER FREE
5455 2001           JR      NZ,GRANS3
5457 0C             INC     C
5458 23      GRANS3 INC     HL          ;BUMP POINTER
5459 10E8           DJNZ    GRANS0      ;LOOP FOR FULL DISK
545B 69             LD      L,C         ;SET UP CONVERSION
545C 60             LD      H,B         ;H=0
545D CD9A0A         CALL    HLACC       ;(HL)-->BASIC ACCUM
5460 CDBD0F         CALL    CVTASC      ;CONVERT TO STRING
5463 23             INC     HL
5464 11F456         LD      DE,GRANS
5467 7E             LD      A,(HL)      ;GET FIRST DIGIT
5468 12             LD      (DE),A
5469 23             INC     HL
546A 13             INC     DE
546B 7E             LD      A,(HL)      ;GET NEXT DIGIT
546C B7             OR      A           ;TEST FOR TERMINATOR
546D 2801           JR      Z,GRANS4
546F 12             LD      (DE),A

             ; ZERO MAP ARRAY AND VARIOUS VARIABLES

5470 21465B  GRANS4 LD      HL,MAP      ;SET UP MAP STRING ARRAY
5473 11475B         LD      DE,MAP+1
5476 010F04         LD      BC,1039     ;COUNT FOR 13*80 ARRAY
5479 3600           LD      (HL),0      ;PROPOGATION CHARACTER
547B EDB0           LDIR
547D AF             XOR     A
547E 323657         LD      (PAGES),A
5481 32D157         LD      (TRK),A

             ; START TO BUILD THE MAP STRING ARRAY OF FILESPEC'S AT
             ; THE LOCATION CORRESPONDING TO THE LOCATION OF THE FILE
             ; ON THE DISKETTE.  THIS IS DONE BY LOOKING AT THE FIRST
             ; BYTE OF EACH FILE DIRECTORY ENTRY (FDE) STARTING WITH
             ; THE SECOND SECTOR OF THE DIRECTORY.  IF IT IS 0 (WHICH
             ; MEANS THE FDE IS EMPTY) OR 90H (WHICH MEANS THAT THE
             ; FDE IS REALLY A FILE EXTENSION DIRECTORY ENTRY (FXDE)
             ; OF AN ALREADY EXISTING FPDE) THEN IGNORE IT.  LOOK AT
             ; THE FDE ENTRIES FOR ALL THE REST AS THEY ARE THE FILE
             ; PRIMARY DIRECTORY ENTRIES (FPDE).  UTILIZE THE "EXTENT"
             ; INFORMATION IN EACH FPDE (AND CORRESPONDING FXDE, IF
             ; NECESSARY) TO STORE THE FILESPEC IN THE MAP TABLE IN
             ; THE LOCATIONS CORRESPONDING TO TRACK/ASSIGNED GRANULES

5484 210062         LD      HL,FDES     ;START OF FDE'S IN BUFFER
5487 E5      SCAN   PUSH    HL          ;SAVE POINTER ON STACK
5488 7E             LD      A,(HL)      ;GET BIT RECORD OF FDE
```

*Listing continues*

*Listing continued*

```
5489 B7               OR    A                  ;TEST FOR NULL
548A 2857             JR    Z,SCNNXT
548C FE90             CP    90H                ;TEST FOR FXDE ENTRY
548E 2853             JR    Z,SCNNXT
5490 CD4256           CALL  FLSPEC             ;BUILD FILESPEC TO FILENM
5493 DDE1             POP   IX                 ;FDE POINTER
5495 DDE5             PUSH  IX
5497 011600    SCAN1  LD    BC,22              ;OFFSET OF EXTENT #1
549A DD09             ADD   IX,BC
549C DD7E00           LD    A,(IX)             ;GET TRACK #
549F FEFF             CP    0FFH               ;TEST FOR END OF XTNT
54A1 2840             JR    Z,SCNNXT
54A3 CD0856           CALL  STORIT             ;STORE FILESPEC
54A6 DD23             INC   IX
54A8 DD23             INC   IX
54AA DD7E00           LD    A,(IX)             ;TRACK FOR EXTENT #2
54AD FEFF             CP    0FFH               ;TEST FOR END
54AF 2832             JR    Z,SCNNXT
54B1 CD0856           CALL  STORIT
54B4 DD23             INC   IX
54B6 DD23             INC   IX
54B8 DD7E00           LD    A,(IX)             ;TRACK FOR EXTENT #3
54BB FEFF             CP    0FFH
54BD 2824             JR    Z,SCNNXT
54BF CD0856           CALL  STORIT
54C2 DD23             INC   IX
54C4 DD23             INC   IX
54C6 DD7E00           LD    A,(IX)             ;TRACK FOR EXTENT #4
54C9 FEFF             CP    0FFH
54CB 2816             JR    Z,SCNNXT
54CD CD0856           CALL  STORIT
54D0 DD23             INC   IX
54D2 DD23             INC   IX
54D4 DD7E00           LD    A,(IX)             ;TRACK FOR EXTENT#5
54D7 FEFF             CP    0FFH               ;TEST FOR END OF EXTENTS
54D9 2808             JR    Z,SCNNXT
54DB FEFE             CP    0FEH               ;TEST FOR FXDE ENTRY NEXT
54DD CAF255           JP    Z,FXDE             ;PROCESS IT
54E0 CD0856           CALL  STORIT
54E3 E1        SCNNXT POP   HL                 ;GET POINTER TO FDE
54E4 012000           LD    BC,32              ;LENGTH OF ONE FDE
54E7 09               ADD   HL,BC              ;BUMP POINTER
54E8 3E6A             LD    A,BUFFER+0A00H<-8  ;TEST FOR END OF BUFFER
54EA BC               CP    H
54EB 209A             JR    NZ,SCAN

               ; NOW THAT THE MAP ARRAY HAS BEEN CONSTRUCTED, DISPLAY
               ; THE DISK GRANULE MAP ON THE VIDEO IN FOUR PAGES OF
               ; TEN TRACKS PER PAGE WITH A HEADER AT TOP EACH TIME.

54ED CDC901    DISP   CALL  CLS                ;CLEAR THE SCREEN
54F0 21163C           LD    HL,VIDEO+22        ;START OF TITLE
54F3 222040           LD    (CURSOR),HL
54F6 217B56           LD    HL,TITLE           ;TITLE LINE
54F9 CD3956           CALL  OUTSTR
54FC 21CB56           LD    HL,DKMSG           ;DRIVE,NAME,DATE,ETC.
54FF CD3956           CALL  OUTSTR
5502 21913C           LD    HL,VIDEO+145       ;START OF GRANULE MAP
5505 222040           LD    (CURSOR),HL
5508 21D557           LD    HL,LINE3
550B CD3956           CALL  OUTSTR
550E 21D13C           LD    HL,VIDEO+209       ;START OF LINE4
5511 11D23C           LD    DE,VIDEO+210
5514 012E00           LD    BC,46
5517 368C             LD    (HL),140           ;GRAPHICS BLOCK
5519 EDB0             LDIR                     ;DRAW LINE
551B 3EBF             LD    A,191              ;GRAPHICS BLOCK
551D 32D13C           LD    (VIDEO+209),A      ;DIVIDE COLUMNS
5520 32E83C           LD    (VIDEO+232),A
5523 32FF3C           LD    (VIDEO+255),A
5526 3E0A             LD    A,10               ;NUMBER OF "LINEN"'S
5528 11003D           LD    DE,VIDEO+256       ;START OF DISPLAY
552B 014000    DISP1  LD    BC,64              ;BYTE COUNT
552E 210558           LD    HL,LINEN
5531 EDB0             LDIR                     ;WRITE ONE LINE
5533 3D               DEC   A                  ;TEST FOR END
5534 20F5             JR    NZ,DISP1
5536 21913F           LD    HL,VIDEO+913       ;BOTTOM LINE
5539 11923F           LD    DE,VIDEO+914
553C 012E00           LD    BC,46
553F 3683             LD    (HL),131           ;LINE BLOCK
5541 EDB0             LDIR

               ; SCREEN IS FORMATTED, NOW WRITE 10 TRACKS WORTH OF
               ; FILESPEC'S TO THE FORMATTED SCREEN

5543 2A3657    DISP0  LD    HL,(PAGES)         ;GET CURRENT PAGE
5546 E5               PUSH  HL
5547 C1               POP   BC                 ;PAGE*10
5548 29               ADD   HL,HL
5549 29               ADD   HL,HL
554A 09               ADD   HL,BC
554B 29               ADD   HL,HL              ;PAGE*10+TRK=TRACK #
554C ED5BD157         LD    DE,(TRK)           ;CURRENT TRK IN PAGE
5550 19               ADD   HL,DE
5551 D5               PUSH  DE                 ;SAVE CURRENT TRK NR
5552 22D357           LD    (TRKNR),HL         ;SAVE COMPUTED TRK NR
5555 CD9A0A           CALL  HLACC              ;INTEGER HL-->ACCUM
5558 CDBD0F           CALL  CVTASC             ;CONVERT FOR OUTPUT
```

*Listing continues*

*Listing continued*

```
555B D1                        POP    DE               ;GET TRK NR FOR INDEX
555C EB                        EX     DE,HL            ;DE=PTR TO STRING,HL=TRK
555D 29                        ADD    HL,HL            ;*2
555E 29                        ADD    HL,HL            ;*4
555F 29                        ADD    HL,HL            ;*8
5560 29                        ADD    HL,HL            ;*16
5561 29                        ADD    HL,HL            ;*32
5562 29                        ADD    HL,HL            ;*64
5563 010D3D                    LD     BC,VIDEO+256+13  ;BASE FOR TRACK NUMBER
5566 09                        ADD    HL,BC
5567 E5                        PUSH   HL
5568 1A               DISP2    LD     A,(DE)           ;GET STRING VALUE
5569 B7                        OR     A                ;TEST FOR 0 BYTE TERMINAL
556A 2805                      JR     Z,DISP3
556C 77                        LD     (HL),A
556D 13                        INC    DE
556E 23                        INC    HL
556F 18F7                      JR     DISP2

5571 2AD357           DISP3    LD     HL,(TRKNR)       ;GET TRACK NR AS INDEX
5574 29                        ADD    HL,HL            ;DOUBLE TO GET GRANULE
5575 E5                        PUSH   HL
5576 D1                        POP    DE
5577 29                        ADD    HL,HL            ;*2
5578 19                        ADD    HL,DE            ;*3
5579 29                        ADD    HL,HL            ;*6
557A 29                        ADD    HL,HL            ;*12
557B 19                        ADD    HL,DE            ;*13
557C 114658                    LD     DE,MAP           ;BASE ADDRESS
557F 19                        ADD    HL,DE            ;BASE+GRANULE*13
5580 D1                        POP    DE               ;LINE ADDR
5581 D5                        PUSH   DE               ;KEEP IT
5582 E5                        PUSH   HL               ;SAVE MAP ARRAY ADDR
5583 EB                        EX     DE,HL
5584 010A00                    LD     BC,10            ;TAB OVER TO FILESPEC
5587 09                        ADD    HL,BC
5588 1A               DISP30   LD     A,(DE)           ;GET NEXT FILE CH
5589 B7                        OR     A                ;TEST FOR 0 TERMINAL
558A 2805                      JR     Z,DISP4
558C 77                        LD     (HL),A           ;STORE IT IN LINE
558D 13                        INC    DE
558E 23                        INC    HL
558F 18F7                      JR     DISP30
5591 012100           DISP4    LD     BC,33            ;TAB TO 2ND GRAN ON LINE
5594 D1                        POP    DE               ;STRING ARRAY ADDRESS
5595 E1                        POP    HL               ;LINE ADDRESS
5596 09                        ADD    HL,BC
5597 EB                        EX     DE,HL
5598 010D00                    LD     BC,13            ;BASE+13*(GRANULE+1)
559B 09                        ADD    HL,BC
559C 7E               DISP5    LD     A,(HL)           ;GET FILESPEC NEXT CH
559D B7                        OR     A                ;TEST FOR TERMINAL
559E 2805                      JR     Z,DISP6
55A0 12                        LD     (DE),A
55A1 23                        INC    HL
55A2 13                        INC    DE
55A3 18F7                      JR     DISP5
55A5 3AD157           DISP6    LD     A,(TRK)          ;INCR CURRENT TRK ON PAGE
55A8 3C                        INC    A
55A9 32D157                    LD     (TRK),A
55AC FE0A                      CP     10               ;TEST FOR 10 THIS PAGE
55AE 2093                      JR     NZ,DISP0         ;FILL UP SCREEN
55B0 3A3657                    LD     A,(PAGES)
55B3 3C                        INC    A
55B4 323657                    LD     (PAGES),A        ;UPDATE CURRENT PAGE
55B7 FE04                      CP     4                ;TEST FOR ALL DONE
55B9 2816                      JR     Z,EXIT
55BB 21D63F                    LD     HL,VIDEO+982
55BE 222040                    LD     (CURSOR),HL
55C1 217A57                    LD     HL,PRMPT1
55C4 CD3956                    CALL   OUTSTR
55C7 CD4900                    CALL   GETKEY
55CA AF                        XOR    A                ;ZERO LOCAL PAGE TRK CNT
55CB 32D157                    LD     (TRK),A
55CE C3ED54                    JP     DISP             ;DISPLAY NEXT 10 TRACKS

55D1 21D13F           EXIT     LD     HL,VIDEO+977
55D4 222040                    LD     (CURSOR),HL
55D7 21A257                    LD     HL,PRMPT2
55DA CD3956                    CALL   OUTSTR
55DD CD4900                    CALL   GETKEY
55E0 FE40                      CP     '@'              ;CHECK FOR DISPLAY AGAIN
55E2 200A                      JR     NZ,DOSEXT        ;NO - EXIT
55E4 AF                        XOR    A
55E5 323657                    LD     (PAGES),A        ;RESET POINTERS
55E8 32D157                    LD     (TRK),A
55EB C3ED54                    JP     DISP             ;REDISPLAY THE MAP
55EE FB               DOSEXT   EI
55EF C32D40                    JP     DOS

                      ; ROUTINE <FXDE> IS USED BY THE MAP BUILDER TO RESET THE
                      ; IX REGISTER TO THE CORRECT LOCATION IN THE BUFFER WHICH
                      ; WILL CORRESPOND TO THE FXDE LINKED TO THE CURRENT FPDE.
                      ; THE INFORMATION IS EXTRACTED FROM THE SECOND BYTE OF
                      ; THE EXTENT, EG. "FE 44" IS READ AS FOLLOWS:
                      ;        X'FE'=EXTENT POINTS TO FXDE
                      ;        X'44'=010 00 100B
                      ;        THE 1ST 3 BITS IS THE LOCATION OF THE FXDE IN THE
                      ;        DIRECTORY SECTOR IDENTIFIED BY THE LAST 3 BITS.
                      ;        010B=2 ==> THE FXDE IS IN 2*20H=40H ENTRY IN
```

*Listing continues*

See List of Advertisers on Page 451

*Listing continued*

```
                  ;         SECTOR 6 OF THE DIRECTORY TRACK (100B=4, SO
                  ;         THIS IS THE 4TH SECTOR FROM THE "HIT" SECTOR)

55F2 DD7E01       FXDE      LD      A,(IX+1)        ;GET THE BIT RECORD
55F5 E607                   AND     7               ;GET SECTOR NUMBER
55F7 C602                   ADD     A,2             ;OFFSET COUNT
55F9 C660                   ADD     A,BUFFER<-8     ;CORRECT FOR MEM ADDR
55FB 67                     LD      H,A
55FC DD7E01                 LD      A,(IX+1)        ;GET BIT RECORD
55FF E6E0                   AND     0E0H            ;STRIP OFF ENTRY LOC
5601 6F                     LD      L,A
5602 E5                     PUSH    HL              ;SET UP ADDR IN IX
5603 DDE1                   POP     IX
5605 C39754                 JP      SCAN1           ;BACK IN BUSINESS

                  ; SUBROUTINE <STORIT> USES THE TRACK NUMBER IN (A), THE
                  ; GRANULE NUMBER WHICH IS BIT 5 OF THE SECOND BYTE OF
                  ; EXTENT TO INDEX THE MAP TABLE AND STORE THE "FILESPEC"
                  ; IN <FILENM> IN THE NUMBER OF CONTIGUOUS ENTRIES IN MAP
                  ; ACCORDING TO THE LOW ORDER 5 BITS IN THE SECOND BYTE
                  ; OF THE EXTENT.

5608 6F           STORIT    LD      L,A             ;TRACK NR TO HL
5609 2600                   LD      H,0
560B 29                     ADD     HL,HL           ;*2 = GRANULE NUMBER
560C DD7E01                 LD      A,(IX+1)        ;GET SECOND BYTE
560F CB6F                   BIT     5,A             ;TEST FOR GRANULE 0/1
5611 2801                   JR      Z,STOR1
5613 23                     INC     HL              ;COMPUTED GRANULE NUMBER
5614 E61F         STOR1     AND     31              ;GET GRANULE COUNT
5616 3C                     INC     A
5617 47                     LD      B,A             ;SET UP LOOP
5618 CD1E56       STOR2     CALL    MOVEIT          ;MOVE FILESPEC
561B 10FB                   DJNZ    STOR2
561D C9                     RET

                  ; SUBROUTINE <MOVEIT> IS USED TO MOVE THE FILESPEC
                  ; CONTAINED IN <FILENM> TO THE 13 BYTE VECTOR POINTED
                  ; TO BY THE HL REGISTER (WHICH IS TAKEN AS A GRANULE
                  ; NUMBER TO INDEX THE <MAP> ARRAY)

561E E5           MOVEIT    PUSH    HL
561F D1                     POP     DE
5620 E5                     PUSH    HL              ;SAVE POINTER ON STACK
5621 29                     ADD     HL,HL           ;INDEX*2
5622 19                     ADD     HL,DE           ;INDEX*3
5623 29                     ADD     HL,HL           ;INDEX*6
5624 29                     ADD     HL,HL           ;INDEX*12
5625 19                     ADD     HL,DE           ;INDEX*13
5626 114658                 LD      DE,MAP          ;BASE
5629 19                     ADD     HL,DE           ;BASE+INDEX*13
562A 110B57                 LD      DE,FILENM
562D 1A           MOV       LD      A,(DE)          ;MOVE CHARACTER
562E B7                     OR      A
562F 2805                   JR      Z,MOV1          ;EXIT ON 0 TERMINAL
5631 77                     LD      (HL),A
5632 23                     INC     HL
5633 13                     INC     DE
5634 18F7                   JR      MOV
5636 E1           MOV1      POP     HL              ;RESTORE POINTER
5637 23                     INC     HL
5638 C9                     RET

                  ; SUBROUTINE <OUTSTR> COPIES THE INPUT STRING TO THE
                  ; VIDEO DISPLAY UNTIL A TERMINAL CHARACTER OF 0 IS
                  ; DETECTED IN THE INPUT STREAM

5639 7E           OUTSTR    LD      A,(HL)          ;GET NEXT CHARACTER
563A B7                     OR      A
563B C8                     RET     Z               ;RETURN ON END OF STRING
563C CD3300                 CALL    DISPL           ;DISPLAY IT
563F 23                     INC     HL
5640 18F7                   JR      OUTSTR

                  ; SUBROUTINE <FLSPEC> BUILDS FILESPEC IN THE FORM OF
                  ; "FILESPEC/EXT" FROM THE INFORMATION IN THE FPDE IN THE
                  ; DIRECTORY POINTED TO BY THE (HL)
                  ; THE RESULT IS RETURNED IN THE LOCATIONS STARTING AT
                  ; <FILENM> AND TERMINATING WITH A 0 BYTE

5642 E5           FLSPEC    PUSH    HL              ;SAVE FPDE POINTER
5643 010500                 LD      BC,5            ;OFFSET FOR NAME
5646 09                     ADD     HL,BC
5647 110B57                 LD      DE,FILENM
564A 0608                   LD      B,8             ;MAX LENGTH OF NAME
564C 7E           FLS0      LD      A,(HL)          ;MOVE ONE CHARACTER
564D 12                     LD      (DE),A
564E 23                     INC     HL
564F FE20                   CP      ' '             ;TEST FOR EARLY END
5651 2803                   JR      Z,FLS1
5653 13                     INC     DE
5654 10F6                   DJNZ    FLS0            ;LOOP FOR WHOLE NAME
5656 AF           FLS1      XOR     A
5657 12                     LD      (DE),A          ;TERMINATE NAME
5658 E1                     POP     HL
5659 010D00                 LD      BC,5+8          ;OFFSET FOR EXTENSION
565C 09                     ADD     HL,BC
565D 7E                     LD      A,(HL)          ;TEST FOR NOT NULL
565E FE20                   CP      ' '             ;IF SO-RETURN
5660 C8                     RET     Z
```

*Listing continues*

*Listing continued*

```
5661 EB                   EX      DE,HL
5662 362F                 LD      (HL),'/'        ;SET EXTENSION IN FILENM
5664 23                   INC     HL
5665 77                   LD      (HL),A          ;MOVE 1ST CHAR
5666 EB                   EX      DE,HL
5667 13                   INC     DE
5668 23                   INC     HL
5669 7E                   LD      A,(HL)          ;TEST FOR TERM
566A FE20                 CP      ' '
566C 280A                 JR      Z,FLEXIT
566E 12                   LD      (DE),A
566F 13                   INC     DE
5670 23                   INC     HL
5671 7E                   LD      A,(HL)          ;THIRD CHARACTER
5672 FE20                 CP      ' '             ;TEST FOR TERM
5674 2802                 JR      Z,FLEXIT
5676 12                   LD      (DE),A
5677 13                   INC     DE
5678 AF          FLEXIT   XOR     A
5679 12                   LD      (DE),A          ;FORCE TERMINAL
567A C9                   RET


             ; DATA DEFINITION AREA

567B 2A      TITLE    DEFM    '* DISK GRANULE MAP *'
568F 0D00             DEFW    CR
5691 0D      ERR1     DEFB    CR
5692 2A               DEFM    '**** ILLEGAL DRIVE NUMBER!'
56AC 0D00             DEFW    CR
56AE 0D      ERR2     DEFB    CR
56AF 2A               DEFM    '**** DRIVE '
56BA 30      DRNR     DEFM    '0 IS NOT READY!'
                      DEFW    CR
             DKMSG    DEFM    'DRIVE:'
56C9 0D00    DRVNR    DEFM    '0    '
56CB 44      DKNAME   DEFM    'DISKNAME   '
56D1 30      DATE     DEFM    'MM/DD/YY  GRANS FREE:'
56D5 44      GRANS    DEFM    '0    TRKS LOCKED OUT:'
56DF 4D      TLO      DEFM    '0 '
56F4 30               DEFB    0
5708 30      FILENM   DEFM    'FILESPEC/EXT'
570A 00               DEFB    0
570B 46      SELBIT   DEFB    0
5717 00      DIRERR   DEFB    CR
5718 00               DEFM    '**** DIRECTORY READ ERROR!'
5719 0D               DEFW    CR
571A 2A      PAGES    DEFW    0
5734 0D00    PRMPT0   DEFW    0D0DH
5736 0000             DEFM    'PLACE THE DISKETTE TO MAP IN DRIVE 0'
5738 0D0D             DEFB    CR
573A 50               DEFM    'PRESS <ENTER> TO CONTINUE'
575E 0D               DEFW    CR
575F 50      PRMPT1   DEFM    'PRESS ANY KEY FOR NEXT 10 TRACK DISPLAY'
5778 0D00             DEFB    0
577A 50      PRMPT2   DEFM    'PRESS @ TO DISPLAY AGAIN,'
57A1 00               DEFM    'ANY OTHER KEY TO EXIT'
57A2 50               DEFB    0
57BB 41      TRK      DEFW    0
57D0 00      TRKNR    DEFW    0
57D1 0000    LINE3    DEFB    191
57D3 0000             DEFM    '      1ST GRANULE      '
57D5 BF               DEFB    191
57D6 20               DEFM    '      2ND GRANULE      '
57EC BF               DEFW    191
57ED 20      LINEN    DEFM    'TRACK NUMBER: 0   '
5803 BF00             DEFB    191
5805 54               DEFM    ' <------------------> '
5816 BF               DEFB    191
5817 20               DEFM    ' <------------------> '
582D BF               DEFW    191
582E 20
5844 BF00    ;
             ; SYSTEM LABEL DEFINITIONS
             ;

             CMDBUF   EQU     4318H           ;DOS COMMAND BUFFER
             STATUS   EQU     37ECH           ;FDC STATUS REGISTER
4318         CMDREG   EQU     37ECH           ;FDC COMMAND REGISTER
37EC         SECTOR   EQU     37EEH           ;FDC SECTOR REGISTER
37EC         DATARG   EQU     37EFH           ;FDC DATA REGISTER
37EE         SELECT   EQU     37E1H           ;FDC SELECT LINES
37EF         BUFFER   EQU     6000H           ;DIRECTORY BUFFER
37E1         GAT      EQU     BUFFER
6000         FDES     EQU     BUFFER+0200H    ;START OF FDE'S IN BUFFER
6000         VIDEO    EQU     3C00H
6200         DIRTRK   EQU     17              ;DIRECTORY ON TRK 17
3C00         NRTRKS   EQU     40              ;NUMBER OF TRACKS
0011         DOS      EQU     402DH
0028         CURSOR   EQU     4020H
402D         CLS      EQU     01C9H           ;BASIC CLEAR SCREEN
4020         HLACC    EQU     0A9AH           ;ROM ROUTINE HL-->ACCUM
01C9         CVTASC   EQU     0FBDH           ;ROM CVT ACCUM TO STRING
0A9A         GETKEY   EQU     0049H           ;WAIT FOR KEY
0FBD         DISPL    EQU     0033H
0049         DELAY    EQU     0060H
0033         CR       EQU     13
0060
000D
5846         MAP      EQU     $               ;STARTING ADDRESS FOR MAP
5300                  END     DSKMAP
```



✓ See List of Advertisers on Page 451

*Scramble space ships instead of eggs.*

# Astro-Scramble

*Randy Long*
*4791 Torida Way*
*Yorba Linda, CA 92686*

Combining Astro-Blasters and Scramble yields the fast action arcade game, Astro-Scrambler. I'll use an earlier version, a game called Dodge, to demonstrate the evolution of Astro-Scrambler. To better understand Astro-Scrambler, let's walk through Dodge.

### Program Sketch

To start, we need a spaceship:

```
20  CLS: RANDOM: P = 31
30  PRINT @960,""
40  PRINT @P, CHR$(130) CHR$(175)
    CHR$(135);
```

| | | |
|---|---|---|
| HS | — | High score |
| HU | — | Highest set reached |
| A$(1-26) | — | Objects which shoot at you |
| N$ | — | Credit |
| D$ | — | Downward Line Feed |
| U$ | — | Upward Line Feed |
| L$ | — | INKEY$ |
| L | — | Lights new high score |
| P | — | Ship position |
| C | — | Used in PEEKing |
| T | — | Used in loops |
| N | — | Score counter |
| B | — | Number of objects in each set |
| U | — | Number of sets passed |
| X$ | — | Used in instructions |

*Table 1. List of Variables*

```
60  IF PEEK(14400) = 32 THEN
    P = P - 2: IF P<2 THEN P = 3: GOTO
    70 ELSE 70
70  IF PEEK(14400) = 64 THEN
    P = P + 2: IF P>890 THEN P = 890:
    GOTO 80
80  REM used later
100 GOTO 30
```

The ship's location is the variable P (see Table 1). P is increased or decreased (and the ship moved to either side) with the left and right arrow keys. PEEK (14400) is the fastest way to scan the keyboard to check which key is being pressed.

Type this addition into your program:

```
30  PRINT @ 960 + RND(60), CHR$(174)
    CHR$(191) CHR$(157)
```

Since line 30 does not have a semi-colon at the end, the screen moves up one line to print the next asteroid. The program, as it is now, is fast and fun, but not very challenging—you can fly unchallenged through the asteroids.

PEEKing locations 15360–16383 makes the computer look one space ahead of your ship. If it finds something in this space it reasons you have crashed into an asteroid. If nothing is in this space you continue flying.

Now add this line:

```
50  C = PEEK(15360 + P + 65): IF C = 32
    THEN 60 ELSE PRINT @P - 3,
    "BOOM!!";: GOTO 50
```

The variable C is assigned the ASCII code number of the space in front of your ship. If nothing is there, C will equal 32. If it does not equal 32, you have run into an asteroid.

You now have version 1: Dodge. To stay alive you must dodge the objects shooting at you.

In Program Listing 1, I added many new objects and a scoring system creating the game Astro-Scrambler. This game is just like a real action arcade. It is not easy to win!

### Program Modifications

For a longer game change line 280 to:

```
280 N = N + 1 :B = B + 1 :IF B = ### THEN
    B = 0 :U = U + 1 :IF U = 26 THEN 300
```

Make ### high (60) for a longer game or low (15) for a shorter game.

Make your own objects shoot

*Program Listing 1. Astro-Scrambler*

```
10 '  ASTRO-SCRAMBLER BY RANDY LONG
20 '                    4791 TORIDA WAY
30 '                    YORBA LINDA, CAL. 92686
40 '                    (714) - 528 - 1840
50 CLEAR 1100 :HS=6280 :HU=26 :DIM A$ (26)
60 N$ = CHR$(198) + "80-MICROCOMPUTING"
70 D$ = CHR$(26) :U$ = CHR$(27) :GOSUB 310
80 PRINT@ 960,"PUSH ENTER TO START/P=POINT";:GOSUB 430 :CLS :IFL$="P"THEN L
=0 :PRINTTAB(6) CHR$(23)"POINT CHART":PRINT:PRINT:PRINT"EACH CRASH  - 200 P
OINTS":PRINT:PRINT:PRINT:PRINT"TOTAL NUMBER OF SHIPS":PRINT"AT END OF TRIP
 + 600":ELSE 100
90 L$ = "" :PRINT :PRINT"EACH SET PASSED      + 100 EACH" :PRINT:PRINT"EACH
OBJECT PASSED  +10 POINTS" :GOTO 80
100 FOR T = 1 TO 25 :A$(T) = "" :NEXT T
110 L = 0 :A$(1) =CHR$(166)+CHR$(153) :A$(2) =CHR$(174)+CHR$(191)+CHR$(157)
 :A$(3) =STRING$(2,27)+CHR$(176)+CHR$(26)+STRING$(2,8)+CHR$(166)+" "+CHR$(1
53)+CHR$(26)+STRING$(2,8)+CHR$(131)
120 A$(4) =U$+CHR$(152)+CHR$(167)+CHR$(151)+CHR$(164)+D$+STRING$(4,8)+CHR$(
137)+CHR$(185)+CHR$(183)+CHR$(158) :A$(5) =U$+CHR$(160)+CHR$(184)+CHR$(191)
+CHR$(191)+CHR$(188)+CHR$(176)+D$+STRING$(6,8)+CHR$(139)+CHR$(191)+CHR$(191
)+STRING$(2,143)+CHR$(135)
130 A$(7) =A$(6) :A$(8) =U$+U$+CHR$(160)+CHR$(140)+CHR$(131)+CHR$(141)+CHR$
(140)+CHR$(144)+D$+STRING$(6,8)+CHR$(165)+CHR$(144)+CHR$(134)+CHR$(153)+CHR
$(137)+CHR$(162)+CHR$(132)+D$+STRING$(6,8)+CHR$(138)+CHR$(140)+CHR$(164)+CH
R$(152)+CHR$(129)
140 A$(9) =CHR$(184)+CHR$(189)+CHR$(144)+CHR$(237):A$(10)=CHR$(130)+CHR$(17
5)+CHR$(135)
```

*Listing continues*

### *The Key Box*

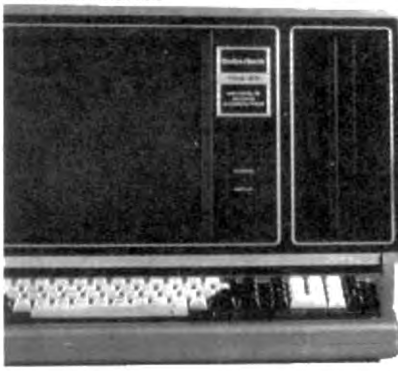
**Basic Level II**
**Model I**
**4K RAM**

# PRIORITY ONE ELECTRONICS

# 3 DRIVE MODEL II MASS STORAGE FROM INTERNATIONAL INSTRUMENTATION INC.

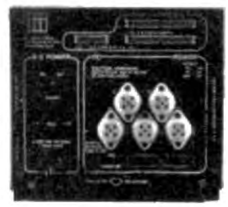
## AT LAST! An Alternative to Radio Shack's Over-Priced, Over-Sized Model II Disk Expansion!

The International Instrumentation THIN THREE™ disk drive enclosure is the perfect alternative to Radio Shack's expensive, bulky disk expansion box for the Model II. The THIN THREE™ is designed to hold three half size Tandon TM848-1 8" disk drives. These are the same drives Radio Shack is using in the new Model 16™. The THIN THREE™ occupies less than half of the space required for Radio Shack's disk box, leaving you more room to work. Best of all the THIN THREE™ with drives and cables installed, is priced at only **$1795.00. THAT'S $555.00 LESS THAN THE RADIO SHACK DISK BOX!**



### FEATURES

- Connects directly to the disk expansion port on the back of your Model II; no hardware modifications are required; completely Model II compatible
- Filtered positive pressure forced air cooling for dust free, reliable disk drive operation
- Internal data cable for easy hook-up
- AC input via 3 wire 7 foot international cord/socket set
- AC input EMI filtered to six amps to help prevent disk crashes due to power spikes and line noise
- Slide out chassis with front panel pulls for easy access to drives
- Disk Environment Monitor Panel monitors supply voltages and displays internal case temperature on standard DVM. Alarm sounds if case temperature exceeds safe limits, protects your disks
- ¼" aluminum rear panel with socketed power pass devices for cool trouble free power



- 14 gauge main chassis
- Easy externally replaceable fan filter
- Integral power supply with 5V@6A/-5V@1A/24V@6A
- Double-sided custom PC power board and supply engineered especially for the THIN THREE™
- Each DC supply and AC separately fused
- Overvoltage protected 5V supply
- LED power supply monitors and test jacks for each supply on rear panel

**8LPDBIIITRS** *A & T with external data cable and* **$1795.00**
IIIM2PATCH TRSDOS ENHANCEMENT PACKAGE
(Shipped Motor Freight Collect)

### III TRSDOS™ UTILITY ENHANCEMENTS

This package of 12 individually selectable utility programs is designed for the sophisticated Model II TRSDOS™ user. Each utility is designed to increase system efficiency by overcoming the limitations of TRSDOS™. Here are three examples:

- Modifies TRSDOS™ 2.0 or 2.0A to run with DC motor disk drives such as the Tandon TM848-1
- Elimination of password protection on TRSDOS™ files to allow total user manipulation of these files
- Changes TRSDOS™ step rate from 20ms to 12ms, user may adjust to as low as 6ms

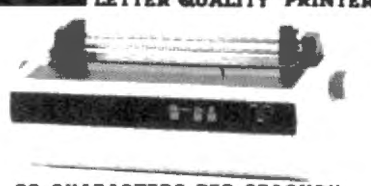
**8LIIIM2PATCH** Supplied on 8" disk **$99.95**

TRS80 Model II, TRS80 Model 16, TRSDOS and Radio Shack are trademarks of Tandy Corp. THIN THREE is a trademark of International Instrumentation Inc.

# MORE PERIPHERALS FOR THE TRS-80 MODEL II*

## *Also For Your TRS-80 Models I and III

## FUJITSU FROM SELLUM LETTER QUALITY PRINTERS



### 80 CHARACTERS PER SECOND!!

**That's over three times as fast as Radio Shack's Daisy Wheel II**

The Fujitsu SP83 letter quality printer from Sellum is not only the fastest daisy wheel letter quality printer on the market, it is also the most versatile. It may be hooked up to either the parallel or one of the serial ports on your Model II*. It has five different printer protocols built in so it will run with just about any word processing package on the market including SCRIPSIT*. The printer is built to be solid and quiet, it won't shake the office down with every carriage return. It also has a 16K buffer (48K optional). The buffer allows your computer to send data out to the printer and not have to wait for it before allowing you to continue. No more staring at the screen waiting for the printer to finish!!

**MORE SPECIAL FEATURES:** • Z80 CPU • 50-19.2K baud on serial interface • Distinct alarm sound for each error condition • Intelligent bi-directional logic seeking • Complete word processing features including sub and super script, underlining, and double-strike characters • Auto reprint • Auto Clear • Proportional spacings and supports automatic justification • Front panel forms control • Compatible with most mechanical sheet feeders • Quiet-60db • Six month warrantee

| Part No. | Description | List Price | SALE |
|---|---|---|---|
| 8LPDBSLMF86 | Printer w/16K buffer | $3495.00 | $2795.00 |
| 8LPDBSLMF86B48 | Printer w/48K buffer | $3895.00 | $3095.00 |
| 8LSLMF86VFT | Optional vertical forms tractor | | $ 200.00 |
| 8LSLMPS34002 | Model II Parallel Cable | | $ 80.00 |
| 8LSLMPE34003 | Model I/III Parallel Cable | | $ 80.00 |

*(Printer shipped via motor freight collect)*

### 8" CP/M® SOFTWARE FOR MODEL II

#### MICROPRO INTERNATIONAL

| | | |
|---|---|---|
| 8LMDSWDSTR | Wordstar Version* 3.0 | $249.00 |
| 8LMDSMLMRG | Mail Merge* | $100.00 |
| 8LMDSSPSTR | Spell Star* | $150.00 |
| 8LMDSCALSTR | Calc Star* | $200.00 |
| 8LMDSDATSTR | Data Star* | $200.00 |
| 8LMDSSUPSET | Supersort* | $150.00 |

**Trademark of Micropro International*

#### SORCIM

| | | |
|---|---|---|
| 8LSORCAL | SUPERCALC* a Superior Electronic Worksheet | $229.00 |

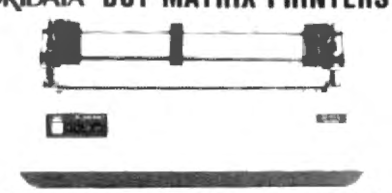
**Trademark of Sorcim*

#### MICROSOFT

| | | |
|---|---|---|
| 8LMDSBAS80 | BASIC-80* Version 5.X Extended Disk Basic | $300.00 |
| 8LMCPBASC | BASIC* Compiler Version 5.X | $395.00 |
| 8LMDSMFT | FORTRAN-80* | $500.00 |

**Trademark of Microsoft Consumer Products*

## OKIDATA DOT MATRIX PRINTERS



### COMPARE OKIDATA'S FEATURES TO ANY OTHER PRINTER! COMPARE PRIORITY ONE'S PRICES TO ANY — THEN DECIDE!!

### DON'T BE FOOLED BY THOSE WHO TRY TO "SOAK" YOU FOR THE TRACTOR

- BI-DIRECTIONAL 120 CPS
- 9 x 9 Matrix (Alphanumeric)
- 6 x 9 or 12 Matrix for graphics
- 10 different character sets
- 5, 8.3, 10 or 16 Characters per inch
- 80 CPL @ 10 CPI for 82A
- 132 CPL @ 10 CPI for 83A & 84
- 82A & 83A have both Centronics Parallel & 1200 baud Serial I/O
- Self Test

All Okidata printers feature 10 front panel switch selectable from lengths from 3" to 14". Front or bottom paper loading of up to four part forms, tear bar, and out of paper switch. Most important of all OKIDATA printers are built to survive the business or industrial environment where you'll use them. All of these features make OKIDATA the best value in low cost printers!

| PART NO. | DESCRIPTION | SALE |
|---|---|---|
| 8LPDBOKI82ATG | OKIDATA 82A w/tractor & graphics | $ 539.00 |
| 8LPDBOKI83ATG | OKIDATA 83A w/tractor & graphics | $ 750.00 |
| 8LOKISER2KBF | 9600 Baud 2K Serial buffer/interface | $ 149.00 |
| 8LOKIDAT84AP | 132 col printer parallel interface with graphics and tractor | $1095.00 |
| 8LOKIDAT84AS | 132 col printer 9600 buad serial 2K interface graphics and tractor | $1250.00 |
| 8LPRIAMPSKT | 5' Model II Parallel Cable | $ 19.95 |
| 8LPRIAMPTR80 | 5' Model I/III Parallel Cable | $ 19.95 |
| 8LCNBOKISERBF | 8' OKIDATA Serial Interface Cable | $ 24.95 |

*(Shipping Weights: 82A 25 lbs. 83A & 84 35 lbs.)*

**SEND $1.00 TODAY FOR THE NEW, FULL COLOR SPRING 1982 ENGINEERING SELECTION GUIDE!**

## MODEMS

NEW!

### Hayes SMARTMODEM

The HAYES Smartmodem will alow you to take advantage of the many database services available for your TRS-80 such as Dow Jones, the Source and others. The Smartmodem runs directly off an RS232 serial port.

**FEATURES:** • Auto Answer, Auto Dial • Variable data formats • 0-300 baud • Internal Z8 Microprocessor with 2Kb control program • 40 character command buffer • Easy to program with simple commands from your keyboard

| PART NO. | DESCRIPTION | LIST PRICE | SALE |
|---|---|---|---|
| 8LDCH0200P | Smartmodem (Sh. Wt. 6 lbs.) | $279.00 | $225.00 |
| 8LDCH0400P | Smartmodem 1200 baud | $699.00 | $649.00 |
| 8LDCH0300P | Chronograph (Sh. Wt. 3 lbs.) | $249.00 | $205.00 |
| 8LCNDRS232BF | 8' RS232 Class 1 Cable | | $ 19.95 |

### THE STAR by PRENTICE

- 0 to 300 buad • Bell 103 and 113
- Exclusive triple seal acoustic chamber
- Self Test • Runs off RS232 Port

| | | | |
|---|---|---|---|
| 8LPRNSTAR | RS232 TTL 20Ma Current Loop | $199.00 | $127.00 |
| 8LCNDRS232BF | 8' RS232 Class 1 Cable | | $19.95 |

### 5" DISKETTES!

**SOFT SECTOR
40 TRACK SINGLE SIDED
DOUBLE DENSITY WITH
HUB REINFORCING RINGS**

*PACKAGE OF 10* **$19.95**

### BONUS!

**FREE!! KASSETTE 10 LIBRARY CASE WITH PACKAGE OF 10 DISKETTES**
**A $4.25 VALUE!!** **8LPRI5SDD** *(Sh. Wt. 2 lbs.)*

**8LPRI5DD** *package of 80 less Library Case* **$120.00**

-477


PRIORITY ONE ELECTRONICS

VISA · master charge

9161 DEERING AVE • CHATSWORTH, CA 91311

NEDA · MEMBER MITA

**ORDER TOLL FREE (800) 423-5922 CA, AK, HI CALL (213) 709-5464**


Terms: U.S. VISA, MC, BAC, Check, Money Order, U.S. Funds Only. CA residents add 6½% Sales Tax. MINIMUM PREPAID ORDER $15.00. Include MINIMUM SHIPPING & HANDLING of $3.00 for the first 3 lbs. plus 40¢ for each additional pound. Orders over 50 lbs. sent freight collect. Just in case, please include your phone number. Prices subject to change without notice. We will do our best to maintain prices through December 15, 1982. Credit Card orders will be charged appropriate freight. If you haven't received your Spring 82 Engineering Selection Guide, send $1.00 for your copy today! Sale prices are for prepaid orders only.

at you by putting your string in at line 210. Change what is in the quotes to your new object (for example, 210 A$ (21) = "XX + XX":).

Change your ship's looks in line 240. Try this:

```
240 PRINT@ P, CHR$(133) CHR$(191)
    CHR$(138);
```

For a more colorful game change line 230 to:

```
230 PRINT@ 959 + RND(60), A$(RND(26))
```

To make your ship move faster from side to side change line 260 and 270:

```
260 IF PEEK(14400) = 32 THEN P = P - 4:
    IF P<3 THEN P = 3
270 IF PEEK(14400) = 64 THEN P = P + 4:
    IF P>890 THEN P = 890
```

Enjoy, good luck and keep trying! ■

*Randy Long, an 18 year-old Senior at Troy High School in Fullerton, CA, plans to enroll at California State Polytechnic University to study Computer Sciences.*

*Program continued*

```
150 A$(11)=U$+CHR$(184)+CHR$(189)+CHR$(144)+D$+STRING$(3,8)+CHR$(175)+CHR$(
175)+CHR$(133) :A$(12) =U$+U$+U$+CHR$(149)+D$+CHR$(8)+CHR$(149)+D$+CHR$(8)+
CHR$(149)+D$+CHR$(8)+CHR$(149) :A$(13) =U$+CHR$(136)+CHR$(179)+CHR$(132)+D$
+STRING$(3,8)+CHR$(176)+CHR$(191)
160 A$(13) =A$(13)+CHR$(176)
170 FOR T = 1 TO 2 :A$(14) =A$(14)+CHR$(168)+CHR$(189)+CHR$(200) :NEXT T
180 A$(15) =A$(12)+CHR$(201)+A$(12) :A$(16)=CHR$(149) :A$(17)=CHR$(191)+CHR
$(230)
190 A$(18) =U$+U$+U$ :FOR T = 1 TO 3 :A$(18) =A$(18)+STRING$(5,191)+D$+STRI
NG$(5,8) :NEXT T
200 A$(19) =U$+U$+CHR$(160)+CHR$(180)+D$+STRING$(2,8)+CHR$(176)+CHR$(181)+C
HR$(144)+D$+STRING$(4,8)+CHR$(151)+STRING$(2,143)+CHR$(135)+CHR$(149) :A$(2
0) =CHR$(131)+CHR$(176)+CHR$(140)+CHR$(240)
210 A$(21) ="O" :A$(22) ="A    A" :A$(23) =STRING$(3,91)+CHR$(212)+STRING$(
3,91) :A$(24) ="*-*-*-*" :A$(25)="<+()+>"
220 CLS :RANDOM :P=31
230 PRINT@ 959+RND(60), A$(U)
240 PRINT@ P,CHR$(130) CHR$(175) CHR$(135);
250 C = PEEK( 15360+P+65 ):IF C = 32 THEN 260 ELSE FOR T = 1 TO 20 :PRINT@
P-1,"BOOM!"; :PRINT@ P-2,STRING$(7,128+RND(63)); :PRINT@ P+63,STRING$(5,128
+RND(63)); :PRINT@ P+60+RND(3),CHR$(128+RND(63)); :PRINT@ P+66+RND(3),CHR$(
128+RND(63)); :NEXT T :GOSUB 400
260 IF PEEK( 14400 )= 32 THEN P=P-2 :IF P < 3 THEN P = 3 :GOTO 270
270 IF PEEK( 14400 )= 64 THEN P=P+2 :IF P > 890 THEN P = 890 :GOTO 280 ELSE
 280
280 N = N + 1 :B = B + 1 :IF B = 35 THEN B = 0 :U = U + 1 :IF U = 26 THEN 3
00
290 GOTO 230
300 PRINT@ 960,; :FOR T = 1 TO 190 :PRINT TAB(13)"YOU MADE IT THROUGH THE S
CRAMBLER!!!" :NEXT T :GOSUB 60
310 CLS :PRINT CHR$(23);
320 PRINT@ 64,"A S T R O -" :PRINT TAB(4)"S C R A M B L E R" :PRINT
330 PRINT :PRINT :PRINT :PRINT N$ :PRINT"  USING THE RIGHT AND LEFT" :PRINT
"ARROW KEYS YOU MUST MANUVER" :PRINT"YOUR SHIP THROUGH THE SCRAMBLER" :PRIN
T"  THE IDEA IS TO GO AS FAR AS"
340 PRINT"POSSILBLE INTO THE SCRAMBLER" :PRINT"YOU EARN EXTRA SHIPS AND" :P
RINT" POINTS BY CONTINUALLY MOVING" :PRINT"YOUR SHIP TO THE RIGHT."
350 N = N + ( P*10 ) :N = N + ( 100*U )
360 IF N > HS THEN HS = N :HU = U :L = 1
370 IF U > HU THEN HU = U :HS = N :L = 1
380 PRINT@ 192,"LAST TRIP  "USING"#####"; N ;:PRINT"   SET  "U:PRINT"HIGH S
CORE "HS;:PRINT"  SET  "HU; :U = 0 :N = 0 :IF L = 1 THEN PRINT@ 328,"**** N
EW HIGH SCORE ****";
390 RETURN
400 '
410 IF P < 64 THEN PRINT@ 896,"SORRY, YOU ARE OUT OF SHIPS!" :PRINT TAB(20)
"PLAY AGAIN!" :FOR T = 1 TO 2000 :NEXT T :GOTO 60 ELSE 420
420 P = P -64 :N = N - 200 :U = U - 1 :CLS :B = 0 :RETURN
430 '
440 X$ = STRING$(31,129+RND(62)) :PRINT@ 128,X$;:PRINT@ 320,X$;
450 IF L = 1 PRINT@ 328,"**** NEW HIGH SCORE ****";
460 L$ = INKEY$ :IF L$ = "" THEN 440 ELSE IF L$ = CHR$(13) THEN RETURN ELSE
 IF L$ = "P" THEN RETURN ELSE 440
```

# We've grown on you

## Because Our Programs Deliver What They Promise.

**With our sophisticated programs, you're never disappointed. *And*, our customer service is unparalleled in excellence.**

# With Acorn Programs, Your TRS–80 Does Things You Wouldn't Think Possible

## Finefont

**New!**

By Vincent Bly from Acorn

Now Epson owners can have ***true letter-quality*** printing with FINEFONT at 10, 11, or 12 characters per inch:

| Finefont and |
|---|
| standard Epson |

With boldface, underlining and incremental spacing (for proportional right justification), 32 special characters including those needed for mathematics, plus a graphic editor to ***create*** or edit character sets. It supports all Epson print modes, and is compatible with word processors including Model I SCRIPSIT with SUPERSCRIPT, Model III SCRIPSIT, and Lazy Writer (Model I, III). Print speed is approx. 20 CPS, unidirectionally. It's compatible with most DOS's including LDOS 5.1, TRSDOS, NEWDOS/80 Ver 2.0. **Requires following Epson printers:** MX-80; MX-80 FT with Graftrax-80 or Graftrax-plus; MX-100 ***with*** Graftrax-plus ***only***

**Requires 48K Disk, $49.95***

## SUPERSCRIPT

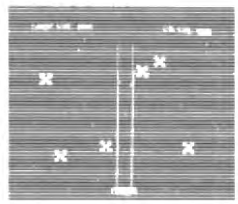
By Richard Wilkes from Acorn

Using your SuperScript modified Scripsit Word Processor and a compatible printer, you can now underline, boldface, insert text during printout, slash zeros, set type pitch, subscript and, of course, superscript! You can even read your directory and kill files without ever leaving Scripsit.

SuperScript comes with drivers for popular serial and parallel printers (now including Centronics 737 and RS Daisy II), and easy instructions for patching to your Scripsit program (does not include Scripsit).

**32K Disk, $49.95** Use Model I Scripsit *

***plus $2.00 shipping and handling**



## Evade

By Tim Knight

You're in charge of the USS Krestan, a tri-ship on interstellar exploration. You must defend yourself against groups of enemy aliens, each requiring a different strategy. Then you can go for the big points by docking with another ship. Be careful, or your ship will be vaporized! **Now** you're ready for the evil Mallicians . . and the meteor showers. Not for the faint-hearted!

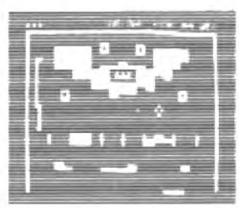
**16K Tape or 32K Disk: TRS-80 I & III.**
**$19.95 plus $2.00 shipping and handling**

## Demise

By P. Case

Shoot down the enemy ships which are coming—and firing—at you. After obliterating them all, you must carefully navigate a tunnel before you are in the clear. You can move your ship left, right, up, down and diagonally; your weapons include a laser cannon and "smart bombs" that render ***all*** ships on the screen impotent. You start with 5 ships and get another ship and "smart bomb" for each 10,000 points. 3 levels of play; great sound effects.

**16K Tape or Disk: TRS-80 I & III.**
**$19.95 plus $2.00 shipping and handling**

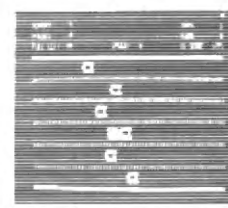


## King of the Jungle

By Patrik Vandewalle

An unusual combination of fast pinball action plus the challenges and dangers of an arcade game. Score points as you maneuver your ball from obstacle to obstacle, while moving through the jungle searching for the hidden crown. You start with 5 balls and get another for each 10,000 points. Choice of 3 screens. Joystick compatible.

**16K Tape or Disk: TRS-80 I & III.**
**$19.95 plus $2.00 shipping and handling**



## Enemy Evasion

By Raul Rivera

An original fast-paced arcade style game requiring quick mental and physical reflexes. You must control your Rover to avoid oncoming Rovelan Ramcraft. You gain points for each successful pass through the "death lanes", but the Ramcraft speed increases as you play. If you reach the double bonus mode, watch out for the land pellets they drop! Joystick compatible.

**32K Disk: TRS-80 I & III.**
**$19.95 plus $2.00 shipping and handling**

**Acorn** Software Products, Inc.
634 North Carolina Avenue, S.E. Washington, D.C. 20003

✓79

## We've Grown On You.

To Order By Phone CALL: **(202) 544-4259**

Dealer Inquiries ***Only***, Call Computer Software Distributors: **(800) 424-3708**

Please send these Acorn Programs: Indicate Tape or Disk ☐ **Please send your catalog**

Charge to: **TOTAL: $**

☐ VISA ☐ MASTERCARD ☐ CHECK ENCLOSED

CARD NO. EXP. DATE

T-12

Name

Address

City State Zip

*Checking the odds.*

# Craps Unrolled

*Bruce Chaffee*
*6571 Edgemont*
*Huntington Beach, CA 92647*

A chance remark by a Las Vegas devotee about losing his shirt at craps led to an interesting experiment which demonstrates the power and versatility of the TRS-80.

The premise is simple: To find the probability of winning at craps, simulate the game on the computer and play a huge number of games. The probability of winning should then be close to the ratio wins over wins plus losses. If this ratio is greater than .5 you may expect to win more often than lose, and craps is a good bet. Is it really that easy?

Here are two such programs, using the rules shown in Fig. 1. Program Listing 1 plays the game 10,000 times and Program Listing 2 allows you to choose the number of games the computer will play. If you load and run the program in Listing 2 (set the number of games at about 10,000), you will probably observe that the first few digits soon settle down to .49XXXX. Sounds great! All you need do is let the computer run until more and more digits stay fixed.

| Trial | Number of Games | Probability of Winning |
|---|---|---|
| 1 | 10,000 | .4969 |
| 2 | 10,000 | .4928 |
| 3 | 10,000 | .4923 |
| 4 | 10,000 | .4972 |
| 5 | 10,000 | .5003 |
| 6 | 10,000 | .4958 |
| 7 | 10,000 | .4882 |
| 8 | 10,000 | .4956 |
| 9 | 10,000 | .4802 |
| 10 | 10,000 | .4911 |

*Table 1*

### Problems

Assuming that the roll of real dice is random, we are still left with two problems with our simulation. First, how certain are we of an answer close to the real answer? Second, how random are the "pseudo-random" numbers the computer generates?

To answer the first of these questions, play 10,000 games 10 times, and use the standard deviation of the 10 individual results to estimate the accuracy. We will approach the second question more indirectly. If we calculate the actual result from principles of probability, we can then compare the theoretical result to the experimental result. The agreement, or lack of it, will be a test of the random number generator.

### Random Numbers

I have provided an outline of the calculation of the theoretical answer. You may want to skip the calculation and simply note the result at the end of this paragraph.

P(win) = P(win first roll) + P(win on subsequent rolls)
P(win first roll) = P(7 first roll) + P(11 first roll) = 6/36 + 2/36

Calculating P(win first roll) is easy. Calculating P(win on subsequent rolls) is a bit more complicated because winning can occur if you roll a 4, 5, 6, 8, 9 or 10 on the first roll and then match your point before you roll a 7. You can calculate these probabilities separately and then add them. For example:

P(win with point = 4) = P(4 on first roll) [P(4 on second roll) + P(no win on second roll)* P(4 on third roll) + . . .]

$$P(\text{win with point} = 4) = 3/36(3/36 + 3/36(1 - 9/36) + 3/36(1 - 9/36)^2 + \ldots)$$

Notice the right side is a geometric series which you can add. It equals 1/3. The other cases are similar, yielding a theoretical result of $.492\overline{929}$, a repeating decimal.

Program Listing 3 is a program to analyze the data.

I put the program in Listing 1 in a loop which executed the Random command after each 10,000 games, and found the ten results shown in Table 1. The results can be summarized by say-

***The Key Box***
**Basic Level II**
**Model I or III**
**16K RAM**
**Printer required**

ing the overall average is .49304 and the standard deviation is .00569351. That is, with:

50 percent certainty $.492659 < p < .493422$
68.3 percent certainty $.492471 < p < .493809$
95.6 percent certainty $.491901 < p < .494179$
99.7 percent certainty $.491332 < p < .494748$

From before, the true answer is $.4929\overline{29}$, so the experimental answer is off by $[.49304 - .4929\overline{29}] \cong .000111$, which is much less than one standard deviation.

The method seems to work. We did something more than approximate an answer we already knew. The method would not have worked if the pseudo-ran-

| | First Roll | Subsequent Rolls |
|---|---|---|
| Win | 7,11 | Match first roll, or "point" |
| Lose | 2, 3 or 12 | 7 |

*Figure 1*

```
5 DEFINT I,S,J,N
6 FOR NU=1TO10
7 I=0:S=0:J=0:W=0:L=0
10 CLS:PRINT@32,CHR$(32);:PRINT"10,000 GAMES":N=10000
14 '
15 '            ***** SET UP HEADINGS *****
16 '
30 PRINT@128,"FIRST ROLL"
40 PRINT@256,"SUBSEQUENT ROLLS"
50 PRINT@512,"WINS"
60 PRINT@527,"LOSSES"
70 PRINT@542,"TOTAL"
80 PRINT@704,"'PROBABILITY'"
84 '
85 '            ***** MAIN LOOP *****
86 '
90 FOR I=1 TO N
100 S1=RND(6)+RND(6)
120 IF S1= 2 OR S1=3 OR S1=12 THEN L=L+1: GOTO 220  'TRY OTHER RUL
ES HERE
130 IF S1=7 OR S1=11 THEN W=W+1: GOTO 220
140 J=0
150 S2=RND(6)+RND(6)
170 J=J+1
180 IF S2=S1 THEN W=W+1: GOTO 220
190 IF S2=7 THEN L=L+1: GOTO 220
200 GOTO 150
205 '
206 '            ***** PRINT RESULTS *****
207 '
220 PRINT@606,(W+L)
230 PRINT@770,CHR$(30);W/(W+L)
240 NEXTI
300 LPRINT"RANDOM EVERY 10,000 W =";W;"L =";L;"   P = ";W/(W+L)
310 RANDOM
320 NEXT NU
```

*Program Listing 1*

```
5 DEFINT S,J: W=0:L=0
10 CLS:PRINT@32,CHR$(32);:INPUT"NUMBER OF GAMES";N
14 '
15 '            ***** SET UP HEADINGS *****
16 '
30 PRINT@128,"FIRST ROLL"
40 PRINT@256,"SUBSEQUENT ROLLS"
50 PRINT@512,"WINS"
60 PRINT@527,"LOSSES"
70 PRINT@542,"TOTAL"
80 PRINT@704,"'PROBABILITY'"
84 '
85 '            ***** MAIN LOOP *****
86 '
90 FOR I=1 TO N
95 IF INT(N/1000)*1000=N THEN RANDOM
100 S1=RND(6)+RND(6)
110 PRINT@140,S1
120 IF S1= 2 OR S1=3 OR S1=12 THEN L=L+1: GOTO 210  'TRY OTHER RUL
ES HERE
130 IF S1=7 OR S1=11 THEN W=W+1: GOTO 210
140 J=0
150 S2=RND(6)+RND(6)
160 PRINT@320+J*3,S2
170 J=J+1
180 IF S2=S1 THEN W=W+1: GOTO 210
190 IF S2=7 THEN L=L+1: GOTO 210
200 GOTO 150
205 '
206 '            ***** PRINT RESULTS *****
207 '
210 PRINT@320,CHR$(30):PRINT@384,CHR$(30)
220 PRINT@576,W:PRINT@591,L:PRINT@606,(W+L)
230 PRINT@770,CHR$(30);W/(W+L)
240 NEXTI
```

*Program Listing 2*



✓See List of Advertisers on Page 451

dom numbers were not close to random. The results suggest that the pseudo-random numbers were suitable for this technique and next time you can use them with more confidence.

### Final Thoughts

It is very difficult to control the seed number of the random number generator. This makes it difficult to duplicate an experiment.

Are the results presented here too good? The overall average is surprisingly close to the true value. In other experiments the overall average was not so close. I selected the data presented here because they are the first 10 results produced after I turn on the machine, and in this sense they can be duplicated.

If you ran the program with 10,000 games, then you might think that 20,000 games would give an answer that is twice as accurate. Such is not the case. In general, the expected error varies as $1/\sqrt{n}$. In other words, to get one more significant digit (decrease the error by a factor of 10) you must increase the number of trials by a factor of 100 (to 1,000,000). Often, other methods of increasing accuracy are more productive.

If you change the rules of the game, what happens to the probability of winning? The first time I played craps we used the rule that you did not lose if you got a three on the first roll. That is, three was just like any other point. Does this change things much? Change line 120 by deleting OR S1 = 3, and see what the simulation tells you. (The theoretical value is $.50\overline{681}$.)

The Monte Carlo technique, of which this is a crude example, can be used on very difficult problems to give an approximate answer. The method does, however, lead fairly quickly to some unavoidable mathematics. I have tried to minimize the math used here in order not to obscure the central ideas. You can refer to any standard text on statistics for a more detailed explanation.■

*Mr. Chaffee received a master's degree in mathematics. He works as a consultant in microcomputer applications.*

```
10 '     ***** PROGRAM TO ANALYZE DATA *****
15 CLS
20 INPUT" HOW MANY NUMBERS";N
30 DIM N(N)
40 FOR I=1 TO N
50 INPUT"ENTER NUMBER"; N(I)
60 NEXT I
70 '     ***** FIND SUM *****
80 SUM = 0
90 FOR I=1 TO N
100 SUM = SUM + N(I)
110 NEXT I
120 '  ***** AVERAGE *****
130 AVG = SUM/N
140 ' ***** SIGMA *****
150 DEV = 0
160 FOR I=1 TO N
170 DEV = DEV +(AVG-N(I))[2
180 NEXT I
190 SIG = SQR(DEV/(N-1))
200 '    ***** RESULTS *****
210 FOR I = 1 TO N
220 LPRINT N(I);
230 NEXT I
240 LPRINT
245 LPRINT
250 LPRINT " AVG = "; AVG
260 LPRINT "STANDARD DEVIATION =";SIG
270 LPRINT
275 LPRINT"50%  CERTAINTY   ";SUM-.67*SIG;" <= SUM <= ";SUM+.67*SI
G
280 LPRINT"68.3% CERTAINTY   ";SUM-SIG;" <= SUM <= ";SUM+SIG
285 LPRINT"95.6% CERTAINTY  ";SUM-2*SIG;" <= SUM <= ";SUM+2*SIG
290 LPRINT"99.7% CERTAINTY   ";SUM-3*SIG;" <= SUM <= ";SUM+3*SIG
300 LPRINT
310 LPRINT
320 LPRINT"ANALYSIS OF AVERAGE VALUE"
325 LPRINT"50%  CERTAINTY    ";(SUM-.67*SIG)/N;" <= 'P' <= ";(SUM+
.67*SIG)/N
330 LPRINT"68.3% CERTAINTY   ";(SUM-SIG)/N;" <= 'P' <= ";(SUM+SIG)
/N
335 LPRINT"95.6% CERTAINTY   ";(SUM-2*SIG)/N;" <= 'P' <= ";(SUM+2*
SIG)/N
340 LPRINT"99.7% CERTAINTY   ";(SUM-3*SIG)/N;" <= 'P' <= ";(SUM+3*
SIG)/N
350 LPRINT
360 LPRINT"TRUE VALUE = .492929"
370 LPRINT"THIS DIFFERS FROM THE AVERAGE BY";ABS(.492929-AVG)
380 LPRINT"THIS REPRESENTS ";ABS(.492929-AVG)/SIG;" STANDARD DEVIA
TIONS"
```

*Program Listing 3*

# IT'S MORE FUN THAN A THREE-RING CIRCUS

A WAYNE GREEN PUBLICATION

# COMPUTER CARNIVAL

**EDUCATIONAL FEATURES**

EXPANDS CREATIVITY
DEVELOPS KEYBOARD SKILLS
GEOGRAPHICAL QUIZZES
MATH AND WORD GAMES

YOUR CHILD WILL BE CAPTIVATED for hours with your gift of the 60 programs in **Computer Carnival**. Tests of skill, haunted houses, mazes, wizards and dragons are all a part of this funhouse book. Written in TRS-80* Level II Basic. Author Richard Ramella creates many characters, from the mysterious Sugar Louie to the mystical Madam Zelda, as he takes your child through his arcade of games, graphics and quizzes.

**Carnival Companion**™ is a 30-minute cassette of the longer listings which eliminates the aggravation of typing and searching for errors, leaving more time to spend on the midway.

LARGE READABLE TYPE
SPIRAL BINDING
SHORT, EASY-TO-TYPE PROGRAMS
READY-TO-LOAD CASSETTE OF THE LONGER LISTINGS
AND
**MORE**

## ORDER INFO.

Toll Free **1-800-258-5473** for your credit card order. Or mail your order with payment or complete credit card information plus $2 shipping and handling to:
Wayne Green Inc.
Attn: Book Sales
Peterborough, NH 03458
Dealer Inquiries Invited

All orders shipped UPS if complete street address is provided.

*TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

**Package price: $24.97**
**Book only: $16.97**

## COUPON

☐ **COMPUTER CARNIVAL** and **CARNIVAL COMPANION** CC7389 $24.97
☐ **COMPUTER CARNIVAL** (book only) BK7389 $16.97
(Include $2.00 per order for shipping and handling charges.)
☐ Payment enclosed ☐ MC ☐ VISA ☐ AmEx
Card # ________ Expires ________ MC Bank # ________
Signature ________
Name ________
Address & City ________
State ________ Zip ________
CC12

Photocopied coupon is acceptable for ordering.


# WE'RE WRITING

# THEM AS FAST AS WE CAN!

At the rate we're going, we'll have these pages filled by 2083. And by 2084, people will be clamoring for the next Infocom creation.

We hate to disappoint our public. So we keep you waiting. Because while the software factories are cranking out arcade game after arcade game, pulpy adventure after trite fantasy, we're writing and rewriting, honing and perfecting. Before a single person enters one of Infocom's worlds, it must be crafted into a living, riveting, definitive experience.

Judging from the public's reaction, it's worth the wait. For instance, *Creative Computing* welcomed DEADLINE™ as "thoroughly engrossing and realistic," while a *Softalk* readers' poll recently voted ZORK™ I and ZORK II the most popular adventures of 1981.

And now, for the moment, your wait is over. ZORK III, your final step in the underground trilogy, and STARCROSS,™ an exploration of a new dimension in science fiction, are ready for you.

Look at them up there, the little worlds of Infocom. As our universe expands, companions will come to help fill that vast expanse of white space. Till then, they'll continue to stand alone as the best of all possible worlds.

INFOCOM™
55 Wheeler Street, Cambridge, MA 02138

**Infocom's worlds are available for Apple,® Atari,® IBM, TRS-80,® Commodore, NEC, Osborne, CP/M,® and DEC.™**

LOAD 80

*Find it fast in your Basic programs.*

# Search Them Strings

*John R. Gunter*
*Charles Gunter*
*756 Twin Branch Drive*
*Birmingham, AL 35226*

If necessity is the mother of invention, then aggravation must be a close relative. That's why I wrote this string search utility. Written in machine language for a 16K Level II TRS-80, it searches even the largest of Basic programs in 10 seconds or less.

Search is a small program that takes about 365 bytes of memory. I kept it short by using ROM routines extensively.

It's simple to use: Single keys command everything. It's also easy to access: You need only five keystrokes to get from Basic to the utility; only one key to return to Basic.

### Inspiring Aggravation

The aggravation that inspired this program started with a Basic simulation game that we bought. Never willing to let well enough alone, we immediately set out to modify the new program. Enter the aggravation. This was one of those packed Basic programs. You know the kind—when you list it, the screen turns almost solid white as a blizzard of symbols fly past.

After some frustrating attempts to find things in the program, we called a time-out to write the search utility.

The simulation program filled the top half of the screen with labels and numbers that were the results of complex equations. To find out which variables drove the numbers on the screen, we input the label into the search utility. A few seconds later, the print message was on the screen, complete with variable name.

Next, by searching for the variable name, we found the equation that calculated the variable.

Now, we wanted to see how the factors in the equation behaved, so we wrote some print statements for the bottom part of the screen. Just as we were starting to learn something, one of the simulation program's "disaster messages" appeared on the screen—where else, but right on top of our newly inserted print statements.

Back to the search utility, and in a few seconds the Print @ statement appeared on the screen. We moved the Print @ lower on the screen and went back to run the Basic program.

Zap! Another disaster message. A second search showed that this disaster message was another Print @ for the same location. Being clever, we detected a pattern emerging. Time for another search, this time for Print @ and the appropriate screen locations. This search located all the rest of the disaster messages, which we then moved.

We were all set! Then, all of our messages began flashing like a neon sign. We returned to the search utility once again to look for a clear to the end of the screen message (PRINT CHR$(31)). We moved all these.

Next, we wanted to write some equations of our own, but which variables were free (that is, unused in the program)? We picked a variable name we liked, and entered it as a search argument in the utility program. If the "string not found" message came back from the utility, the variable was free for us to use.

### Using the Utility

To use the utility, set the memory size to 32425. Then type SYSTEM and respond to the *? prompt with the program name SEARCH.

When the program has loaded, type / and press Enter. This loads the start address of the utility into 418FH. This is the DOS jump address for the command NAME.

Now, load the Basic program you want to search.

To access the utility, type NAME and press Enter. The screen clears and the message "output to printer (Y/N)?" appears. When you make your selection, the message "enter string:" appears. Type the string you want to search for and press Enter.

If the utility finds the string, it lists the Basic line containing it on the screen or printer, whichever you selected.

To continue searching for subsequent occurrences, press Enter. To start a new search, press Break. To return to Basic, press shift up-arrow.

When the search reaches the end of the Basic program, the message "string not found" will appear on the screen (even if you selected the printer as the output device). At this point either Break or shift up-arrow is valid.

### Details of the Utility

Search, shown in Program Listing 1 is completely relocatable. Just redefine the first line. Note that the buffer variable extends from the end of the program, so don't jam the program against something else. I allowed 28 bytes.

Lines 140 to 170 put the start address in the NAME jump address and return you to Basic.

Lines 220 to 290 ask for your choice of output device. If you don't want to use a printer, delete the lines that have an asterisk in the comment column.

If you select the printer, a one is stored in the output device control byte (lines 300–310). If not, the screen is the output device, since the byte is initialized to zero (lines 200–210).

Next, the program outputs the prompt to enter the string (lines 360–410). The character is accepted (line 450), checked for enter (line 460), stored in the buffer (lines 530–540) and displayed on the screen (line 550).

The utility stores the pointer to the first line in PRSADD (lines 650–660). Similarly, it recovers and stores the pointer to the next line in NXTADD (lines 670–710).

The contents of the Basic line are read into the input buffer using a routine from the ROM that also expands the Basic tokens (lines 770–820). (For an explanation of the structure for Basic line storage in memory, see William Barden's column "The Assembly Line" in the January, 1981 issue of *80 Microcomputing*.)

Next, Search compares the first character of the string you want to each character in the Basic program line (lines 840–960). If the first character matches (line 970), it compares subsequent characters (lines 1020–1050).

If all characters match, it outputs the line number (lines 1110–1160) and its contents (lines 1170-1220). Following this, the program waits for input (line 1230), accepting either Enter (lines 1240–1250), Break (lines 1260–1270) or Shift up-arrow (lines 1280–1290).

If it finds no match in a line, the NXTADD becomes the PRSADD. It checks to determine if this is the last line (lines 720–760). If so, it outputs the "string not found message" (lines 1310–1360); only Break or Shift up-arrow (lines 1370–1410) are valid now.

### A Basic Approach

If you haven't an assembler, don't despair. I have provided not one, but two Basic programs

*Program Listing 1. Search*

```
                00100 ;  SEARCH UTILITY FOR BASIC PROGRAMS
                00110 ;  FOR 16K LEVEL II TRS-80
                00120 ;
7EA0            00130 START   EQU     7EA0H           ;DETERMINES PROGRAM LOCATION
7E96            00140         ORG     START-10        ;SET MEM SIZE TO THIS VALUE
7E96 21A07E     00150         LD      HL,START        ;PUTS START ADDRESS IN THE
7E99 228F41     00160         LD      (418FH),HL      ;"NAME" JUMP ADDRESS
7E9C C36600     00170         JP      66H             ;RETURN TO BASIC
7EA0            00180         ORG     START
7EA0 CDC901     00190 START   CALL    1C9H            ;CLEAR THE SCREEN
7EA3 AF         00200         XOR     A               ;*SET OUTPUT DEVICE TO BE
7EA4 329C40     00210         LD      (409CH),A       ;*THE SCREEN
7EA7 21A17F     00220         LD      HL,MSSG1        ;*POINT TO MESSAGE # 1
7EAA CDA728     00230         CALL    28A7H           ;*OUTPUT IT TO THE SCREEN
7EAD CD4900     00240         CALL    49H             ;*GET A CHARACTER FROM KEYBOARD
7EB0 F5         00250         PUSH    AF              ;*SAVE THE CHARACTER AND
7EB1 CD3300     00260         CALL    33H             ;*DISPLAY IT ON THE SCREEN
7EB4 F1         00270         POP     AF              ;*RESTORE THE CHARACTER AND
7EB5 FE59       00280         CP      59H             ;*SEE IF IT'S A "Y"
7EB7 2005       00290         JR      NZ,NOPRNT       ;*IF NOT SKIP THE NEXT 2 LINES
7EB9 3E01       00300         LD      A,1             ;*SET THE OUTPUT DEVICE
7EBB 329C40     00310         LD      (409CH),A       ;*FLAG TO PRINTER
7EBE 3E0D       00320 NOPRNT  LD      A,13            ;*OUTPUT CARRIAGE RETURN
7EC0 CD3300     00330         CALL    33H             ;*TO THE SCREEN
7EC3 3E0E       00340         LD      A,14            ;TURN ON THE CURSOR
7EC5 CD3300     00350         CALL    33H             ;OUTPUT IT
7EC8 21BA7F     00360         LD      HL,MSSG2        ;POINT TO SECOND MESSAGE
7ECB 3A9C40     00370         LD      A,(409CH)       ;*GET USER'S OUTPUT DEVICE
7ECE F5         00380         PUSH    AF              ;*AND SAVE IT
7ECF AF         00390         XOR     A               ;*SET OUTPUT DEVICE TO BE
7ED0 329C40     00400         LD      (409CH),A       ;*THE SCREEN, TEMPORARILY
7ED3 CDA728     00410         CALL    28A7H           ;OUTPUT THE SECOND MESSAGE
7ED6 F1         00420         POP     AF              ;*RESTORE THE USER'S CHOICE
7ED7 329C40     00430         LD      (409CH),A       ;*OF OUTPUT DEVICE
7EDA 21E17F     00440         LD      HL,BUFFER       ;POINT TO BUFFER
7EDD CD4900     00450 READ    CALL    49H             ;GET CHARACTER FROM KEYBOARD
7EE0 FE0D       00460         CP      13              ;IS IT THE "ENTER" KEY?
7EE2 2811       00470         JR      Z,FIND          ;IF SO, GO TO FIND
7EE4 FE08       00480         CP      8               ;IS IT THE "BACK ARROW" KEY?
7EE6 2006       00490         JR      NZ,CONT         ;IF NOT, THEN GO TO CONT
7EE8 2B         00500         DEC     HL              ;DECREMENT THE BUFFER POINTER
7EE9 CD3300     00510         CALL    33H             ;BACKUP THE SCREEN CURSOR
7EEC 18EF       00520         JR      READ            ;GO GET ANOTHER CHARACTER
7EEE 77         00530 CONT    LD      (HL),A          ;PUT CHARACTER IN BUFFER
7EEF 23         00540         INC     HL              ;INCREMENT THE BUFFER POINTER
7EF0 CD3300     00550         CALL    33H             ;PUT CHARACTER ON SCREEN
7EF3 18E8       00560         JR      READ            ;G_ GET ANOTHER CHARACTER
                00570 ;
                00580 ;BEGINING OF SEARCH ROUTINE
                00590 ;
7EF5 3600       00600 FIND    LD      (HL),0          ;MARK END OF BUFFER WITH ZERO
7EF7 3E0D       00610         LD      A,13            ;PUT A CARRIAGE RETURN
7EF9 CD3300     00620         CALL    33H             ;OUT TO THE SCREEN
7EFC 3E0F       00630         LD      A,15            ;TURN OFF THE CURSOR
7EFE CD3300     00640         CALL    33H             ;OUTPUT IT TO THE SCREEN
7F01 2AA440     00650         LD      HL,(40A4H)      ;GET ADDRESS OF FIRST LINE
7F04 22DB7F     00660 NWLINE  LD      (PRSADD),HL     ;PUT IT IN PRESENT ADDRESS
7F07 7E         00670         LD      A,(HL)          ;GET THE LOW BYTE OF THE NEXT
7F08 32DD7F     00680         LD      (NXTADD),A      ;LINE ADDRESS AND STORE IT
7F0B 23         00690         INC     HL
7F0C 7E         00700         LD      A,(HL)          ;GET THE HIGH BYTE OF THE NEXT
7F0D 32DE7F     00710         LD      (NXTADD+1),A    ;LINE ADDRESS AND STORE IT
7F10 2ADB7F     00720         LD      HL,(PRSADD)     ;GET THE PRESENT LINE ADDRESS
7F13 7C         00730         LD      A,H             ;CHECK TO SEE IF THE VALUE
7F14 B5         00740         OR      L               ;IS ZERO WHICH INDICATES THE
7F15 FE00       00750         CP      0               ;END OF THE BASIC PROGRAM
7F17 286A       00760         JR      Z,THRU          ;IF SO, GO TO THRU
7F19 23         00770         INC     HL              ;MOVE OVER
7F1A 23         00780         INC     HL              ;TO THE FIRST
7F1B 23         00790         INC     HL              ;CHARACTER OF THE
7F1C 23         00800         INC     HL              ;BASIC PROGRAM LINE
7F1D CD7E2B     00810         CALL    2B7EH           ;PUT LINE IN INPUT BUFFER
7F20 21E841     00820         LD      HL,41E8H        ;POINT TO ADDRESS OF INPUT BUFFE
7F23 11E17F     00830 BUFF1   LD      DE,BUFFER       ;POINT TO SEARCH PROGRAM BUFFER
7F26 AF         00840         XOR     A               ;INITIALIZE THE MATCH FLAG
7F27 32DF7F     00850         LD      (MTCHFL),A      ;WITH A ZERO--NO MATCH
7F2A 1A         00860 BUFFNX  LD      A,(DE)          ;GET NEXT CHARACTER FROM THE
7F2B 47         00870         LD      B,A             ;SEARCH PROGRAM'S BUFFER
7F2C AF         00880         XOR     A               ;AND CHECK TO SEE IF IT IS
7F2D B8         00890         CP      B               ;THE END OF BUFFER MARKER
7F2E 281F       00900         JR      Z,MATCH         ;IF SO, GO TO THE MATCH ROUTINE
7F30 7E         00910 NXCHAR  LD      A,(HL)          ;GET NEXT CHAR FROM INPUT BUFFER
7F31 FE00       00920         CP      0               ;IS THIS THE END OF THIS LINE?
7F33 2815       00930         JR      Z,NXTLIN        ;IF SO, GO TO GET NEXT LINE
7F35 23         00940         INC     HL              ;DOES THE CHARACTER FROM THE
7F36 B8         00950         CP      B               ;SEARCH BUFFER (B) MATCH THE
7F37 2809       00960         JR      Z,MATCH1        ;ONE FROM INPUT? IF SO GO
7F39 3ADF7F     00970         LD      A,(MTCHFL)      ;HAVE WE PREVIOUSLY MATCHED
7F3C FE00       00980         CP      0               ;ANY CHARACTERS IN THIS LINE?
7F3E 28F0       00990         JR      Z,NXCHAR        ;IF NOT, GET THE NEXT CHAR
```

*Listing 1 continues*

*Listing 1 continued*

```
7F40 18E1                  JR     BUFF1         ;IF SO, GO TO BUFF1
7F42 3E01          MATCH1  LD     A,1           ;WHEN THE FIRST CHAR MATCHES
7F44 32DF7F                LD     (MTCHFL),A    ;SET THE MATCH FLAG AND
7F47 13                    INC    DE            ;POINT TO THE NEXT CHARACTER
7F48 18E0                  JR     BUFFNX        ;AND GO TO BUFFNX
7F4A 2ADD7F        NXTLIN  LD     HL,(NXTADD)   ;TAKE THE VALUE OF THE NEXT
7F4D 18B5                  JR     NWLINE        ;ADDRESS AND GO
                   ;
                   ;ROUTINE WHEN A COMPLETE MATCH HAS BEEN FOUND
                   ;
7F4F DD2ADB7F      MATCH   LD     IX,(PRSADD)   ;GET THE PRESENT ADDRESS
7F53 DD23                  INC    IX            ;AND MOVE OVER TO THE
7F55 DD23                  INC    IX            ;PROGRAM LINE NUMBER
7F57 DD6E00                LD     L,(IX+0)      ;GET THE LOW AND HIGH BYTES
7F5A DD6601                LD     H,(IX+1)      ;OF THE LINE NUMBER
7F5D CDAF0F                CALL   0FAFH         ;OUTPUT THE LINE NUMBER
7F60 3E20                  LD     A,32          ;GET A SPACE CHARACTER
7F62 CD2A03                CALL   32AH          ;AND OUTPUT IT
7F65 21E841                LD     HL,41E8H      ;POINT TO THE INPUT BUFFER
7F68 CD752B                CALL   2B75H         ;OUTPUT THE LINE
7F6B 3E0D                  LD     A,13          ;GET A CARRIAGE RETURN
7F6D CD2A03                CALL   32AH          ;AND OUTPUT IT
7F70 CD4900        LOOP    CALL   49H           ;GET CHARACTER FROM KEYBOARD
7F73 FE0D                  CP     13            ;IS IT THE "ENTER" KEY?
7F75 28D3                  JR     Z,NXTLIN      ;IF SO, CONTINUE THE SEARCH
7F77 FE01                  CP     1             ;IS IT THE "BREAK" KEY?
7F79 CAA07E                JP     Z,START       ;IF SO, START OVER
7F7C FE1B                  CP     27            ;IS IT SHIFT UP ARROW?
7F7E CA191A                JP     Z,1A19H       ;IF SO, RETURN TO BASIC
7F81 18ED                  JR     LOOP          ;IF NONE OF ABOVE, TRY AGAIN
7F83 21CA7F        THRU    LD     HL,MSSG3      ;POINT TO MESSAGE 3
7F86 AF                    XOR    A             ;*CHANGE THE OUTPUT DEVICE TO
7F87 329C40                LD     (409CH),A     ;*THE SCREEN
7F8A CDA728                CALL   28A7H         ;OUTPUT MESSAGE 3
7F8D 3E0D                  LD     A,13          ;GET A CARRIAGE RETURN
7F8F CD2A03                CALL   32AH          ;AND OUTPUT IT
7F92 CD4900        THRUIN  CALL   49H           ;GET A CHARACTER FROM KEYBOARD
7F95 FE01                  CP     1             ;IS IT THE "BREAK" KEY?
7F97 CAA07E                JP     Z,START       ;IF SO, START OVER
7F9A FE1B                  CP     27            ;IS IT THE SHIFT UP ARR W?
7F9C CA191A                JP     Z,1A19H       ;IF SO, GO TO BASIC
7F9F 18F1                  JR     THRUIN        ;IF NONE OF ABOVE, TRY AGAIN
7FA1 4F            MSSG1   DEFM   'OUTPUT TO PRINTER (Y/N)?'     ;*
7FB9 00                    DEFB   0                              ;*
7FBA 45            MSSG2   DEFM   'ENTER STRING:  '
7FC9 00                    DEFB   0
7FCA 53            MSSG3   DEFM   'STRING NOT FOUND'
7FDA 00                    DEFB   0
7FDB 0000          PRSADD  DEFW   0
7FDD 0000          NXTADD  DEFW   0
7FDF 0000          MTCHFL  DEFW   0
7FE1 0000          BUFFER  DEFW   0
7E96                       END    START-10
00000 TOTAL ERRORS
```

to POKE the machine language program into memory.

The first version (see Program Listing 2) is the same as the assembly program. With the other version you cannot use the printer and it's shorter (see Program Listing 3).

With either version, set the memory size to the value shown in the first remark and type the program.

When you run the program, you first get a poor man's cyclical checksum of your data. This checks for typographical indiscretions and transpositions. If you pass this check you should be in good shape.

When the program completes its run and returns the ready prompt, try a few searches on the Basic program that POKEd in the data.

Assuming that works, save your Basic program on tape, and it's ready for serious use.

### Tips on Using the Utility

Remember that in Basic only the first two letters are significant. Suppose you want to introduce a new variable to hold the last value input, and you want to call it Last. Search only for LA, but bear in mind the program will also find a variable named Large. Remember, defining too large a search argument will cause you to miss conflicting variables that have the same first two letters.

Watch out for embedded blanks. A search for:

PRINT @ 660

will not find:

PRINT      @ 660

You can search for references to line numbers too. Suppose you plan to delete lines 1120 to 1180 from your Basic program. Before you do so, you can quickly find out whether any GOTOs, GOSUBs or If...Thens reference these lines by searching for 11. Any reference will quickly show up.

Try this utility to soothe your frustrations, and speed your way through your next Basic programming job. ■

```
10 '     MEMORY SIZE = 32415
20 '  LINES 30 TO 110 DO THE CHECKSUM DATA VALIDATION
30 DEFINT D,Z : CHECKSUM = 0
40 FOR K = 1 TO 7 STEP 2
50 READ D
60 IF D = -1 THEN 100
70 CHECKSUM = CHECKSUM + D * K
80 NEXT K
90 GOTO 40
100 IF CHECKSUM <> 116705 PRINT "CHECKSUM ERROR--RECHECK DATA ST
ATEMENTS":END
110 PRINT "DATA CHECKSUM OK"
120 '  LINES 130 TO 190 POKE THE MACHINE LANGUAGE INTO HIGH MEMO
RY
130 RESTORE
140 K = 32416
150 READ D
160 IF D = -1 THEN 210
170 POKE K,D
180 K = K + 1
190 GOTO 150
200 '  LINE 210 POKES THE JUMP ADDRESS OF THE "NAME" COMMAND
210 POKE 16783,160 : POKE 16784,126
220 PRINT "FINISHED"
230 DATA 205,201
240 DATA 1,175,50,156,64,33,164,127,205,167,40,205
250 DATA 73,0,245,205,51,0,241,254,89,32,5,62
260 DATA 1,50,156,64,62,13,205,51,0,62,14,205
270 DATA 51,0,33,189,127,50,156,64,245,175,50,156
280 DATA 64,205,167,40,241,50,156,64,33,220,127,205
290 DATA 73,0,254,13,40,17,254,0,32,6,43,205
300 DATA 51,0,24,239,119,35,205,51,0,24,232,54
310 DATA 0,62,13,205,51,0,62,15,205,51,0,42
320 DATA 164,64,34,222,127,126,50,224,127,35,126,50
330 DATA 225,127,42,224,127,124,181,254,0,40,109,42
340 DATA 222,127,35,35,35,35,205,126,43,33,232,65
350 DATA 17,220,127,175,50,226,127,26,71,175,184,40
360 DATA 31,126,254,0,40,21,35,184,40,9,58,226
370 DATA 127,254,0,40,240,24,225,62,1,50,226,127
380 DATA 19,24,224,42,224,127,24,170,221,42,222,127
390 DATA 221,35,221,35,221,110,0,221,102,1,205,175
400 DATA 15,62,32,205,42,3,33,232,65,205,117,43
410 DATA 62,13,205,42,3,205,73,0,254,13,40,211
420 DATA 254,1,202,160,126,254,27,202,102,0,24,237
430 DATA 33,205,127,175,50,156,64,205,167,40,62,13
440 DATA 205,42,3,205,73,0,254,1,202,160,126,254
450 DATA 27,202,102,0,24,241,79,85,84,80,85,84
460 DATA 32,84,79,32,80,82,73,78,84,69,82,32
470 DATA 40,89,47,78,41,63,0,69,78,84,69,82
480 DATA 32,83,84,82,73,78,71,58,32,32,0,83
490 DATA 84,82,73,78,71,32,78,79,84,32,70,79
500 DATA 85,78,68,0,-1
```

*Program Listing 2. Basic Program to POKE Search*

```
10 '     SET MEMORY SIZE TO 32485
20 '  LINES 30 TO 110 DO THE CHECKSUM DATA VALIDATION
30 DEFINT D,K : CHECKSUM = 0
40 FOR K = 1 TO 7 STEP 2
50 READ D
60 IF D = -1 THEN 100
70 CHECKSUM = CHECKSUM + D * K
80 NEXT K
90 GOTO 40
100 IF CHECKSUM <> 92158 PRINT "CHECKSUM ERROR--RECHECK DATA STA
TEMENTS":END
110 PRINT "DATA CHECKSUM OK"
120 '  LINES 130 TO 190 POKE THE MACHINE LANGUAGE INTO HIGH MEMO
RY
130 RESTORE
140 K = 32489
150 READ D
160 IF D = -1 THEN 210
170 POKE K,D
180 K = K + 1
190 GOTO 150
200 '  LINE 210 POKES THE JUMP ADDRESS OF THE "NAME" COMMAND
210 POKE 16783,233 : POKE 16784,126
220 PRINT "FINSIHED"
230 DATA 205,201,1,62,14,205,51,0,33,189,127,205
240 DATA 167,40,33,228,127,205,73,0,254,13,40,17
250 DATA 254,8,32,6,43,205,51,0,24,239,119,35
260 DATA 205,51,0,24,232,54,0,62,13,205,51,0
270 DATA 62,15,205,51,0,42,164,64,34,222,127,126
280 DATA 50,224,127,35,126,50,225,127,42,224,127,1
290 DATA 181,254,0,40,109,42,222,127,35,35,35,35
300 DATA 205,126,43,33,222,65,17,228,127,175,50,226
310 DATA 127,26,71,175,184,40,31,126,254,0,40,21
320 DATA 35,184,40,9,58,226,127,254,0,40,240,24
330 DATA 225,62,1,50,226,127,19,24,224,42,224,127
340 DATA 24,176,221,42,222,127,221,35,221,35,221,110
350 DATA 0,221,102,1,205,175,15,62,32,205,42,3
360 DATA 33,232,65,205,117,43,62,13,205,42,3,205
370 DATA 73,0,254,13,40,211,254,1,202,233,126,254
380 DATA 27,202,102,0,24,237,33,205,127,205,167,40
390 DATA 62,13,205,42,3,205,73,0,254,1,202,233
400 DATA 126,254,27,202,102,0,24,241,69,78,84,69
410 DATA 82,32,83,84,82,73,78,71,58,32,32,0
420 DATA 83,54,82,73,78,71,32,78,79,84,32,70
430 DATA 79,65,78,68,0,-1
```

*Program Listing 3. Basic Version Without Print Option*





✓ See List of Advertisers on Page 451

**The Key Box**

**Model I or III**
**Cassette Basic**

*Eliminate bad saves and wasted time.*

# Certify Your Cassettes

*Kenneth J. Bigelow*
*4117 55th Avenue*
*Bladensburg, MD 20710*

After spending three days keying in a new adventure program, to protect it you CSAVE it before running it. A long program saves in half an hour. When you come back, Ready is prompting and the computer awaits you.

With the program on tape you are protected. Right? Even if it crashes, you have taped it. Right? Maybe. Before you power down or run your program, you check the recorded tape using CLOAD? to play back the tape and compare it with what is still in memory. That will tell you whether or not your program was correctly recorded. If it's not, you'd better CSAVE it all over before you turn off the machine—you don't want to key it in by hand all over again!

If you already CLOAD? your tapes right after recording that's a good start. So once again you type CLOAD? and knowing that it will take just as long to check the tape as to record it, you leave to do something else for another 30 minutes. But when you get back, the screen says Bad just above the Ready prompt. A quick check of the recorder shows that the tape only ran for two minutes before the computer found a mistake on the tape. You rewind the tape and CLOAD? again. This time you watch while the computer starts its check. Sure enough, two minutes later, the message Bad appears and the tape stops.

Well, maybe the head is dirty. It has been a while since you cleaned it. Out comes the head cleaner tape, the goo and the swabs. Then, your tape goes back in and you CLOAD? again. Two minutes later Bad appears again and the tape stops. This time, you decide the recording was bad, probably due to the dirty heads. So, you rewind the tape and CSAVE the program again. Another 30 minutes later you rewind the tape, cross your fingers and CLOAD? one more time. You say a prayer in computer language and watch the screen anxiously. But no luck. Two minutes later Bad appears once more and there does not seem to be anything you can do about it.

### Tape Dropout

The problem is "dropout" on the tape. A dropout is a spot where the magnetic medium is thinner than necessary, or absent altogether. That spot on the tape can hold no data. In ordinary audio use, this spot would go unnoticed. But the pulses used by the TRS-80 are very narrow and must occur in the right places. If one of those pulses occurs right on the dropout it is not recorded. On playback, the computer spots the missing pulse because it does not match the contents of memory. At that point the CLOAD? operation stops, indicating a bad tape.

Can you prevent this? One way is to buy only certified cassettes. These tapes have been tested and are guaranteed to contain no dropouts. Unfortunately, they are also expensive. The next choice is top-of-the-line cassettes. These tapes are less expensive than certified tapes, and *usually* contain no dropouts. However, there is no guarantee, and you can expect an occasional defect.

Two options are available to you: buying the more expensive tapes or tolerating the occasional defective cassette. Neither option is particularly satisfying. Testing the tape before wasting time recording would solve the problem, but what do we look for? The computer does the hard part.

### Checking the Tape

Make a visual inspection. Turn the take-up spool past the blank leader and inspect the tape surface. Reject any tape that is even slightly rough. Some of the 3-for-a-dollar or 25¢ tapes might as well have been polished by a knurling tool and look more like alligator hide than recording tape. The abrasive surface will wear your tape heads early. I buy these cassettes if the housing is screwed together, well made, and if wheels guide the tape. I throw away the tape and splice a saved program into the empty housing. The tape itself is dreadful!

When you have a smooth tape surface, let the computer hunt for dropouts. The computer will write a pattern of 1's onto the tape and then read them back. If you receive a non-1 pattern on playback, the tape had a dropout at that spot. If only 1's read back the tape is OK.

To understand how this works, let's take a look at the format of data on the tape. Figure 1 shows how it works. The cassette output routine generates a series of clock pulses and tapes them. One such clock pulse is generated every two milliseconds (one millisecond is equal to .001 second). The pause is left for data pulses. If a 1-bit is recorded, a data pulse appears halfway between the pulses. If a 0-bit is recorded, no pulse appears between pulses. Therefore, if we record an all-one pattern, the tape places a pulse every millisecond. The tape speed in audio recorders is standardized at 1 7/8 inches per second. The computer will pulse at intervals of .001875 inch.

Since Basic is too slow for such operations, read and write routines must be in machine language. But having the main part of the program in Basic facilitates entry and running. Also, with Basic we can enter the program without a lot of POKEs, T-Bug, EDTASM or any other utility programs and get back and forth between Basic and the machine language subroutines with the USR function.

### Writing the Program

Now write the Basic program to generate the required machine language subroutines

Learn it in a FLASH
michael shrayer's
ELECTRIC
pencil
word processing system
● Easy to Learn – Like lightning you will be up and running with a comprehensive and sophisticated word processor. Our manual says you can be an expert in one hour, our users tell us it takes less than 30 minutes. The manual's 128 pages are packed with figures, illustrations and examples for the beginner and "old pro".
● Easy to Use – With the ELECTRIC PENCIL you process words, not commands. ELECTRIC PENCIL's menus and simple two keystroke commands keeps your mind on your work, NOT on your manual. ELECTRIC PENCIL is so easy to use your secretary will WANT to use it... so sophisticated you'll be glad she did. David Ahl, the editor of Creative Computing says, "ELECTRIC PENCIL 2.0 is the most user friendly word processing package available."
● Features – ELECTRIC PENCIL has a 36k+ text buffer (48k disk system), supports disk, tape and Stringy Floppy files (disk version)* and has every major feature you want in a word processor. It is "bullet proof" – Data recovery is a built-in feature. The exclusive DICT-A-MATIC† feature gives your office the flexibility of dictation-word processing without fancy equipment. Your dictation playback is controlled from the computer keyboard.
● Flexible – It is the only word processing system that is designed like an operating system. You can add new programs and features to ELECTRIC PENCIL, such as RED and BLUE PENCIL dictionary/correction – with a 50,000 word dictionary (sold separately) – without patches and upgrades. Simply press one of the control key combinations and new additions to your system are instantly available. Add automatic proofing, spelling correction, dynamic print formatting, proportional printing, communications, graphics and typesetting and many other add-on features when you need them.
● Versatile – ELECTRIC PENCIL runs on TRS-80 model I and model III computers under all versions of TRSDOS, and NEWDOS without modification or patching (Patches required for other operating systems). It also supports parallel and serial printers as well as single and double density disk systems.
ELECTRIC PENCIL is the choice of thousands - make it yours for only $89.95 (disk version) or $79.95 (tape and Stringy Floppy versions). Manual only $24.95.
ELECTRIC PENCIL is available at computer stores, selected B. Dalton Booksellers and selected independent book dealers. If your dealer is out of stock, order direct (specify disk tape or Stringy Floppy version). Include $4.00 for shipping and handling. Foreign residents add $11.00 plus purchase price, in U.S. funds.
*Tape and Stringy Floppy versions support tape and stringy files only
1953 West
11th Street
Upland, CA
91786 (714)
946-5805
Stringy Floppy trademark EXATRON Corp.
NEWDOS trademark APPARAT, Inc.
TRS80, TRSDOS trademark TANDY Corp.
† trademark IJG, Inc.

when run. The program should run whether the system is Level II Basic, Model III Basic or Disk Basic. The machine language portions should be relocatable; the computer decides where to put them as well as the Basic program. This eliminates POKE commands, because we don't know where to POKE either subroutine. These are character strings and VARPTR finds them. This works with the Model I or the Model III, using Level II or Model III Basic.

For Disk Basic to work, a DEFUSR statement must replace the POKE statements. The USR call must also be modified. Either way, when character strings hold the subroutines an version of Basic can place them into a memory location. This protects the Basic operation, whether you have reserved memory during initialization or not. Both subroutines automatically land in the reserved string space to be retrieved by the VARPTR function.

Listing 1 shows the final Basic program. In this program lines 10 and 20 contain the data (in decimal form) which generate the write and read subroutines. Lines 30 and 40 read these data lists and convert them into character strings W$ and R$.

Lines 50 and 60 ask whether the computer is to write or to read from the tape. Then they set M$ equal to the selected subroutine. M$ will be the character string executed as a subroutine.

In line 70 VARPTR locates M$ in memory and sets the USR pointer to that address. VARPTR returns a memory address related to the variable name the VARPTR argument uses. With string variables, the VARPTR returns an address containing the length of the string. The next two addresses contain the string address itself in memory. Therefore, VARPTR locates the subroutine in memory and we can POKE its address into the USR pointer (locations 16526 and 16527).

Line 80 indicates on screen that the computer accepted input and begins the required subroutine. The message is different for read and write; you can tell at a glance what the machine is doing.

Line 90 executes the USR call. The Radio Shack manual suggests a command like X = USR(0). However, any other command that can specify a function works as well. The Print key is easier to hit than X =. Besides, who needs to define another variable at this stage?

Line 100 closes the program loop. Execution repeats from line 50. Both subroutines stop any time you hit either Shift key. You can regain control without resetting. This permits you to stop checking a bad tape and start over.

**Subroutines**

The Basic program provides the necessary man-machine interface and calls the appropriate machine-language subroutine. The machine-language subroutines prepare and check the tape. You need not enter these subroutines along with the Basic program. Part of the Basic program generates and stores these subroutines. Only the Basic program need be entered and run. The subroutine assembly listings explain how they work.

The write subroutine (see Listing 2) contains 20 bytes. After turning on the recorder and recording the leader and sync byte, the subroutine enters a

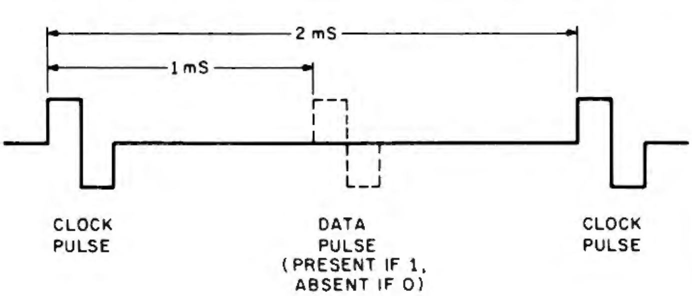



*Fig. 1. The data format for TRS-80 tapes*

*See List of Advertisers on Page 451*

loop which writes hex FF (a pattern of eight 1 bits with their clock pulses) to the tape and then examines the Shift keys. If no Shift key is pressed the loop repeats from WCERT1 in the listing. This loop will continue indefinitely. The Shift key terminates the subroutine by turning off the recorder and returning to the Basic program.

> *"Read and write routines must be in machine language."*

When the Basic program reappears, it prints the digit 0 in response to the Print statement in line 90. It returns to line 50. Since line 50 starts with a CLS command, the zero will disappear, as soon as it is displayed and may even be invisible.

In the Model I TRS-80, the ROM routine CASSON turns on the recorder. If your system includes the expansion interface and you have two recorders, you may wish to preselect the recorder. Entering CASSON at address 0215H can avoid switching recorders. Change the second data element in line 10 of the Basic program from 18 to 21. The rest of the program remains unchanged.

In Model III Basic, CASSON is unnecessary. The CSHOUT routine writes the leader and sync byte to the tape, and turns on the recorder. However, the call to CASSON can remain. The Model III ROM essentially ignores such calls anyway. The only instructions executed by the Model III are XOR A followed by RET. Leaving the call to CASSON in place allows the subroutine to run directly on the Model I (Level II Basic) or the Model III (Model III Basic).

Listing 3 shows the read subroutine—RCERT. It is longer than the write routine (36 bytes) because it has to do more. CASSON turns on the recorder. Again, systems including two recorders may change the entry address to 0215H, by changing the second data element in line 20 of the Basic program from 18 to 21.

### Test Reliability Indicators

We want assurance that the program is reading correct data from tape. A question mark for every wrong byte should show how good the tape is. In RCERT the HL register pair acts as a screen pointer and keeps the ? code in register C.

Once the registers have been initialized, RCERT calls CSHIN in ROM. The leader code of the internal subroutine synchronizes to the clock pulses and checks for the sync code recorded earlier by CSHOUT. The computer will differentiate clock pulses from data pulses, and know where bytes end and begin. In the Model III, as with CSHOUT, CSHIN also activates the recorder.

The next step is to read and check data from the tape. Regardless of the tape's condition, it helps to verify that data was really transmitting from the tape. Since the TRS-80 normally uses the blinking asterisk in the upper right corner we can use the same method. There will be no end-of-line codes on the tape to blink the asterisk automatically, so let's blink it every 64 bytes for a nice steady display. Register B will be the counter and a DJNZ instruction will end the loop.

Now CSIN, the ROM routine, will read a byte from tape. If it is really hex FF we skip the error display code and go to RCERT3 in the listing. If another pattern than FF is recovered from tape the error code will signal. A question mark occurs at the location the HL register pair designates. Then L is incremented to advance the pointer to the next spot. This allows a maximum of 256 question marks (four lines). We could have more, but if the tape is bad you will know by now. If the question marks exceed screen memory, they will overwrite regular memory and destroy all the Basic pointers and vector addresses. To avoid problems restrict the number of question marks.

**Stopping the Program**

At RCERT3 the subroutine looks at the shift keys. As with WCERT, this stops the program any time the computer reads data from the tape. However, WCERT can stop whether or not the tape is moving. RCERT can only stop after a byte reads from tape. While the tape is moving there will be no problem. Once the tape stops RCERT will look for another byte until it gets one. It will check the shift keys again.

In the absence of a Shift, the DJNZ instruction is performed. Normally, this causes a branch back to RCERT2 which reads another byte. This loop repeats until 64 bytes (correct or not) are received from the tape. At that point the DJNZ falls through and the call to Blink will be executed. This ROM routine causes the righthand asterisk to blink on or off every 64 bytes. At 500 baud, this is about once a second. After the asterisk switches, a JR instruction returns the program to RCERT1, which reloads register B before reading the next byte from the tape. This restarts the sequence.

The Blink routine is absent in the Model III. CSIN blinks the asterisk. This is why the asterisk blinks regularly on the Model III. As with CASSON, a call to Blink will do no harm. In fact, this location in the Model III consists of a JR to the same XOR A, RET sequence which replaces CASSON.

**Testing**

To use the program, key it in and type Run. "(W)RITE OR (R)EAD?" will appear. Load a tape and advance it past the blank leader. Press Play and Record. Now, type W (Enter). The display will change to "Press 'Shift' To Stop." A C-60 takes about 30 minutes to load.

A Shift terminates the subroutine and displays the original message: "(W)RITE OR (R)EAD?" Turn the cassette over and advance the tape to the end of the blank leader. Place the tape back in the recorder, press Play and Record, and once more type W (Enter) into the computer. Repeat the process to record the test pattern on side two.

Shift ends the subroutine and returns to the Basic program. Return the tape to side one, load it and press Play and type R (Enter) into the computer. The computer displays "Reading", and the recorder starts. At the sync byte asterisks appear. The right hand asterisk blinks at about a one-per-second rate indicating that the computer is reading data. (The Model III blinks at its own rate.)

As long as the data is correct (All 1's, or hex FF's), the display will not change. If the display is blank except for the top line, the tape is good. If a pulse is missing, a question mark appears as indicated by a zero bit recovered from tape up to a maximum of four lines of question marks.

*"If a pulse is missing, a question mark will appear on the screen."*

**Leader Problems**

The first few seconds of the leader (composed of all 0's to insert the clock pulses) and the sync byte (hex A5) prevent checking this part of the tape as closely as the rest. Nevertheless, this code allows the computer to synchronize itself, byte for byte, with the data coming





See List of Advertisers on Page 451

in. Otherwise timing errors creep in and cause apparent faults where none exist.

If a few question marks begin the tape you need not discard the entire tape. Mechanical damage at one end or the other is common enough. To save the rest, note where the last question mark appears and delete that segment. Splice the remaining tape to the plastic leader. Usually, you will not lose more than 6 to 12 inches.

One to four question marks at the end, just as the plastic leader moves under the playback head, is usually caused by the last incomplete byte recorded previously, and noise generated at crossover. There is no cure for such difficulties; simply make sure you have enough tape to record your program.

### Splicing

You can locate a single fault by pressing Shift as the question mark appears to stop the tape. Then delete the section and splice the ends together. (The splice desynchronizes the tape and appears as a defect. Always play the entire tape to determine the bad spots. If repair is worth it, locate the bad spot or you will have to re-record the entire tape to check the remainder.)

If multiple faults punctuate the entire tape, clean the heads and pinch roller of your recorder. Dirty heads make even the best tape produce garbage. Also check playback volume. This is rather touchy with TRS-80's. The Model III needs a higher playback volume than the Model I. If your results are no better with clean heads and correct volumes, discard the tape.

### Sidestepping the Pitfalls

The certify program on the Model I TRS-80 is simple and straightforward. Be sure to record the program before you turn off the machine. The higher playback volume required by the Model III depends on clean heads and a good recorder even more than the Model I.

The tape can be written at 1500 baud, but playback is something else. If an error occurs and a question mark appears, the computer can lose synchronization with the recorded data. If this happens the slight extra time lag between bytes will foul up the playback timing for the rest of the tape. As a result, the computer will treat the entire tape as defective, even if it actually had only one bad spot which minor surgery could easily remove.

It takes thirty minutes to read or write each side of a C-60 cassette, whether the rate is 500 or

```
1 'CASSETTE TAPE CERTIFY PROGRAM.
2 'WRITTEN BY KENNETH J. BIGELOW.
3 '
4 'WILL RUN ON ANY TRS-80 USING LEVEL 2 OR MODEL 3 BASIC.
5 'MUST BE MODIFIED FOR DISC BASIC.
6 '
8 'DATA FOR WRITE SUBROUTINE.
9 '
10 DATA 205,18,2,205,135,2,62,255,205,100,2,58,128,56,183,40,245
,195,248,1
17 '
18 'DATA FOR READ SUBROUTINE.
19 '
20 DATA 205,18,2,33,0,61,14,63,205,150,2,6,64,205,53,2,254,255,4
0,2,113,44,58,128,56,183,194,248,1,16,238,205,44,2,24,231
27 '
28 'PUT SUBROUTINES INTO MEMORY AS CHARACTER STRINGS.
29 '
30 CLEAR150:W$="":FORI=1TO20:READW:W$=W$+CHR$(W):NEXT
40 R$="":FORI=1TO36:READR:R$=R$+CHR$(R):NEXT:RESTORE
47 '
48 'GET FUNCTION FROM USER.
49 '
50 CLS:INPUT"(W)RITE OR (R)EAD";A$:M$=W$:IFA$="W"THEN70
60 M$=R$:IFA$<>"R"THEN50
66 '
67 'AT THIS POINT, M$ CONTAINS THE ACTUAL SUBROUTINE TO BE RUN.
68 'VARPTR IS USED TO FIND IT AND SET THE USR POINTER.
69 '
70 POKE16526,PEEK(VARPTR(M$)+1):POKE16527,PEEK(VARPTR(M$)+2)
80 CLS:IFA$="W"THENPRINT"PRESS 'SHIFT' TO STOP."ELSE PRINT"READI
NG."
87 '
88 'EXECUTE THE SELECTED SUBROUTINE.
89 '
90 PRINTUSR(0)
97 '
98 'AFTER TERMINATING SUBROUTINE, ASK FOR NEW COMMAND.
99 '
100 GOTO 50
```

*Program Listing 1*

1500 baud. 500 baud helps ensure greater reliability when running this program.

Regardless of which machine you use, keep the program to run whenever your computer is idle. It might just as well be testing tapes. ■

```
                00100 ;WRITE SUBROUTINE FOR TAPE CERTIFY PROGRAM.
                00110 ;THIS ROUTINE IS ENTIRELY RELOCATABLE.
                00120 ;
                00130 ;
                00140 ;ALL CASSETTE HANDLING ROUTINES ARE IN ROM:
                00150 ;
01F8            00160 CASOFF  EQU     01F8H           ;TURN RECORD
ER OFF.
0212            00170 CASSON  EQU     0212H           ;TURN RECORD
ER ON.
0264            00180 CSOUT   EQU     0264H           ;WRITE (A) T
O TAPE.
0287            00190 CSHOUT  EQU     0287H           ;WRITE LEADE
R/SYNC BYTE.
                00200 ;
                00210 ;
                00220 ;TO TERMINATE, THIS ROUTINE CHECKS THE SHIFT
 KEYS AT
                00230 ;ADDRESS 3800H.  IF PRESSED, THE ROUTINE RET
URNS TO
                00240 ;BASIC.
                00250 ;
0000            00260         ORG     0               ;DUMMY START
ING ADDRESS.
0000 CD1202     00270 WCERT   CALL    CASSON          ;START RECOR
DER.
0003 CD8702     00280         CALL    CSHOUT          ;WRITE LEADE
R & SYNC.
0006 3EFF       00290 WCERT1  LD      A,0FFH          ;USE ALL 1'S
 AS PATTERN.
0008 CD6402     00300         CALL    CSOUT           ;WRITE BYTE
TO TAPE.
000B 3A0038     00310         LD      A,(3800H)       ;CHECK SHIFT
 KEYS.
000E B7         00320         OR      A               ;SHIFT KEY P
RESSED?
000F 28F5       00330         JR      Z,WCERT1        ;IF NOT, REP
EAT LOOP.
0011 C3F801     00340         JP      CASOFF          ;RECORDER OF
F AND RETURN.
                00350 ;
0000            00360         END
00000 TOTAL ERRORS
```

*Program Listing 2*

```
                00100 ;READ SUBROUTINE FOR TAPE CERTIFY PROGRAM.
                00110 ;THIS ROUTINE IS ENTIRELY RELOCATABLE.
                00120 ;
                00130 ;
                00140 ;ALL CASSETTE HANDLING ROUTINES ARE IN ROM.
                00150 ;
01F8            00160 CASOFF  EQU     01F8H           ;TURN RECORD
ER OFF.
0212            00170 CASSON  EQU     0212H           ;TURN RECORD
ER ON.
022C            00180 BLINK   EQU     022CH           ;BLINK RIGHT
 ASTERISK.
0235            00190 CSIN    EQU     0235H           ;READ A BYTE
 FROM TAPE.
0296            00200 CSHIN   EQU     0296H           ;READ HEADER
/SYNC BYTE.
                00210 ;
0000            00220         ORG     0               ;DUMMY START
ING ADDRESS.
0000 CD1202     00230 RCERT   CALL    CASSON          ;START TAPE.

0003 21003D     00240         LD      HL,3D00H        ;POINT TO VI
DEO LINE 5.
0006 0E3F       00250         LD      C,'?'           ;PREPARE QUE
STION MARK.
0008 CD9602     00260         CALL    CSHIN           ;WAIT FOR SY
NC BYTE.
000B 0640       00270 RCERT1  LD      B,64            ;BLINK EVERY
 64 BYTES.
000D CD3502     00280 RCERT2  CALL    CSIN            ;READ A BYTE
.
0010 FEFF       00290         CP      0FFH            ;IS IT ALL 1
'S?
0012 2802       00300         JR      Z,RCERT3        ;IF SO, FINE
.
0014 71         00310         LD      (HL),C          ;IF NOT, PRI
NT OUT
0015 2C         00320         INC     L               ;A QUESTION
MARK.
0016 3A0038     00330 RCERT3  LD      A,(3800H)       ;LOOK FOR SH
IFT KEY.
0019 B7         00340         OR      A               ;PRESSED?
001A C2F801     00350         JP      NZ,CASOFF       ;IF SO, QUIT
.
001D 10EE       00360         DJNZ    RCERT2          ;REPEAT LOOP
 64 TIMES.
001F CD2C02     00370         CALL    BLINK           ;BLINK ASTER
ISK.
0022 18E7       00380         JR      RCERT1          ;REPEAT INDE
FINITELY.
                00390 ;
0000            00400         END
00000 TOTAL ERRORS
```

*Program Listing 3*

See List of Advertisers on Page 451

# COMPUTEX

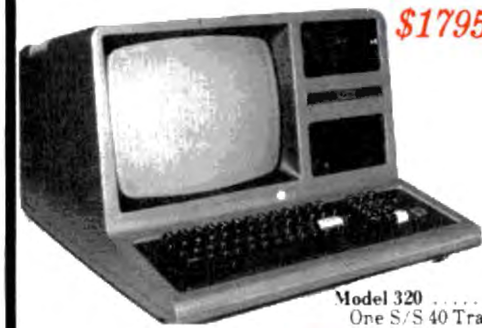
*a subsidiary of Worldwide Data Systems Inc.*

## SPECIAL

~27

### Dual Drive Model III

CUSTOM BUILT
*M321
*$1795.00*



**Model 320** ....... $1695.00
One S/S 40 Track Drive

* **Model 321** ....... ~~$1895.00~~
Two S/S 40 Track Drive

**Model 322** ....... $2295.00
Two D/S 40 Track Drives

**Model 324** ....... $2595.00
Two D/S 80 Track Drives

**Model 325** ....... $3995.00
One TM602 Winchester &
One S/S 40 Track Drive

**Model 326** ....... $4495.00
One TM603 Winchester &
One D/S 80 Track Drive

**FEATURES**

DOS PLUS 3.4 Operating System
Winchester Expansion Option
Gold Plated Edge Cards
Fully Socketed Drive Controller
Switching Power Supply
System Dust Cover
Expansion Capabilities
90 Day Warranty
or
1 Year Warranty Available

P-31 Green Phosphor CRT ........ $89.00

All customized Model III systems from Computex now include at *no extra charge* DOSPLUS 3.4. A Computex customized system starts out as a basic Model III, 32K RAM is added to increase the storage to 48K. Add to this our Model III drive controller with Winchester expansion option and the appropriate Tandon or MPI disk drives per your requirements. The system is then tested vigorously for 48 hours to assure you of peak performance when the system arrives at your doorstep.

## COMPLETE MODEL III DRIVE KITS


Complete Model III Disk Drive Kits include the Computex drive controller *with capabilities of Winchester drive expansion.* Virtually everything you will need to install disk drives in your Model III is included - We even include one of the most popular DOS systems available, DOS PLUS 3.4.

M3DK0 Drive Kit with 1 Single Sided 40 Track Drive
$495.00

M3DK1 Drive Kit with 2 Single Sided 40 Track Drives
$699.00

M3DK2 Drive Kit with 2 Double sided 40 Track Drives
$995.00

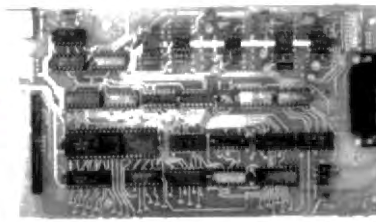
M3DK4 Drive Kit with 2 Double Sided 80 Track Drives
$1150.00

Model III Disk Drive Kit Less Drive's & DOS

**SPECIAL $249.95**

## MODEL III INTERNAL COMMUNICATIONS BOARD

*NEW!*



*The newest addition to the Computex line of Model III add on kits is our* **M3CB1** *Communications board complete with* **RS232 and DIRECT CONNECT MODEM.** *The kit can be supplied in either total kit form or as a partially assembled unit. The partially assembled unit only requires installation of the integrated circuits, all other components have been installed and tested. The* **M3CB1** *mounts in the same location as the Radio Shack RS232 board and installs easily in less than one hour.*

**FEATURES**

•300 to 19,200 Baud RS232•
•300 Baud Full Duplex Operation•
•Uses FSK Modulation Techniques•
•Supports Originate or Answer Modes•
•Will Operate DTE or DCE Equipment•

| | | |
|---|---|---|
| M3CB1-1 | P.C.B. & Manual | $34.95 |
| M3CB1-2 | Assembled & Tested RS232 | $69.95 |
| M3CB1-3 | Complete Kit | $129.95 |
| M3CB1-4 | Partially Assembled Kit | $169.95 |

*Our qualified technicians are available for technical questions at [713] 488-8022 every day from 4:00 to 5:00 PM CST*

## MPI & Tandon 5.25" DISK DRIVES

•Single or Double Density Operation•
•3 - 5 ms Track to Track Access Time•
•48 Hr. Burn in with Pre/Post Testing•
•Compatible with TRS80®, Zenith, IBM•

All disk drives from Computex are fully tested for Radial Head Alignment, Track 0 Switch, Speed, Instantaneous Speed Variation ISV, Azimuth, then verified for data storage integrity, reliability and systems compatibility.

| MPI | Tandon | Description | Price |
|---|---|---|---|
| B51 | TM100-1 | Single Sided 40 Track | $219.95 |
| B52 | TM100-2 | Double Sided 40 Track | $299.95 |
| B91 | TM100-3 | Single Sided 80 Track | $299.95 |
| B92 | TM100-4 | Double Sided 80 Track | $399.95 |

**Power Supply and Case** ........ $49.95

**SPECIAL $249.95**

40 Track Single Sided Disk Drive
Power Supply & Case

*INTRODUCING THE*

# PROMETHEUS 5

## 6.3M Byte External $1395.00

*WINCHESTER FOR THE MODEL III* *Complete*

That's Right! Now you can have over 5M Bytes of formatted data storage ( Options are available to increase from 5 to over 30 M Bytes) by simply plugging the Prometheus 5 into the I/O connector on your Model III.

Developed by Computex in conjunction with leading engineers in the Aerospace Industry, the Prometheus 5 offers quality and reliability using the latest advancements in technology. For example; The Winchester drive used in the system is from Tandon Magnetics, (the leading manufacturer of disk drives). The Winchester controller provides CRC generation / checking and error code correction circuitry, the fully regulated power supply has current and overvoltage protection as well as EMI and transient filtering on the input line.

The Prometheus 5 is operated by DOS PLUS 4.0, the leading Winchester Operating System from Micro Systems Software, and provides capabilities not yet thought of in most other DOS systems.

So--If you're considering mass storage for your Model III, Consider the Prometheus 5 for quality, reliability, and excellence in mass storage peripherals.

**Call or Write for brochure and picture describing the all new**

# PROMETHEUS 5

**AVAILABLE EXCLUSIVELY FROM COMPUTEX**

# SUPER SAVERS FROM COMPUTEX

### PRINTERS

| | |
|---|---|
| C Itoh 8510 Prowriter I Parallel | $549.00 |
| C Itoh 8510 Prowriter I Serial | 695.00 |
| C Itoh 1550 Prowriter II Parallel | 750.00 |
| C Itoh 1550 Prowriter II Serial | 850.00 |
| C Itoh F10-40cps Parallel | 1550.00 |
| C Itoh F10-40cps Serial | 1650.00 |
| C Itoh F10-40cps IEEE488 | 1650.00 |
| Smith Corona TP-1 | 650.00 |

### MODEMS 300 BAUD

| | |
|---|---|
| Novation CAT Acoustic 300 Baud Ans/Org | $140.00 |
| Novation DCAT Ans/Org D.C. | 175.00 |
| Novation Auto CAT Auto Ans/Org D.C. | 215.00 |
| UDS103 OALP Ans/Org D.C. | 175.00 |
| UDS103 JLP Auto Ans/Org D.C. | 209.00 |
| Lexicon LEX11 Acoustic Ans/Org | 135.00 |

### FURNITURE by DATADECKS

All furniture features mar and scratch resistant laminated tops, fully adjustable heights, easy assembly, moulded edges, with finishes available in either Oak or Putty.
Call or write for a free full color brochure that truly shows the beauty of the Data Deck line of computer furniture.

| | |
|---|---|
| Universal Printer Stand 27 * 16 | $115.00 |
| 36 * 30 Basic Work Station | 155.00 |
| 48 * 30 Basic Work Station | 178.00 |
| 60 * 30 Basic Work Station | 195.00 |
| Upper Shelf Assembly for 36 * 30 Desk | 85.00 |
| Upper Shelf Assembly for 48 * 30 Desk | 95.00 |
| Upper Shelf Assembly for 60 * 30 Desk | 109.00 |
| Disk Drive Module for 48 or 60 Desk | 66.00 |
| Lower Shelf Assembly for 36 Desk | 66.00 |

### PLOTTERS

| | |
|---|---|
| Watanabe WX4671 | $1195.00 |

### MEDIA * ACCESSORIES * BOOKS

| | |
|---|---|
| Verbatim 525-01 S.S. D.D. | $26.95 |
| Plastic Library Box 5† | 2.60 |
| TRS80 Disk & Other Mysteries | 19.95 |
| Microsoft Basic Decoded | 24.95 |
| The Custom TRS80 & Other Mysteries | 24.95 |
| Basic Faster & Better | 24.95 |
| Model I or Model III Joysticks | 39.95 |

### SOFTWARE

| | |
|---|---|
| DOS PLUS 3.4 | $119.00 |
| Starfighter A.I. | 24.95 |
| The Eliminator A.I. | 21.95 |
| Z Chess A.I. | 24.95 |
| Stones of Sisyphus A.I. | 24.95 |
| Escape from Traam A.I. | 17.95 |
| Planetoids A.I. | 17.95 |
| Sky Warrior A.I. | 17.95 |
| Missile Attack A.I. | 17.95 |
| Space Intruders A.I. | 17.95 |
| Showdown A.I. | 17.95 |
| Combat A.I. | 17.95 |
| Amored Patrol A.I. | 21.95 |
| Maxi Manager A.I. | 129.95 |
| Maxi Cras A.I. | 89.95 |

### CABLES

| | |
|---|---|
| C Itoh - Model III Printer | 29.95 |
| Drive Extender | 12.95 |
| 2 Drive Cable | 29.95 |
| 4 Drive Cable | 39.95 |

# INTERFACE EXPANSION KITS FOR TRS80 COMPUTERS

The MDX Series Interface Expansion Boards provide a wide range of features for the TRS80® Computers. Bare boards or complete kits are available for the MDX-2 (Model I Expansion I/F and the MDX-3 Model III Expansion I/F) which will allow you to upgrade your system by assembling only those sections or features that you need when you need them. P.C. Boards are silkscreened and soldermasked for ease in assembly and the assembly / user's manual supplied with each kit includes theoretical operation as well as step by step assembly and installation instructions.

| | |
|---|---|
| **MDX-2 Model I Expansion I/F Kit** | **$269.95** |
| RS232 Serial I/F Kit | 24.95 |
| RAM Expansion Kit (less RAM) | 17.95 |
| Disk Controller Kit | 31.95 |
| Hardware & Socket Kit | 29.95 |
| Power Supply Kit | 29.95 |

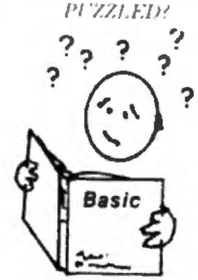
| | |
|---|---|
| **MDX-3 Model III Expansion I/F** | **$259.95** |
| Disk Controller Kit | 89.95 |
| RS232 Serial I/F Kit | 79.95 |
| Phone Modem Kit | 39.95 |

Additional Kits Available - Call For Quotes

If you're considering a IBM, Apple III, or Model 16 computer for your business needs, consider the NorthStar ADVANTAGE - We think you'll be surprised! Why? Simple, The NorthStar ADVANTAGE offers far more capability than any of the aforementioned systems. For example; The NorthStar ADVANTAGE features high resolution graphics, Full CP/M support, 64K RAM (upgradeable), 720K Bytes disk storage, tried and proven software, and the capability of growing with your needs. Optional interfaces are available that will allow you to run 8 or 16 bit software or upgrade to a multiuser system by simple adding NorthNet. Perhaps the most incredible part is the price - $3595.00 (less than you'll pay to upgrade a Model III to the same capabilities).

So if your tired of playing games, or you need a serious system for your business, contact Fred Haskin at (713)488-4215 for further information and a demonstration of the NorthStar ADVANTAGE

PUZZLED?




SEE OUR AD ON PAGE 383

COMPUTEX

*a subsidiary of World Wide Data Systems Inc.*

17321 El Dorado Blvd.
Webster, Texas 77598

(713)488-8022

✓27

CALL OR WRITE FOR NEW CATALOG

**Here's How To Order**

*Phone orders welcomed with VISA, MC, or American Express. Net 30 Terms available to well rated D&B firms, Universities, and Government Organizations. Sorry, COD orders are limited to $100.00. Prices quoted do not include shipping, are available mail order only, and are valid during the month of publication. Prices are subject to change wihtout notice. TRS80, Model I, & Model III are registered trademarks of Radio Shack a Tandy Corporation.*

*Learn position and chain sorts.*

# Another Sort of Sort

*Karl L. Townsend*
*103 Knollwood Drive*
*Lansdale, PA 19446*

Sorting plays a large role in programming data base files such as mailing or telephone lists. With few exceptions most data base applications require data sorting according to some prearranged plan.

| | | |
|---|---|---|
| 1 | Joe | 6 |
| 2 | Sam | 9 |
| 3 | Andy | 2 |
| 4 | Ted | 10 |
| 5 | Mac | 8 |
| 6 | Bill | 3 |
| 7 | Al | 1 |
| 8 | Fred | 5 |
| 9 | Lou | 7 |
| 10 | Ed | 4 |

*Fig. 1*

Most of us started using the bubble sort because it is so simple and quick to program. As we learned more about programming and realized the bubble sort's lack of speed and its inefficiency, we moved on to other methods such as the Shell, and the Heap Sort.

Most sorts move data from location to location as the sort progresses. This very movement of data carried to an extreme makes the bubble sort so poor.

If you do much relocating or exchanging string data, the TRS-80 is likely to go into "mumble mode" while clearing its head. The TRS-80 locks up everything while it rearranges the string space and rids itself of unnecessary strings. You can reduce these effects by clearing additional string space or by using the VARPTR instruction for your exchanges. But why move the data at all?

```
10 REM   POSSORT/V01
100 FOR I = 1 TO 10
110 READ A$(I)   'READS IN TEST NAMES FROM DATA LIST
120 NEXT I
200 FOR I = 1 TO 10
210 C(I) = 1   'PRESETS POSITION COUNT ARRAY TO ALL ONES
220 NEXT I
230 FOR I = 10 TO 2 STEP -1  'START SORT
240 FOR J = I-1 TO 1 STEP -1
250 IF A$(I) => A$(J) THEN C(I) = C(I) + 1 ELSE C(J) = C(J) + 1
260 NEXT J
270 NEXT I   'END SORT
900 FOR I = 1 TO 10
910 LPRINT I, A$(I), C(I)
920 NEXT I
930 LPRINT
1000 DATA JOE, SAM, ANDY, TED, MAC, BILL, AL, FRED, LOU, ED
```

*Program Listing 1*

How about a method of sorting without moving data out of its assigned storage location? This solves the string clearing problem, and also avoids excessive writing and rewriting of data files. Position sorting and chaining may be for you.

### Position Sorting

To start, we will experiment with a few routines to look at a list of data elements and record their relative positions in a file, in this case, an array. Program Listing 1 shows a short program to do just this. The program consists of three parts. The first section, lines 100–120, reads names to be sorted into the A$ array.

Since we are not going to move the original data, we must keep track of our sort. Array C(x) accepts and stores the numerical position of records once they are sorted. This array, the same size as our data array, has a storage position to match each data item in the main storage array. Lines 200–220 preset this position array to all ones, preparing it for the sort.

| | |
|---|---|
| 1 | Al |
| 2 | Andy |
| 3 | Bill |
| 4 | Ed |
| 5 | Fred |
| 6 | Joe |
| 7 | Lou |
| 8 | Mac |
| 9 | Sam |
| 10 | Ted |

*Fig. 2*

Lines 230–270 contain the sort. The test sort is set up to arrange data in ascending order. As the sort progresses, each data item is compared against each of the others. Size, as used here, means relative alphabetic position. For example, B is larger than A, and Fred is larger than Andy. If a given data item is larger than the one against which it is being compared, the sort adds a one to its equivalent position in the C array. This process is called incrementing. Larger items are incremented in the C array more often than smaller items. The largest data item is incremented every time it is compared; the smallest, never incremented, remains a one.

Lines 900–930 form a simple print routine enabling us to see the results of our labors. If you have no printer, change all LPRINT instructions to Print to show output on the screen.

Type in the program in Listing 1. Figure 1 shows the result of our sort. Column one lists the consecutive positions of the data in the arrays. Column two lists data as read from the data statement. But take a look at column three. These numbers in sequence from 1 through 10 indicate the sorted position of their associated data elements. If we read through the array based on the sequence shown in column three, we are reading the data in ascending order. We have kept our promise; the original data is unchanged in position or order.

We can now print the data in sorted order. Add the following lines to your original program.

```
950 FOR I = 1 TO 10 'PRINT DATA IN
    ASCENDING ORDER
960 FOR J = 1 TO 10
970 IF C(J) = I THEN LPRINT I, A$(I)
980 NEXT J
990 NEXT I
```

As the program runs, you will again see the output in Fig. 1. However, immediately following it is the list of names printed in ascending order (see Fig. 2).

This print routine is a rather poor one. Although the large amount of time is not obvious with this small file, the procedure is very slow on a larger one. If you analyze the nested loops, you will see that the entire list of 10 items is read each time one item is printed. The computer searches for a match in the C array for each print. This can be speeded by jumping out of the inner loop as soon as a print is completed, but on the average, the computer reads half of the entries for each print. To print a file of 10,000 data items would require about 500,000 reads. We need a better way to print or a different sort.

### Chaining

Chaining lets us jump from one record to the next higher one as we print without extensive searching. We only need to read each record once to print it. Add the following statements to your program:

```
600 X = 0
610 FOR I = 10 TO 1 STEP-1
620 FOR J = 10 TO 1 STEP-1
630 IF C(J) = I THEN X(J) = X: X = J
640 NEXT J
650 NEXT I
```

Also, change line 910 to:

```
910 LPRINT I, A$(I), C(I), X(I)
```

Delete lines 950–990. Now run.

Figure 3 shows the result. The first three columns remain the same; a fourth column depicts the X array containing the chaining information. The numeral 3 beside Al's name says that the next record (Andy) is found in array position 3. At this location, column four says to go to position six where we find Bill. If you trace each link of the chain, you will arrive at Ted. A zero in the chaining column indicates the end of the file.

It appears that we can now step through the file from record to record in ascending order without any exchanges. We still have a problem, however: How do we get to Al's record to begin the sequence? Since it is unlikely that the first physical record is the first in the sort order, let's insert a dummy record which will always be first. In this case, use the name AAA since no normal English word will sort before it.

Change Joe in line 1000 to AAA, leaving the rest of the names as they are. Now run again.

Figure 4 shows the result. AAA is the first record; its chaining number (seven) leads right to Al. Trace the chaining numbers and you will step through the entire file, ending with Ted and a zero to indicate end of file.

To print the names in sequence from this chaining information add the following lines to your program:

```
950 T = X(1)
960 FOR I = 1 TO 9
970 LPRINT I, A$(T)
980 T = X(T)
990 NEXT I
```

On running again, you will see the same data, followed by the file printed in ascending order.

Let's review the additions we just made. Line 950 takes the chaining information from the dummy record and saves it in variable T. Line 960 begins the print loop. Note that it only goes from 1 to 9 as we do not want to print the dummy record. Line 970 is the print line. On the first time through the loop it prints A$(T), the first record as directed by the chaining information. The key to the chained print, however, is in line 980; T is set equal to the next link in the chaining column. This process continues until all records are printed in order.

### Bubble Sort or Chain Sort?

I decried the bubble sort as slow and inefficient. It appears at this point that the chain sort procedure is even worse. More programming is required to support the sort, and data is more difficult to print.

Have we really accomplished anything?

Are there any advantages to the chain sorting procedure? Figure 5 shows information on 10 people. Suppose we want to print three different listings of this data: name order, city order, and state order. Normally this requires sorting into each order wanted and then printing. If later you need a print in another order, you must sort again. How can we apply the techniques just learned to this type of situation?

First, we generate a dummy record to precede the file. We insert the record AAA before the first entry in each column. Enter the program shown in Program Listing 2.

To make things easier when printing our file, the program provides a multidimensioned array to contain the data. Line 100 dimensions A$ to 20 rows and 3 columns. I allowed for a few extra lines in the array in case you

| | | | |
|---|---|---|---|
| 1 | Joe | 6 | 9 |
| 2 | Sam | 9 | 4 |
| 3 | Andy | 2 | 6 |
| 4 | Ted | 10 | 0 |
| 5 | Mac | 8 | 2 |
| 6 | Bill | 3 | 10 |
| 7 | Al | 1 | 3 |
| 8 | Fred | 5 | 1 |
| 9 | Lou | 7 | 5 |
| 10 | Ed | 4 | 8 |

*Fig. 3*

```
10 REM  POSSORT/V06
100 DIM A$(20,3), C(20), X(20,3)
110 FOR I = 1 TO 11
120 FOR J = 1 TO 3
130 READ A$(I,J)  'READ IN TEST DATA
140 NEXT J
150 NEXT I
200 FOR I = 1 TO 3  'SORT COLUMNS IN ORDER
210 FOR J = 1 TO 11  'PRESET C ARRAY TO ALL ONES
220 C(J) = 1
230 NEXT J
240 FOR K = 11 TO 2 STEP -1   ' START POSITION SORT
250 FOR L = K-1 TO 1 STEP -1
260 IF A$(K,I) => A$(L,I) THEN C(K) = C(K)+1 ELSE C(L) = C(L)+1
270 NEXT L
280 NEXT K
290 X = 0
300 FOR K = 11 TO 1  STEP -1   'CHAIN COLUMN
310 FOR L = 11 TO 1  STEP -1
320 IF C(L) = K THEN X(L,I) = X: X = L: L = 1
330 NEXT L
340 NEXT K
350 NEXT I
400 INPUT"TO PRINT BY NAME -ENTER 1, BY CITY -ENTER 2, BY STATE -ENTER 3";P
R
410 T = X(1,PR)
420 FOR I = 1 TO 10
430 LPRINT A$(T,1), A$(T,2), A$(T,3)  'PRINT ROUTINE
440 T = X(T,PR)
450 NEXT I
460 GOTO 400
1000 DATA AAA,AAA,AAA,JONES,MORRISVILLE,PA,BROWN,PHILADELPHIA,PA,SMITH,CAMD
EN,NJ,GREEN,TRENTON,NJ,OTIS,PHILADELPHIA,PA
1010 DATA TALL,LANSDALE,PA,BROWN,PHILADELPHIA,PA,ARNESS,BUFFALO,NY,JAMES,TR
ENTON,NJ,SAMPSON,DOVER,DEL
```

*Program Listing 2*

want to include additional test data. Since we will sort on all three columns, array X is also dimensioned to 20 by 3. Remember, the chaining array must provide a position to match each data key on which a sort is made.

Lines 110–150 read the test data into the A$ array using nested loops. Each time the I loop is incremented by one, the J loop reads in three data items: name, city and state.

Line 200 starts the sort. The program steps through each of the three columns in order. When I equals one, the names are sorted; on two, the city; and on three, the state.

Lines 210–230 preset the position array to all ones just before each column is sorted.

The position sort itself is in lines 240–280. We are now working with two-dimensional arrays. The first time through the sort, the I in each array is a one; all sort comparisons are based on the first column of data (names). Subsequent sorts change this to a two and then a three to sort by city and state.

Lines 290–340 chain the data as before except that the chaining information is stored in the multidimensional array X. The I loop controls chain storage to correspond with the data column being sorted.

In this area, I added another small change at the end of line 320. L equals one sets the loop counter to the loop end count when a find is made. When the Next L statement is reached, the program jumps out of the loop. This avoids cycling through the loop after a find has already been made and the chain number has been stored in the X array. On a large file, this is a significant difference; it cuts the average time in this loop by about one-half.

Line 400, a mini-menu, allows you to select the column you want sorted. The print routine is as used before except that you can select the chaining column for the particular print requirements.

The program will hesitate while reading in the data and performing the sorts. When the prompt line for printing appears, try each option. You can print, in order, any of the three columns without re-sorting the data.

This ability to print in various sequences without re-sorting is an asset in large files. In the multiple sort situation chained files really shine.

To save your sort information, just save the chaining information right along with your data. This enables you to print data in your selected sequence next time you use the file. Use the following routine:

```
800 FOR I = 1 TO 11
810 PRINT # - 1, X(I,1), X(I,2), X(I,3),
        A$(I,1), A$(I,2), A$(I,3)
820 NEXT I
```

I have developed some of the concepts of chaining and data sorting without rearranging a file. The somewhat crude and unsophisticated examples should demonstrate possible applications for your programming efforts. ■

*Karl Townsend works on system interfaces on the Navy Aegis program for RCA.*

| | | | |
|---|---|---|---|
| 1 | AAA | 1 | 7 |
| 2 | Sam | 9 | 4 |
| 3 | Andy | 3 | 6 |
| 4 | Ted | 10 | 0 |
| 5 | Mac | 8 | 2 |
| 6 | Bill | 4 | 10 |
| 7 | Al | 2 | 3 |
| 8 | Fred | 6 | 9 |
| 9 | Lou | 7 | 5 |
| 10 | Ed | 5 | 8 |

*Fig. 4*

| | | | |
|---|---|---|---|
| 1 | Jones | Morrisville | PA |
| 2 | Brown | Philadelphia | PA |
| 3 | Smith | Camden | NJ |
| 4 | Green | Trenton | NJ |
| 5 | Otis | Philadelphia | PA |
| 6 | Tall | Lansdale | PA |
| 7 | Brown | Philadelphia | PA |
| 8 | Arness | Buffalo | NY |
| 9 | James | Trenton | NJ |
| 10 | Sampson | Dover | DE |

*Fig. 5*

✓ See List of Advertisers on Page 451

LOAD 80

**The Key Box**

**Level II Basic**
**Model III**

*Modify Uni-Key for your Mod III.*

# Autokey

*Don Rigg*
*17 Bittersweet Bay*
*Winnipeg, Manitoba*
*Canada*

A member of my TRS-80 user's group recently showed me a most remarkable program. A Basic keyword appears whenever he presses the shift key in combination with any other key. The program is called Uni-Key, by Rowand Archer, Jr. of Hillsborough, NC. It appeared in the September 1980 issue of *80 Microcomputing*.

The program should run on my Model III, but it most inconveniently uses a lowercase letter for generating the keywords. This works well on the Model I, as a shifted character is indeed lowercase. In my case I was usually generating lowercase letters and the Shift produced uppercase. I would have to do all my typing holding down the shift key (except when I wanted a keyword) or keep using the shift 0 uppercase only toggle. Neither solution was very satisfactory, so I customized Uni-Key for the Model III and included a few other worthwhile features.

My version of Uni-Key gives you an alternate keyboard layout which generates the keywords when you press the shift key. You can also turn the special keyboard off to revert to normal upper and lowercase typing. When in keyword mode all the words shown in Fig. 1 are available at the touch of only one letter key. (You can substitute Assembly language or Pascal keywords for the Basic keywords.)

I designed Fig. 1 so you can copy it onto a file reference card and keep it at your keyboard. The dashed lines associate keywords with letter keys.

### Design Specifications

My Autokey program has the following features: It retains the original Uni-Key concept of using the shift key to generate keywords; it uses additional control keys on the Model III by pushing shift, down arrow and a letter key. You can turn the routine off to return to the regular keyboard. A routine automatically sets the memory size, another executes the program as soon as it reads in the tape, and a third starts the computer with the all caps switch off rather than on as is usual.

The program changes lowercase characters to uppercase; its logic assigns keywords whenever you press an uppercase or control character.

My program includes a routine which you can execute via System. This routine resets all Basic control pointers, allowing you to re-access a Basic program if you inadvertently type New or CLOAD when you mean CSAVE.

Two keys to understanding this program are the functions of the keyboard device control block, and ASCII code.

### Device Control Block

The computer often executes a ROM routine which looks at the keyboard to see if you have pressed a key. That is why the computer reacts so quickly when you do. At one point in this routine it looks at an address pointer in RAM for where to go next. This provides you with an opportunity to tell the computer where to go (see Fig. 2). By changing this address you will successfully intercept part of the computer's operating system (its brain); you can easily fool it into changing things around.

As you intercept each character on its way from the keyboard to the Basic interpreter, you can change some of them to suit your new purposes.

### ASCII Coding Structure

The computer stores alphanumeric information in its memory in ASCII (American National Standard Code for Information Interchange).

In a TRS-80, the keyboard scan routine looks at a collection of wires between memory addresses and the physical keyboard. Voltages change when a key has been pressed; the changes are then translated into the proper ASCII value and put into the A register.

### Changing Characters

Look at the code outlined in the ASCII tables in the Model III reference manual (Appendix C in mine) and notice that all control characters (shift, down arrow, character) are grouped from code numbers 01H to 1FH. Similarly, all uppercase characters are grouped from 41H to 5AH, while the lowercase characters run from 61H to 7AH. Notice that lowercase characters are coded exactly 20H higher than the corresponding uppercase letters. This symmetry proves very valuable. For example, you can change a lowercase b (62H) to uppercase by subtracting 20H.

My program leaves the control characters alone and looks up keywords numbered 01H to 1FH (the first 31). It subtracts 21H from the codes for an uppercase letter; they continue the number sequence immediately after the control codes. Capital A with a code of 41H becomes 41H – 21H, or 20H (next in the sequence above 1FH). The program changes lowercase characters to uppercase by subtracting 20H whenever you input one from the keyboard. All other codes remain unchanged.

### Assigning Keywords

I developed Fig. 1 when I assigned keywords to keys. I grouped the keywords near the key of the same letter so you can find them easily. You can easily change the sequence or the words to whatever you want.

After I developed the keyboard layout, I put words into sequence with ASCII codes so I could put them in the right order in the program. The numeric listing is shown in Fig. 3. You will find this useful if you want to change the words. I reserved several control codes because some control functions per-

> *"A Basic keyword appears whenever he presses the shift key in combination with any other key."*

form their usual functions (Enter, backspace, and so on). Here also, you can bypass codes if you like.

**Program Structure**

After I established the basic structure for the keywords I prepared a flowchart (see Fig. 4) to give a broad overview of the main program functions.

To use the flowchart start at the top left and follow the arrows based on the answers you get at a diamond-shaped box.

There are four separate modules. Module 1 has an ORG of 41E2H, in System RAM. It changes an address pointer referred to (called) by a ROM routine just after it has loaded a System tape. It gives the program the Auto Start feature.

Module 2 is used just once to set the pointers and flags and to change the address in the Keyboard Device Control Block (DCB) so the new routine will be used. Completely automatic, it resets the address pointer changed in Module 1 so the computer operates normally if you load another System tape.

Module 3, the main part of the program, determines which key you pressed and whether to take any special action.

Module 4 is the stand-alone routine to restart a Basic program.

**The Program**

Program Listing 1 contains the Assembly language version of the program. It is set up to run in a 16K cassette-based system but, by changing the ORG at lines 320 and 2860 and the memory size values in lines 410 and 430, you can run the program in any size memory.

Program Listing 2 is a Basic language routine which POKEs the same program into memory and executes it.

The program is written in the same sequence as the flowchart in Fig. 4. Lines 410-450 limit the amount of memory Basic will use so it does not destroy this program. (If you want to change this, set the value in line 410 to the highest address you want Basic to use minus 2. Set line 430 to the

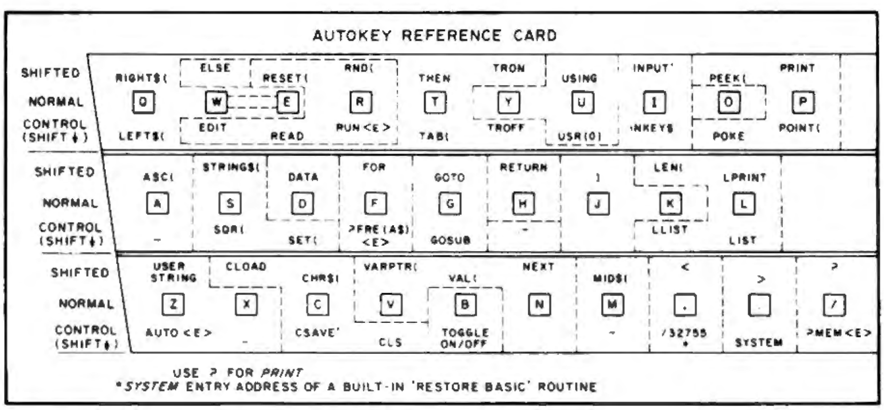



Figure 1

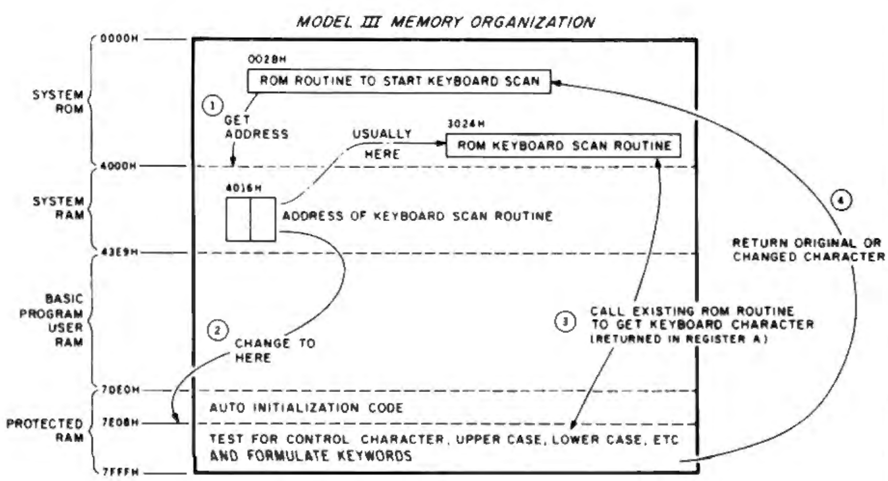



Figure 2



See List of Advertisers on Page 451

value in line 410 minus the amount of string space you want to set aside. It is currently set at 32H to give the usual 50 decimal bytes for string storage. Clearxxx in Basic will also work as usual.)

Lines 790–870 test for the control B toggle, which returns normal keyboard operation until you type control B again.

Lines 990–1300 determine whether the key pressed (as returned in register A) is a control, lowercase or uppercase character. I found this particularly interesting as I had not used the carry flag feature of CP (Compare) before. A CP is really a subtract which does not change the registers: If the A register is smaller than the number compared to it, a borrow sets the carry flag on. Then Z means register A was equal to the value compared to; NZ means register A was not equal to the value; C means the value in A was less than the value compared to; and NC means the value in A was equal or greater than that compared to.

Lines 1340–1630, fundamentally the same as Uni-Key, skip through the list of keywords to get to the right one. They count the number of 00 bytes from the start of the words table in line 1730. It is very important that each word (or phrase) be followed by a 00. Several commands include an Enter so one keystroke also executes the command. Examples of these are Run, Auto, and so on. They are shown on the reference chart (Fig. 1) with the symbol <E>. The control J and N keys do not have anything assigned and are available for your additions. If you add words, make sure the end of the last letter in line 2820 does not flow over the start of the Restart routine in line 2870, as only 10 bytes are left. If you need more room, change the ORG in line 320 and the memory protect values.

**Using the Program**

If you use the Basic version just CLOAD it right after you turn the computer on and run it immediately. Enter NEW to remove the program and it will work as described below. Watch the auto memory size setting in line 110: If you need a different value, set the memory size when you turn on the computer and delete line 110 or change the values in line 110 before running the program.

If you use the Assembly language version just start your object code loading by Entering System and the name you gave it (mine is KEY) and the program will do the rest. It returns you to the Basic Ready prompt, with Autokey waiting to ease your program typing.

Each key functions as indicated in Fig. 1. For example, if you push the G key you will get a capital G. If you push shift G the word GOTO appears. If you push the shift key, the down arrow key and the G key the word GOSUB will appear. All other keys work

| Key pressed shift down arrow | ASCII code returned in A register | Changed purpose | Key Pressed Shift plus | ASCII code returned in A register | Changed purpose | Key pressed | ASCII code returned in A register | Changed purpose |
|---|---|---|---|---|---|---|---|---|
| A | Ø1 | Break | A | 41 | ASC( | a | 61 | A |
| B | Ø2 | Toggle | B | 42 | VAL( | b | 62 | B |
| C | Ø3 | CSAVE" | C | 43 | CHR$( | c | 63 | C |
| D | Ø4 | Set | D | 44 | Data | d | 64 | D |
| E | Ø5 | Read | E | 45 | Reset | e | 65 | E |
| F | Ø6 | FRE(A$) | F | 46 | For | f | 66 | F |
| G | Ø7 | GOSUB | G | 47 | GOTO | g | 67 | G |
| H | Ø8 | Reser | H | 48 | Return | h | 68 | H |
| I | Ø9 | INKEY$ | I | 49 | Input | i | 69 | I |
| J | ØA | avail | J | 4A | ) | j | 6A | J |
| K | ØB | LLIST | K | 4B | LEN( | k | 6B | K |
| L | ØC | List | L | 4C | LPRINT | l | 6C | L |
| M | ØD | Reser | M | 4D | MID$( | m | 6D | M |
| N | ØE | avail | N | 4E | Next | n | 6E | N |
| O | ØF | POKE | O | 4F | PEEK( | o | 6F | O |
| P | 1Ø | Point | P | 5Ø | PRINT @ | p | 7Ø | P |
| Q | 11 | LEFT$( | Q | 51 | RIGHT$ | q | 71 | Q |
| R | 12 | Run | R | 52 | RND( | r | 72 | R |
| S | 13 | SQR( | S | 53 | STRING$ | s | 73 | S |
| T | 14 | TAB( | T | 54 | Then | t | 74 | T |
| U | 15 | USR(Ø) | U | 55 | Using | u | 75 | U |
| V | 16 | CLS | V | 56 | VARPTR( | v | 76 | V |
| W | 17 | Edit | W | 57 | Else | w | 77 | W |
| X | 18 | Reser | X | 58 | CLOAD | x | 78 | X |
| Y | 19 | TROFF | Y | 59 | TRON | y | 79 | Y |
| Z | 1A | Auto | Z | 5A | User String | z | 7A | Z |
| ↑ | 1B | Reser | | Subtract 21H from each to fit after | | | Subtract 2ØH from each to change to uppercase | |
| , | 1C | /32755 | | | | | | |
| 9 | 1D | avail | | | | | | |
| . | 1E | System | | | | | | |
| / | 1F | ? MEM | | | | | | |

Figure 3

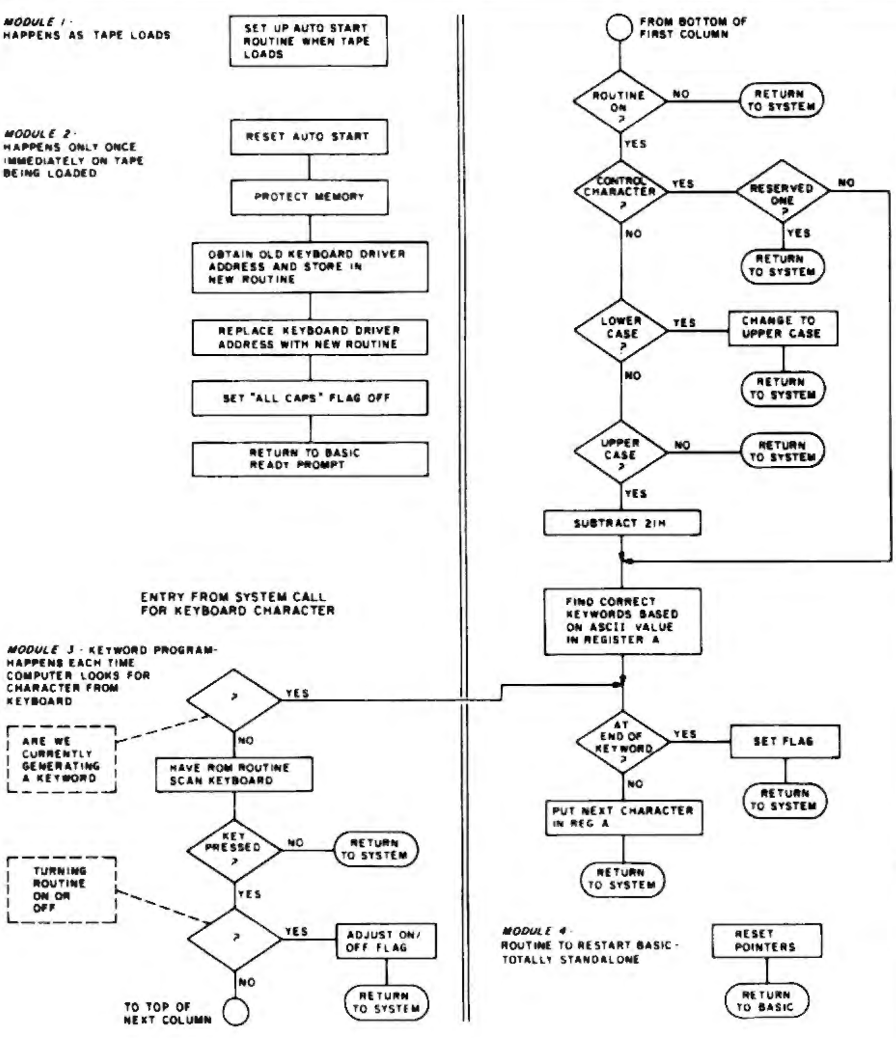



Figure 4



✓See List of Advertisers on Page 451

# For TRS-80 THE FANTASTIC FOUR

FROM Adventure INTERNATIONAL

## STARFIGHTER

**BY SPARKY STARKS**

*Creative Computing says, "The game is great — nothing less than excellent."*

**STARFIGHTER** is the finest space battle simulation ever written for the TRS-80, and that's no idle boast. 80-U.S. Journal says that **STARFIGHTER** "...is like no other game ever seen." BYTE magazine reports that the game "...offers excitement and excellent use of TRS-80 graphics and sound." **STARFIGHTER's** Arcade Action Graphics(tm) and sounds are enjoyably combined to produce truly mind-challenging game situations.

The **STARFIGHTER** package includes two cassettes (with separate Main Mission and New Pilot Training Simulator tapes), or one self-loading disk. Also included is an extremely detailed 32-page instruction manual.

16K Tape Model 1 and 3
010-0120 $24.95
32K Disk Model 1 and 3
012-0120 $29.95

BRAND NEW!

## LASER BALL

**BY NEIL MATSON AND MATTHEW McMAHON**

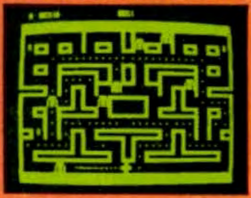
*"This is the one I've been waiting for ... an incredible game in every sense of the word!" Will Hathaway, Arcader*

*"Superb graphics, great sounds, and, best of all, the game really taxes your playing skills ..." Pat Henderson, Arcader*

Stand back, arcaders, it's **LASER BALL** — a white-hot sound and graphics extravaganza that'll practically fuse the circuits in your TRS-80! Your **LASER BALL** must race through a twisting maze avoiding or attacking killer disruptoids, depending on your energy status. Only the fast will survive — can you?

**LASER BALL** doesn't miss a trick with its crisp Arcade Action Graphics(tm), high score save, 24 separate playing levels, joystick compatibility and more. Brand new from Adventure International!

16K Tape Model 1 and 3
Now Available $24.95
32K Disk Model 1 and 3
Now Available $24.95



## LUNAR LANDER

**BY MIKE WALL AND JACK MONCRIEF**

*"Certainly one of the most realistic and imaginative simulations for the TRS-80..." 80-U.S. Journal*

The tension is nearly unbearable as your Lunar Excursion Module plummets towards a craggy lunar surface that's rife with jagged peaks and impossibly deep fissures. Only a firm hand on the controls and a steady eye on the monitor screen can spell the difference between a successful landing and an explosive death.

Your Lunar Lander comes fully equipped with everything that you'll need to challenge this dangerous landing site, including sound, running high score, two player option and breathtaking Arcade Action Graphics(tm).

16K Tape Model 1 and 3
010-0094 $24.95
32K Disk Model 1 and 3
012-0094 $24.95

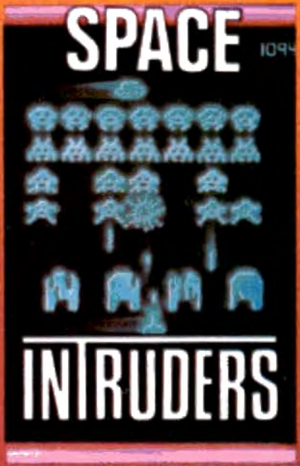
## SPACE INTRUDERS

**BY DOUG KENNEDY**

*"The program is fantastic ... the graphics are excellent." 80-U.S. Journal*

Those crafty Invaders are back with a vengeance and Adventure International has 'em! **SPACE INTRUDERS** has been praised by major microcomputing publications from 80-U.S. to Creative Computing as "fantastic," "excellent," "top of the line," and, well, you get the idea! **SPACE INTRUDERS** faithfully reproduces the hit Deluxe arcade machine in all of its frustratingly enjoyable glory. An absolute must for any serious game collection!

**SPACE INTRUDERS** comes ready for battle with two player option, sounds, joystick compatibility and Arcade Action Graphics(tm) that'll knock your socks off.




16K Tape Model 1 and 3
010-0116 $24.95
32K Disk Model 1 and 3
012-0116 $24.95

ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
(ORDERS ONLY PLEASE)
SHIPPING & HANDLING ARE EXTRA
PRICES SUBJECT TO CHANGE WITHOUT NOTICE

22

Adventure INTERNATIONAL

BOX 3435, LONGWOOD, FL 32750
(305) 830-8194 (QUESTIONS)
Send for our free catalog of 180 programs

*"This 13-byte routine has saved me hundreds of hours . . ."*

similarly.

To return to the normal keyboard push shift, down arrow, and B. You can now type lower and uppercase letters or use the standard shift 0 feature to lock in all caps. Pushing shift, down arrow, B again will restore the keyword mode. If you used shift 0 you must toggle it off.

**User Definable String**

You can program an alphanumeric string of your choice (up to 68 characters) at any time and insert it into your program.

To use this feature type in a line 0 jumping to the first line in your program (for example, 0 GOTO 10). Type in a line 1 setting a variable named Z$ equal to your desired string (for example, 1 Z$ = "This Is My Desired String"). Push shift Z; a preprogrammed line 2 will appear. Enter RUN1. From now on, whenever you push shift Z, your selected string will appear in your program.

If you leave this little routine in place until you are finished you can change your string at any time by putting a new string in line 1 Z$ = ". . ." and typing RUN1. Before saving the final version of your program, type Delete 0–2 and you will not have these as a permanent appendage. (Of course, you cannot use lines 0–2 or Z$ in your own programs.)

**Restore Basic Pointers Routine**

This 13-byte routine has saved me hundreds of hours retyping because I typed CLOAD when I meant CSAVE, or when new USR routines locked up the computer.

When you issue the CLOAD command the computer first resets all the Basic pointers, ready to put the new program in the right memory locations. But unless you actually read in some new code, the old pro-

*Program Listing 1*

```
              ;*****************************************************
              ;*                                                   *
              ;*                 A U T O  -  K E Y                 *
              ;*                   JULY    1981                    *
              ;*                                                   *
              ;*          A Program to Produce BASIC KEYWORDS      *
              ;*          With a Single Keystroke-TRS80-MODIII     *
              ;*                                                   *
              ;*                     Don Rigg                      *
              ;*                17 Bittersweet Bay                 *
              ;*            Winnipeg, Manitoba, Canada             *
              ;*               Tel (204)253-9230                   *
              ;*                                                   *
              ;*****************************************************
              ;
              ; ***   Set Up Auto Start    ***
              ;
41E2                  ORG     41E2H    ;SYSTEM Jump Address
41E2 C3E07D           JP      START    ;AUTOKEY Initialization Rtn.
              ;
              ; ***   Initialization Routines    ***
              ;
7DE0                  ORG     7DE0H
              ;
              ;       RESET AUTO START
              ;
7DE0 3EC9     START   LD      A,0C9H          ;RET Code
7DE2 32E241           LD      (41E2H),A       ;To SYSTEM RAM
              ;
              ;       SET MEMORY PROTECT
              ;
7DE5 21DE7D           LD      HL,7DDEH        ;7DE0H-2
7DE8 22B140           LD      (40B1H),HL      ;Put in 'END OF BASIC' Ptr
7DEB 21AC7D           LD      HL,7DACH        ;Mem Size-32H (50D)
7DEE 22A040           LD      (40A0H),HL      ;Put in 'START OF STRING SPA
CE' Ptr
7DF1 CD721B           CALL    1B72H           ;Reset ALL Pointers
              ;
              ;       REPLACE KEYBOARD DRIVER ADDRESS
              ;
7DF4 2A1640           LD      HL,(4016H)      ;Obtain OLD KBD Driver Addre
ss
7DF7 220F7E           LD      (GETCHR+1),HL   ;Put in CALL Statement
7DFA 21087E           LD      HL,NEWDVR       ;Put New Driver
7DFD 221640           LD      (4016H),HL      ;Address in KBD DCB
              ;
              ;       SET ALL CAPS OFF
              ;
7E00 3E00             LD      A,0H            ;0 = Off
7E02 321940           LD      (4019H),A       ;All CAPS Flag
              ;
              ;       RETURN TO BASIC READY PROMPT
              ;
7E05 C3191A           JP      1A19H
              ;
              ; ***   Main AUTOKEY Program    ****
              ;
              ;       ENTRY FROM ROM CALL TO SCAN KEYBOARD
```

*Listing 1 continues*

```
                 00660 ;
7E08 3A7E7E      00670 NEWDVR  LD    A,(SUBFLG)     ;If flag >< 0 in middle
7E0B B7          00680         OR    A              ;of a KEYWORD
7E0C 205E        00690         JR    NZ,CONTI       ;Substitution-CONTINUE
                 00700 ;
                 00710 ;       Get Character
                 00720 ;
7E0E CD0000      00730 GETCHR  CALL  $-$    ;Original Keyboard Scan Rtn.
7E11 B7          00740         OR    A      ;Character Returned in A
7E12 C8          00750         RET   Z      ;No Key Pressed-Process Ordinarily
                 00760 ;
                 00770 ;       TEST FOR SWITCH CHANGE
                 00780 ;
7E13 F5          00790         PUSH  AF     ;Save Kbd Char
7E14 FE02        00800         CP    02H    ;CNTRL B Toggle
7E16 200C        00810         JR    NZ,SWTST  ; Don't Change
7E18 3A7D7E      00820         LD    A,(ONOFF) ;Get Status Switch
7E1B C680        00830         ADD   A,80H  ;Results in 00 or 80
7E1D 327D7E      00840         LD    (ONOFF),A ;Update
7E20 F1          00850         POP   AF     ;Restore Original Character
7E21 3E00        00860         LD    A,0H   ;Change to NULL=No Key Pressed
7E23 C9          00870         RET          ;Back to ROM Routine
                 00880 ;
                 00890 ;       TEST IF RTN ON or OFF
                 00900 ;
7E24 3A7D7E      00910 SWTST   LD    A,(ONOFF) ;Get Status
7E27 FE00        00920         CP    0H     ;Rtn ON
7E29 2802        00930         JR    Z,CTRTST;Continue This Routine
7E2B F1          00940         POP   AF     ;Get Original Character
7E2C C9          00950         RET          ;Routine OFF-Process Ordinarily
                 00960 ;
                 00970 ;       TEST FOR CONTROL CHAR
                 00980 ;
7E2D F1          00990 CTRTST  POP   AF     ;Get Character Pressed
7E2E FE20        01000         CP    20H    ;Key < Spacebar
7E30 3802        01010         JR    C,CNTRL ;Must be CONTROL CHAR
7E32 1811        01020         JR    LWRTST ;Not CONTROL Test for Lower Case
7E34 FE08        01030 CNTRL   CP    8H     ;Skip Selected Characters-Backspace
7E36 C8          01040         RET   Z      ;Process Ordinarily
7E37 FE01        01050         CP    1H     ;Break
7E39 C8          01060         RET   Z      ;Process Ordinarily
7E3A FE0D        01070         CP    0DH    ;Enter
7E3C C8          01080         RET   Z      ;Process Ordinarily
7E3D FE1B        01090         CP    1BH    ;Shifted UP Arrow
7E3F C8          01100         RET   Z      ;Process Ordinarily
7E40 FE18        01110         CP    18H    ;Shifted LEFT Arrow
7E42 C8          01120         RET   Z      ;Process Ordinarily
7E43 1812        01130         JR    FINDST ;Find String for Remaining CONTROL C
hars.
                 01140 ;
                 01150 ;       TEST FOR lower case
                 01160 ;
7E45 FE61        01170 LWRTST  CP    61H    ;=a
7E47 3806        01180         JR    C,UPTST ;Below Lower Case-See If UPPER Case
7E49 FE7B        01190         CP    7BH    ;>z
7E4B D0          01200         RET   NC     ;TOO High
7E4C D620        01210         SUB   20H    ;Must be a Lower Case
7E4E C9          01220         RET          ;Change & Process as UPPER Case
                 01230 ;
                 01240 ;       TEST FOR UPPER CASE
                 01250 ;
7E4F FE41        01260 UPTST   CP    41H    ;=A
7E51 D8          01270         RET   C      ;Too Low-Between Spacebar & @
7E52 FE5B        01280         CP    5BH    ;>Z
7E54 D0          01290         RET   NC     ;Too High Between Up Arrow & -
7E55 D621        01300         SUB   21H    ;Follow after Controls
                 01310 ;
                 01320 ;       FIND LOCATION IN TABLE
                 01330 ;
7E57 E5          01340 FINDST  PUSH  HL     ;Save Original
7E58 21817E      01350         LD    HL,WORDS ;Start of KEYWORD Table
7E5B 47          01360         LD    B,A    ;Save Number of KEYWORDS to Skip
7E5C 7E          01370 NOTDN   LD    A,(HL) ;Look @ BYTE in Table
7E5D 23          01380         INC   HL     ;Get Ready for Next
7E5E B7          01390         OR    A
7E5F 20FB        01400         JR    NZ,NOTDN ;Not End of KEYWORD
7E61 10F9        01410         DJNZ  NOTDN  ;Count Number of KEYWORDS
                 01420 ;
                 01430 ;       SEARCH FINISHED - STORE LOCATION
                 01440 ;
7E63 227F7E      01450         LD    (STRPTR),HL ;Save Pointer to String Wanted
7E66 3E01        01460         LD    A,1    ;Set Flag to
7E68 327E7E      01470         LD    (SUBFLG),A ;Indicate Substitution
7E6B E1          01480         POP   HL     ;Restore Original
                 01490 ;
                 01500 ;       ONE CHARACTER TO SCREEN
```

Listing 1 continues

```
                 ;        ALSO BRANCH FROM ENTRY
                 ;        POINT IF CONTINUING
                 ;
7E6C E5          CONTI    PUSH    HL        ;Save Original
7E6D 2A7F7E               LD      HL,(STRPTR) ;Get Addr of Char to Process
7E70 7E                   LD      A,(HL)    ;Put in 'A'
7E71 B7                   OR      A         ;See if Zero
7E72 2003                 JR      NZ,MORE   ;Continue if not Last
7E74 327E7E               LD      (SUBFLG),A ;Set Flag to Stop Process
7E77 23          MORE     INC     HL        ;Point to Next Character
7E78 227F7E               LD      (STRPTR),HL ;Save Address
7E7B E1                   POP     HL        ;Restore Original
7E7C C9                   RET               ;Output Character
                 ;
                 ;        SWITCHES/FLAGS
                 ;
7E7D 00          ONOFF    DEFB    0         ;FLAG-If Routine ON (00) or OFF (80H
)
7E7E 00          SUBFLG   DEFB    0         ;FLAG-If Currently Substituting(00)
or Not (01)
7E7F 0000        STRPTR   DEFW    0         ;Address of Char TO BE Substituted
                 ;
                 ;        REPLACEMENT WORDS
                 ;
7E81 0000        WORDS    DEFW    00        ;Skip Null(00H),CONTROL A <Break> (0
1H),
7E83 00                   DEFB    0         ;and CONTROL B <On/Off Toggle> (02H)

7E84 43                   DEFM    'CSAVE"'
7E8A 00                   DEFB    0
7E8B 53                   DEFM    'SET('
7E8F 00                   DEFB    0
7E90 52                   DEFM    'READ'
7E94 00                   DEFB    0
7E95 3F                   DEFM    '?FRE(A$)'
7E9D 0D00                 DEFW    000DH     ;Include ENTER Code
7E9F 47                   DEFM    'GOSUB'
7EA4 00                   DEFB    0
7EA5 00                   DEFB    0         ;Skip Backspace <0BH>
7EA6 49                   DEFM    'INKEY$'
7EAC 00                   DEFB    0
                 ; *** CONTROL J Available
7EAD 00                   DEFB    0
7EAE 4C                   DEFM    'LLIST'
7EB3 00                   DEFB    0
7EB4 4C                   DEFM    'LIST'
7EB8 00                   DEFB    0
7EB9 00                   DEFB    0         ;Skip Enter <0DH>
                 ;*** CONTROL N Available
7EBA 00                   DEFB    0
7EBB 50                   DEFM    'POKE'
7EBF 00                   DEFB    0
7EC0 50                   DEFM    'POINT('
7EC6 00                   DEFB    0
7EC7 4C                   DEFM    'LEFT$('
7ECD 00                   DEFB    0
7ECE 52                   DEFM    'RUN'
7ED1 0D00                 DEFW    000DH
7ED3 53                   DEFM    'SQR('
7ED7 00                   DEFB    0
7ED8 54                   DEFM    'TAB('
7EDC 00                   DEFB    0
7EDD 55                   DEFM    'USR(0)'
7EE3 00                   DEFB    0
7EE4 43                   DEFM    'CLS'
7EE7 00                   DEFB    0
7EE8 45                   DEFM    'EDIT'
7EEC 00                   DEFB    0
7EED 00                   DEFB    0         ;Skip Shift LEFT Arrow <18H>
7EEE 54                   DEFM    'TROFF'
7EF3 00                   DEFB    0
7EF4 41                   DEFM    'AUTO'
7EF8 0D00                 DEFW    000DH
7EFA 00                   DEFB    0         ;Skip Shift UP Arrow <1BH>
7EFB 2F                   DEFM    '/32755'  ;Entry for Restore BASIC Routine
7F01 00                   DEFB    0
                 ;*** CONTROL 9 Available
7F02 00                   DEFB    0
7F03 53                   DEFM    'SYSTEM'
7F09 00                   DEFB    0
7F0A 3F                   DEFM    '?MEM'
7F0E 0D00                 DEFW    000DH
7F10 41                   DEFM    'ASC('
7F14 00                   DEFB    0
7F15 56                   DEFM    'VAL('
7F19 00                   DEFB    0
7F1A 43                   DEFM    'CHR$('
```

Listing 1 continues

# ACCEL3

ACCEL3

ACCEL3

ACCEL3 $99.95

... YOU OWE IT TO YOUR BASIC PROGRAM

BASIC Compiler, Model I/III, all DOS

ACCEL2 Plus:

- Bigger optimized subset ①
- Quicker compilation ②
- More compact output ③
- Almost total compatibility ④

developed in England by Southern Software

ALGORIX
Allen Gelder Software

① Includes INP, OUT, multi-dimensional arrays.
② About 40 lines/second.
③ Only 10% -35% code growth.
④ Even unstructured for-next loops, variable-bound arrays.

Table treats ① and ③. Speed improvement ratios give minimum expected results!

Also **EDIT** Full screen BASIC editor.

**$99.95**
+ $2 Shipping
CA add 6%

SPEED/SPACE Performance Table

| Speed Improvement (Ratio) INT | SNG | DBL | STR | Operation | Space Degradation (Bytes) INT | SNG | DBL | STR |
|---|---|---|---|---|---|---|---|---|
| 178 | 28 | 20 | 7.3 | Assignment (LET) | −4 | 0 | 0 | 0 |
| 3.5 | 3.6 | 3.6 | 3.5 | Array Reference (1-dim) | 13 | 13 | 13 | 13 |
| 3.0 | 3.0 | 3.0 | 3.0 | Array Reference (2-dim) | 12 | 12 | 12 | 12 |
| 35 | 1.8 | 1.6 | | AND, OR | 4 | 7 | 7 | |
| 23 | 2.0 | 1.6 | 6.6 | Compare (=, >, <, etc.) | 3 | 10 | 10 | 3 |
| 57 | 1.8 | 1.4 | 3.6 | Add, Concatenate (+) | 1 | 6 | 6 | 2 |
| 48 | 1.8 | 1.3 | | Subtract (−) | 4 | 6 | 6 | |
| 1.5 | 1.5 | 1.1 | | Multiply (*) | 6 | 6 | 6 | |
| 1.08 | 1.17 | 1.02 | | Divide (/) | 6 | 6 | 6 | |
| 77 | 70 | 84 | 9.3 | Constant Reference | 0 | 6 | 4 | 4 |
| 7.1 | 1.9 | | | FOR—NEXT | 6 | 23 | | |
| 111 | 6.8 | 4.8 | | POKE | −1 | 5 | 5 | |
| 10 | 4.5 | 3.6 | | SET, RESET | −1 | 5 | 5 | |
| 47 | 4.6 | 3.0 | 8.1 | IF THEN ELSE | 3 | 9 | 9 | 3 |
| 33 | 4.3 | 3.5 | | ON expression GOTO | −2 | 0 | 0 | |
| 50 | 6.8 | 5.1 | | ON expression GOSUB | 0 | 3 | 3 | |
| 1.2 | 1.01 | 1.03 | 1.2 | PRINT simple-variable | −1 | −1 | −1 | −1 |
| 61 | 5.0 | 3.7 | | OUT | 5 | 11 | 11 | |
| | | | | Flow of Control | | | | |
| | 216 | | | GOTO | | −7 | | |
| | 74 | | | GOSUB/RETURN | | −10 | | |
| | | | | Functions | | | | |
| inf | inf | inf | inf | VARPTR | −3 | −3 | −3 | −3 |
| 5.2 | 1.9 | 1.7 | | POINT | 3 | 9 | 9 | |
| 38 | 2.3 | 1.7 | | INP | 5 | 8 | 8 | |
| 149 | 2.3 | 2.0 | | PEEK | 0 | 3 | 3 | |
| | | | | String Functions | | | | |
| | | | 53 | ASC | | | | 5 |
| | | | 258 | LEN | | | | 0 |
| | | | 4.8 | LEFT$ | | | | 1 |
| | | | 4.7 | RIGHT$ | | | | 1 |
| | | | 6.4 | MID$ | | | | 2 |
| | | | 25 | CHR$ | | | | 0 |
| | | | 36 | CVI | | | | 0 |
| | | | 16 | MKI$ | | | | 0 |
| | | | 7.1 | CVS | | | | 0 |
| | | | 25 | MKS$ | | | | 0 |
| | | | 5.4 | CVD | | | | 0 |
| | | | 16 | MKD$ | | | | 0 |

④ *Correct BASIC programs (no dynamic renaming, no out-of- range arguments) compile without modification.*

**You owe it to your programs.**

ALGORIX
**Allen Gelder Software**
**(415) 387-3131**
**Box 11721 San Francisco CA 94101**

136

**EDIT** **EDIT is to your line-editor as disk is to tape, lets you work and play in one more dimension.** **$40**

*"I keep this routine in protected high memory."*

gram is still intact—you just cannot get at it.

I keep this routine in protected high memory; putting it on the end of Autokey has worked out very well. I incorporated the system jump address into Autokey making it quite easy to run it.

When you want to use it put the Autokey program in keyword mode. Press shift, down arrow, period and System will appear. Press Enter and the System prompt *? will appear. Press shift, down arrow, comma and the routine's entry address in decimal will appear (/32755). Press Enter. The routine will execute and return you to the Basic Ready prompt ready to run, list, or CSAVE your program. Then you can continue as if nothing had happened. Remember, do *not* turn off the computer, or all is lost. (J. Kovacic developed the code for this routine; it was published in *Interface Age* some time ago.)

You can have your own machine language copy of this program by assembling the source code listing. You can also enter the hex instructions on the left of the listing with a monitor. (The Z-Bug program in the January 1981 issue of *80 Microcomputing*, one of the better ones around, works well on the Model III.) You can also use the Basic program listed, or, if all this seems too much I will give you a free high baud copy of source and object code. Just send me a

*Listing 1 continued*

```
7F1F 00              DEFB    0
7F20 44              DEFM    'DATA'
7F24 00              DEFB    0
7F25 52              DEFM    'RESET('
7F2B 00              DEFB    0
7F2C 46              DEFM    'FOR'
7F2F 00              DEFB    0
7F30 47              DEFM    'GOTO'
7F34 00              DEFB    0
7F35 52              DEFM    'RETURN'
7F3B 00              DEFB    0
7F3C 49              DEFM    'INPUT"'
7F42 00              DEFB    0
7F43 29              DEFM    ')'
7F44 00              DEFB    0
7F45 4C              DEFM    'LEN('
7F49 00              DEFB    0
7F4A 4C              DEFM    'LPRINT'
7F50 00              DEFB    0
7F51 4D              DEFM    'MID$('
7F56 00              DEFB    0
7F57 4E              DEFM    'NEXT'
7F5B 00              DEFB    0
7F5C 50              DEFM    'PEEK('
7F61 00              DEFB    0
7F62 50              DEFM    'PRINT@'
7F68 00              DEFB    0
7F69 52              DEFM    'RIGHT$('
7F70 00              DEFB    0
7F71 52              DEFM    'RND('
7F75 00              DEFB    0
7F76 53              DEFM    'STRING$('
7F7E 00              DEFB    0
7F7F 54              DEFM    'THEN'
7F83 00              DEFB    0
7F84 55              DEFM    'USING'
7F89 00              DEFB    0
7F8A 56              DEFM    'VARPTR('
7F91 00              DEFB    0
7F92 45              DEFM    'ELSE'
7F96 00              DEFB    0
7F97 43              DEFM    'CLOAD'
7F9C 00              DEFB    0
7F9D 54              DEFM    'TRON'
7FA1 00              DEFB    0
7FA2 32              DEFM    '2FORI=1TOLEN(Z$):POKEI+32673'
7FBE 2C              DEFM    ',ASC(MID$(Z$,I,1))'
7FD0 3A              DEFM    ':NEXT:POKEI+32673,0:END'
7FE7 0D00            DEFW    000DH    ;End of Last Substitution String
                     ;
                     ;***    ROUTINE TO RESTART BASIC    ***
                     ;
7FF3                 ORG     7FF3H    ;32755D
7FF3 2AA440          LD      HL,(40A4H)        ;Find Start of Text Storage
Area
7FF6 74              LD      (HL),H            ;Put Something Non-zero into
 PTR
7FF7 CDF81A          CALL    1AF8H             ;Adjust All PTRS
7FFA CD591B          CALL    1B59H             ;Fix(40F9H),(40FAH),perform
CLEAR
7FFD C3191A          JP      1A19H             ;To READY Prompt
0000                 END
00000 TOTAL ERRORS
```

## *"Now you can type in all those super 80 Micro programs in half the time!"*

tape with one of your favorite programs on it and I will return it with Autokey. (Don't forget some instructions!)

### Modifications

As with any program, just when you finish, you think of many things to add and change. I am planning to change the check for 00H bytes at the end of each keyword to check for a high order bit in the last character of the keyword instead. I'll use the space taken up by the zeroes to add Auto Edit from the February 1981 issue of *80 Microcomputing*. I would be very happy to hear from you if you enhance this program—just drop me a line.

I sequenced the initialization code at the beginning of the program to make it easy to restart if you reset the computer and end up back at the CASS? prompt. Push Enter twice, Enter System, then /32224, and it will start up again.

You can also set another protected memory size in answer to the memory size question and retain it by entering System and /32244 to skip the auto memory size routine.

Now you can type in all those super *80 Microcomputing* programs in half the time! ■

*Program Listing 2*

```
10 REM      ***************************
20 REM      **                       **
30 REM      **     A U T O - K E Y   **
40 REM      **       Don  Rigg       **
50 REM      **       July,1981       **
60 REM      **                       **
70 REM      ***************************
80 CLS:PRINT@524,CHR$(23)"AUTOKEY NOW LOADING"
90 REM  *** Protect Memory From 32224 (7DE0H) On ***
100 POKE16561,222:POKE16562,125:CLEAR 50
110 DEFINT B:DEFDBL C
120 REM  *** Read AUTOKEY Object Code ***
130 FOR A% = 32224 TO 32387:      REM 7DE0H TO 7E83H
140     READ B
150     POKE A%,B
160     C=C+B
170 NEXT
180 IF C <> 17956 THEN PRINT "Error in Data Stmts 610 to 710":
 END
190 REM   *** Read RESTORE BASIC Object Code ***
200 C=0
210 FOR A = 32755 TO 32767:      REM 7FF3H TO 7FFFH
220     READ B
230     POKE A,B
240     C=C+B
250 NEXT
260 IF C <> 1432 THEN PRINT "Error in Data Stmt 730":END
270 REM   *** Read KEYWORDS ***
280 C=0
290 FOR A% = 32388 TO 32754:   REM 7E84H TO 7FF2H
300     READ B$
310     B = ASC(B$)
320     IF B = 38 THEN B = 13:   REM Change & to ENTER
330     IF B = 48 THEN B = 0:    REM Change 0 to NULL
340     IF B = 39 THEN B = 34:   REM Change ' to "
350     IF B = 33 THEN B = 48:   REM Change ! to 0
360     POKE A%,B
370     C=C+B
380 NEXT
390 IF C <> 20500 THEN PRINT "Keywords different from Original L
ist      - Program Continuing"
400 REM   *** Execute Object Code ***
410 POKE 16526,224:POKE16527,125:     REM 7DE0
420 CLS
430 A = USR(0)
440 REM     *** Data For AUTOKEY Object Code ***
450 DATA 62,201,50,226,65,33,222,125,34,177,64,33,172,125,34,160

460 DATA 64,205,114,27,42,22,64,34,15,126,33,8,126,34,22,64
470 DATA 62,0,50,25,64,195,25,26,58,126,126,183,32,94,205,36
480 DATA 48,183,200,245,254,2,32,12,58,125,126,198,128,50,125
490 DATA 126,241,62,0,201,58,125,126,254,0,40,2,241,201,241
500 DATA 254,32,56,2,24,17,254,8,200,254,1,200,254,13,200,254
510 DATA 27,200,254,24,200,24,18,254,97,56,6,254,123,208,214
520 DATA 32,201,254,65,216,254,91,208,214,33,229,33,129,126
530 DATA 71,126,35,183,32,251,16,249,34,127,126,62,1,50,126
540 DATA 126,225,229,42,127,126,126,183,32,3,50,126,126,35,34
```

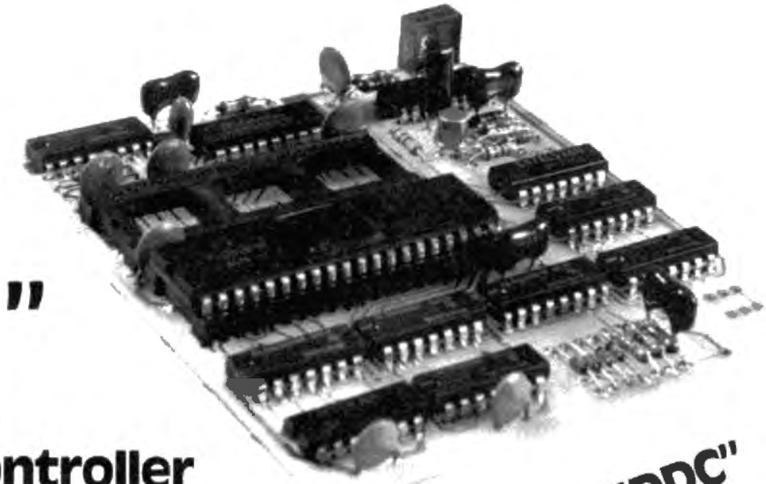
*Listing 2 continues*

✓See List of Advertisers on Page 451

*Listing 2 continued*

```
550 DATA 127,126,225,201,0,0,211,126,0,0,0
560 REM    *** Data For RESTORE BASIC Object Code ***
570 DATA 42,164,64,116,205,248,26,205,89,27,195,25,26
580 REM    *** Data For KEYWORDS ***
590 DATA C,S,A,V,E,',0
600 DATA S,E,T,(,0
610 DATA R,E,A,D,0
620 DATA ?,F,R,E,(,A,$,),&,0
630 DATA G,O,S,U,B,0,0
640 DATA I,N,K,E,Y,$,0,0
650 DATA L,L,I,S,T,0
660 DATA L,I,S,T,0,0,0
670 DATA P,O,K,E,0
680 DATA P,O,I,N,T,(,0
690 DATA L,E,F,T,$,(,0
700 DATA R,U,N,&,0
710 DATA S,Q,R,(,0
720 DATA T,A,B,(,0
730 DATA U,S,R,(,2,),0
740 DATA C,L,S,0
750 DATA E,D,I,T,0,0
760 DATA T,R,O,F,F,0
770 DATA A,U,T,O,&,0,0
780 DATA /,3,2,7,5,5,0,0
790 DATA S,Y,S,T,E,M,0
800 DATA ?,M,E,M,&,0
810 DATA A,S,C,(,0
820 DATA V,A,L,(,0
830 DATA C,H,R,$,(,0
840 DATA D,A,T,A,0
850 DATA R,E,S,E,T,(,0
860 DATA F,O,R,0
870 DATA G,O,T,O,0
880 DATA R,E,T,U,R,N,0
890 DATA I,N,P,U,T,',0
900 DATA ),0
910 DATA L,E,N,(,0
920 DATA L,P,R,I,N,T,0
930 DATA M,I,D,$,(,0
940 DATA N,E,X,T,0
950 DATA P,E,E,K,(,0
960 DATA P,R,I,N,T,@,0
970 DATA R,I,G,H,T,$,(,0
980 DATA R,N,D,(,0
990 DATA S,T,R,I,N,G,$,(,0
1000 DATA T,H,E,N,0
1010 DATA U,S,I,N,G,0
1020 DATA V,A,R,P,T,R,(,0
1030 DATA E,L,S,E,0
1040 DATA C,L,O,A,D,0
1050 DATA T,R,O,N,0
1060 DATA 2,F,O,R,I,=,1,T,O,L,E,N,(,Z,$,),":",P,O,K,E,I,+
1070 DATA 3,2,6,7,3,",",A,S,C,(,M,I,D,$,(,Z,$,",",I,",",1,),)
1080 DATA ":",N,E,X,T,":",P,O,K,E,I,+,3,2,6,7,3,",",I,":"
1090 DATA E,N,D,&,0,0,0,0,0,0,0,0,0,0,0,0
```

# Aerocomp's Proven Best-By Test! The "DDC"



"DDC"

## Double Density Controller

### ★ Technical Superiority

At last! A double density controller for Model I with HIGHER PROBABILITY OF DATA RECOVERY THAN WITH ANY OTHER DOUBLE DENSITY CONTROLLER ON THE MARKET TODAY! The "DDC" from Aerocomp. No need to worry about the problems that keep cropping up on existing products. AEROCOMP'S new analog design phase lock loop data separator has a wider capture window than the digital types currently on the market. This allows high resolution data centering. The finest resolution available with digital circuitry is 125 ns (nano seconds). The "DDC" analog circuit allows infinately variable tuning. Attack and settling times are optimum for 5-1/4 inch diskettes.
The units presently on the market use a write precompensation circuit that is very "sloppy". Board to board tolerance is extremely wide - in the order of ± 100 ns. The "DDC" is accurate to within ± 20 ns.
The bottom line is state of the art reliability!

### ★ Test Proven

Tests were conducted on AEROCOMP'S "DDC", Percom's "Doubler A"* and "Doubler II"* and LNW's "LNDoubler"** using a Radio Shack TRS80*** Model I, Level 2, 48 K with TRS80 Expansion Interface and a Percom TFD100 * disk drive (Siemens Model 82). Diskette was Memorex 3401. The test diskette chosen was a well used piece of media to determine performance under adverse conditions. The various double density adapters were installed sequentially in the expansion interface.
The test consisted of formatting 40 tracks on the diskette and writing a 6DB6 data pattern on all tracks. The 6DB6 pattern was chosen because it is recommended as a "worst case" test by manufacturers of drives and diskettes. An attempt was then made to read each sector on the disk once - no retrys. Operating system was Newdos/80, Version 1.0, with Double Zap, Version 2.0. Unreadable sectors were totalled and recorded. The test was run ten times with each double density controller and the data averaged. Test results are shown in the table.

### ★ Features

TRS80 Model I owners who are ready for reliable double density operation will get (1) 80% more storage per diskette, (2) single and double density data separation with far fewer disk I/O errors, (3) single density compatibility and (4) simple plug-in installation. Compatible with all existing double density software.

### ★ Value

$139.95 **for the BEST double density controller on the market.**
$179.95 for "DDC" complete with DOSPLUS 3.3D
$229.95 for "DDC" complete with LDOS

**★ TEST RESULTS ★**

| MFR & PRODUCT | SECTORS LOCKED OUT (AVG) |
|---|---|
| AEROCOMP "DDC" | 0 |
| PERCOM "DOUBLER II" | 18 |
| PERCOM "DOUBLER A" | 250 |
| LNW "LNDOUBLER" | 202 |

**Note: test results available upon written request. All tests conducted prior to 8-25-81**

Aerocomp's 14 day money back guarantee applies to hardware only.
Specials will be prorated Shipping $2.00 in Cont. US. See opposite page for details.

## Data Separators

The advances that make the "DDC" great are incorporated in the new AEROCOMP Single Density Data Separator ("SDS") and Double Density Data Separator ("DDS").

### ★ Has your original manufacturer left you holding the bag?

**If you already own a Percom "Doubler A", "Doubler II" or LNW "LNDoubler", the AEROCOMP "DDS" will make it right. Look at the test results:**

| MFR. & PRODUCT | SECTORS LOCKED OUT | |
|---|---|---|
| | WITHOUT "DDS" | WITH "DDS" |
| PERCOM "DOUBLER II" | 18 | 1 |
| PERCOM "DOUBLER A" | 250 | 0 |
| LNW "LNDOUBLER" | 202 | 0 |

Note: Same test procedures as "DDC".
* Trademark of Percom Data Co.
** Trademark of LNW
*** Trademark of Tandy Corporation

**★ "DDS" ..... $49.95**
(Use 1791 chip from your DD Controller)

**★ "DDS"** with disk controller chip included ....... **$79.95**

**★ Disk controller chip ..... $34.95**

(Shipping $2.00 Cont. US - see opposite page for details)

**Plugs directly into your existing Double Density Controller.**

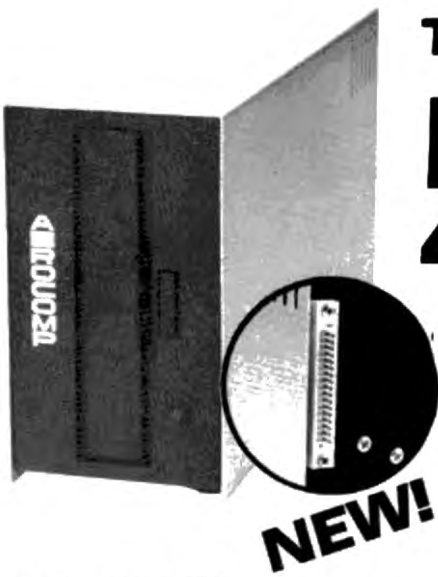
✓82

### Do you need a Single Density Data Separator?

The internal data separator in the WD1771 chip (R/S Expansion Interface) is NOT recommended by WD for reliable data transfer. Do you have any of these problems: Lost data, tracks locked out, CRC errors, disk retry? YOU NEED ONE!

**★ "SDS" ..... $29.95**
(For Mod. I; shipping $2.00)

**See opposite page▸▸▸▸▸▸**




# TRS 80*

# DISK DRIVES

# 40 & 80 TRACK

## as low as $259.95

### NEW LOW PRICES

Thanks to you our sales volume has allowed us to cut costs and we're passing the savings along. We offer the best disk drive value in the market place. Reliability, features and cost tough to beat. We deliver...and we stand behind our products. AEROCOMP is the only manufacturer to offer FREE TRIAL! Buy Aerocomp drives today. You deserve the best.

### BEST FEATURES

★ Fast 5 ms. track-track access time.
★ Operates single or double density.
★ "FLIPPY" feature for lower media cost (40-1 & 80-1)
★ Head load solenoid.
★ Disk ejector.
★ Easy entry door.
★ **NEW** EXTERNAL DRIVE CABLE CONNECTION. (no longer need to remove the cover to hook up cable)

# EPSON MX80 PRINTERS

## $469.00

TRS80 Cable (I & III)....... $9.95*
APPLE Cable & Interface. $105.00
RS232 Interface.......... $69.95
Ribbon Cartridge, Black... $9.95

Add $6.00 shipping (Cont. U.S.) does not include cable.

## MX80 F/T

All the features of MX80 plus Friction Feed. Shipping & cables as above.

## $569.00

*With Purchase of Printer

### ORDER TODAY!

★ 40-Track Drive.......... $299.95
★ 80-Track Drive.......... $409.95
★ 40-Track "FLIPPY" Drive.... $329.95
★ 80-Track "FLIPPY" Drive.... $419.95
★ 40-Track Dual-Head Drive . $419.95
★ 80-Track Dual-Head Drive . $569.95

All above drives are complete with silver enclosure, power supply and external drive cable connection. 115 VAC 50/60 Hz, 230 VAC 50/60 Hz available on spec at order. Add $15.00

★ 40-Track Bare Drive........ $259.95
★ 80-Track Bare Drive........ $359.95
★ 40-Track "FLIPPY" Bare Drive $279.95
★ 80-Track "FLIPPY" Bare Drive $369.95

## ★ SPECIAL PACKAGES ★

STARTER A................ $339.95
40-Track Drive, 2-drive cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

STARTER B................ $369.95
40-Track "FLIPPY" Drive, cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

COMBO C.................. $465.00
40-Track "FLIPPY" Drive, 2-drive cable, **LDOS**, Freight & Ins.

COMBO D.................. $559.00
80-Track "FLIPPY" Drive, 2-drive cable, **LDOS**, Freight & Ins.

COMBO E.................. $709.00
Two 40-Track "FLIPPY" Drives, 4-Drive cable, TRSDOS 2.3 Disk & Mannual, Freight & Ins.

COMBO F.................. $989.00
Two 80-Track "FLIPPY" Drives, 4-drive **LDOS** Ins.

COMBO G.................. $889.00
Two 40-Track Dual-Head Drives, cable, smalLDOS 2.3 Disk & Manual, Freight & Ins.

COMBO H.................. $1289.00
Two 80-Track Dual-Head Drives, cable, **LDOS**, Freight & INS.

DOSPLUS 3.4 or NEWDOS / 80 2.0 available with any package at extra cost.
Add $5.00 per drive for shipping & handling (Cont. US)

### FREE TRIAL OFFER

Order your AEROCOMP Disk Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return it, packed in the original shipping container, for a full refund. (Hardware only. Special packages will be prorated) We have complete confidence in our products and we know you will be satisfied! **ORDER TODAY!!!**

### WARRANTY

We offer you a 120 day unconditional warranty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes nescessary, our service department is fast, friendly and cooperative.

### 100% TESTED

AEROCOMP Disk Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!!

### ORDER NOW!!

To order by mail, specify Model Number(S) of Drive, cable, etc. (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request C.O.D. shipment. Texas residents add 5% sales tax. Add $5.00 per drive for shipping & handling (Cont. US). Please allow 2 weeks for personal checks to clear our bank. No personal checks will be accepted on C.O.D. shipments-cash, money orders or certified checks only. You will receive a card showing the exact C.O.D. amount before your shipment arrives. Be sure to include your name and shipping address. You will be notified of the scheduled shipping date. Your bank credit card will NOT be charged until the day we ship.!!

| DRIVE CABLES | |
|---|---|
| 2-DRIVE | $24.95 |
| 4-DRIVE | $34.95 |

### WRITE AEROCOMP TODAY FOR MORE VALUES !!!

**CALL TOLL FREE FOR FAST SERVICE (800) 824-7888, OPERATOR 24**
**FOR VISA/MASTERCHARGE/C.O.D. ORDERS**
California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.
**TOLL FREE LINES WILL ACCEPT ORDERS ONLY!**
For Applications and Technical information, call (214) 337-4346 or drop us a card.

Dealers inquiries invited

**AEROCOMP**
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

✓82

*A general guide to the regions of ROM.*

# Memory Map—Level II

*Mark D. Goodwin*
*S.R. 79, Box 103*
*Orland, ME 04472*

This memory map covers Level II ROM memory and explains how RAM memory is used by the ROM routines. Level II uses memory locations 4000H–42E7H as its communication region, storing temporary values for Basic interpreter functions. The map also covers the memory mapped input/output from 37DEH-37FFH used to communicate with the keyboard, the printer, disk drives, and the real-time clock.

This map is a general guide to Level II ROM routines, not a complete reference source. Use this map with a disassembled listing of the Level II ROMs.

Write to me if you wish more detail in areas of this memory map.■

*Level II Memory Map*

```
****************************************************************
0000 - 0004    INITIALIZATION ROUTINE

0000           TURNS OFF DISK AND CLOCK INTERRUPTS
0001           ZERO A
0002 - 0004    JUMP TO INITIALIZATION ROUTINE AT 0674
****************************************************************
0005 - 0007    JUMP TO RST 08 CODE AT 1C96 VIA 4000
****************************************************************
0008 - 000A    RST 08

0008 - 000A    JUMP TO RST 08 CODE AT 1C96 VIA 4000
****************************************************************
000B - 000C    DISK ROUTINE

000B           GET JUMP ADDRESS
000C           JUMP TO SPECIFIED ADDRESS
****************************************************************
000D - 000F    DISK ROUTINE

000D - 000F    JUMP TO DISK LOADER ROUTINE
****************************************************************
0010 - 0012    RST 10

0010 - 0012    JUMP TO RST 10 CODE AT 1D78 VIA 4003
****************************************************************
0013 - 0017    KEYBOARD ROUTINE

0013           SAVE BC
0014 - 0015    B = KEYBOARD ENTRY CODE
0016 - 0017    JUMP TO DRIVER ENTRY ROUTINE
****************************************************************
0018 - 001A    RST 18

0018 - 001A    JUMP TO RST 18 CODE AT 1C90 VIA 4006
****************************************************************
001B - 001F    VIDEO ROUTINE

001B           SAVE BC
001C - 001D    B = VIDEO ENTRY CODE
001E - 001F    JUMP TO DRIVER ENTRY ROUTINE
****************************************************************
0020 - 0022    RST 20

0020 - 0022    JUMP TO RST 20 CODE AT 25D9 VIA 4009
****************************************************************
0023 - 0027    DISK ROUTINE

0023           SAVE BC
0024 - 0025    B = ENTRY CODE
0026 - 0027    JUMP TO DRIVER ENTRY ROUTINE
****************************************************************
0028 - 002A    RST 28

0028 - 002A    RET WITH NON DISK VIA 400C
****************************************************************
002B - 002F    KEYBOARD ROUTINE

002B - 002D    DE = KEYBOARD DRIVER ADDRESS
002E - 002F    JUMP TO KEYBOARD DRIVER
               *SCANS KEYBOARD AND RETURNS WITH BYTE IN A
****************************************************************
0030 - 0032    RST 30

0030 - 0032    RET WITH NON DISK VIA 400F
****************************************************************
0033 - 0037    VIDEO ROUTINE

0033 - 0035    DE = VIDEO DRIVER ADDRESS
0036 - 0037    JUMP TO VIDEO DRIVER
               *DISPLAY THE BYTE IN A
****************************************************************
0038 - 003A    RST 38

0038 - 003A    RET WITH NON DISK VIA 4012
****************************************************************
003B - 003F    PRINTER ROUTINE

003B - 003D    DE = PRINTER DRIVER ADDRESS
```

*Map continues*

See List of Advertisers on Page 451

```
003E - 003F     JUMP TO PRINTER DRIVER
                *PRINT THE BYTE IN A
**************************************************************
0040 - 0042     INPUT ROUTINE

0040 - 0042     JUMP TO INPUT ROUTINE AT 05D9
**************************************************************
0043            RETURN
**************************************************************
0044 - 0045     NOTHING HERE
**************************************************************
0046 - 0048     JUMP TO DRIVER ENTRY ROUTINE
**************************************************************
0049 - 004F     KEYBOARD ROUTINE

0049 - 004B     SCAN KEYBOARD
004C            CHECK FOR KEY PRESSED
004D            RETURN IF KEY PRESSED
004E - 004F     LOOP TILL KEY PRESSED
**************************************************************
0050 - 005F     KEYBOARD LOOKUP TABLE

0050            ENTER = 0D
0051            SHIFT ENTER = 0D
0052            CLEAR = 1F
0053            SHIFT CLEAR = 1F
0054            BREAK = 01
0055            SHIFT BREAK = 01
0056            UP ARROW = 5B
0057            SHIFT UP ARROW = 1B
0058            DOWN ARROW = 0A
0059            SHIFT DOWN ARROW = 1A
005A            LEFT ARROW = 08
005B            SHIFT LEFT ARROW = 18
005C            RIGHT ARROW = 09
005D            SHIFT RIGHT ARROW = 19
005E            SPACE = 20
005F            SHIFT SPACE = 20
**************************************************************
0060 - 0065     DELAY LOOP

0060            DEC COUNTER
0061 - 0062     TEST FOR ZERO
0063 - 0064     LOOP TILL ZERO
0065            RETURN
                *BC IS USED AS THE COUNTER
**************************************************************
0066 - 0074     NMI RESET

0066 - 0068     RESET STACK POINTER
0069 - 006E     TEST FOR DISK
006F - 0071     JUMP IF DISK
0072 - 0074     JUMP TO LEVEL II RENTRY
                *THIS IS CALLED BY PRESSING THE RESET IN THE
                BACK OF THE KEYBOARD
**************************************************************
0075 - 0104     INITIALIZATION ROUTINE

0075 - 007F     MOVES LEVEL II POINTERS FROM ROM LOCATIONS
                18F7 - 191D TO RAM LOCATIONS 4080 - 40A6
0080 - 0089     SET LOCATION 41E5 - 41E7 TO INSTRUCTION
                LD A,(2C00)
008A - 0090     SET INPUT BUFFER POINTER (40A7) TO 41E8
0091 - 009E     SET DISK BASIC COMMAND VECTORS FROM 4152 - 41A5
                TO JP 012D INSTRUCTION *L3 ERROR*
009F - 00A7     SET DOS EXIT VECTORS FROM TO RETURNS
00A8 - 00AB     ZERO START OF BASIC MEMORY
00AC - 00AE     SET STACK POINTER TO 41F8
00AF - 00B1     CALL INITIALIZATION OF BASIC POINTERS AND
                VARIABLES ROUTINE
00B2 - 00B4     CLEAR SCREEN
00B5 - 00B7     POINT TO 'MEMORY SIZE?' MESSAGE
00B8 - 00BA     DISPLAY IT
00BB - 00BD     GET RESPONSE
00BE - 00BF     IF BREAK PRESSED DO IT AGAIN
00C0 - 00C3     IF THERE WAS A RESPONSE JUMP
00C4 - 00D5     GET END OF COMPUTER MEMORY AND JUMP
                *THIS IS DONE IF YOU DON'T RESPOND TO MEM SIZE?*
00D6 - 00D8     PUT BINARY OF RESPONSE IN DE
00D9 - 00DC     IF RESPONSE ISN'T RIGHT JUMP TO SN ERROR
00DD - 00DE     HL = RESPONSE - 1
00DF - 00E4     TEST TO SEE IF THIS LOCATION EXISTS
00E5 - 00E6     IF IT DOESN'T EXIST GET NEW RESPONSE
00E7 - 00EB     TEST FOR MINUMUM RESPONSE 17428
00EC - 00EE     IF BELOW THEN OM ERROR
00EF - 00F8     STORE MEMORY SIZE AND START OF STRING SPACE
00F9 - 00FB     CALL INITIALIZATION OF BASIC POINTERS AND
                VARIABLES ROUTINE
00FC - 00FE     POINT TO 'RADIO SHACK LEVEL II BASIC' MESSAGE
00FF - 0101     DISPLAY IT
0102 - 0104     JUMP TO READY ROUTINE
**************************************************************
0105 - 0110     'MEMORY SIZE' MESSAGE
**************************************************************
0111 - 012C     'RADIO SHACK LEVEL II BASIC' MESSAGE
**************************************************************
012D - 0131     L3 ERROR ROUTINE

012D - 012E     E = ERROR CODE
012F - 0131     JUMP TO ERROR ROUTINE
**************************************************************
0132 - 019C     GRAPHICS ROUTINE

0132 - 013A     PUT GRAPHIC CODE ON STACK:
                 0 POINT
                -1 SET
                +1 RESET
013B - 013C     CHECK SYNTAX
013D - 013F     A = X VALUE
0140 - 0141     CHECK FOR > 127
0142 - 0144     IF > THEN FC ERROR
0145            SAVE X VALUE
0146 - 0147     CHECK SYNTAX
0148 - 014A     A = Y VALUE
014B - 014C     CHECK FOR > 47
014D - 014F     IF > THEN FC ERROR
0150 - 017F     COMPUTE VIDEO LOCATION AND BIT MASK
0180 - 0181     IF POINT THEN JUMP
0182            RESTORE GRAPHIC CHARACTER
0183 - 0185     IF SET THEN JUMP
0186 - 018B     RESET BIT
018C - 018D     CHECK SYNTAX
018E            RETURN
018F - 0190     SET BIT AND RETURN
0192 - 019C     TEST BIT AND RETURN
**************************************************************
019D - 01C8     INKEY$ ROUTINE
019D - 019E     ADJUST AND SAVE ENCODED STATEMENT POINTER
019F - 01A4     JUMP IF SHIFT @ WAS PRESSED
01A5 - 01A7     SCAN KEYBOARD
01A8 - 01AA     IF NO KEY PRESSED THEN JUMP
01AB            SAVE CHARACTER
01AC - 01AF     ZERO LAST KEY PRESSED STORAGE LOCATION
01B0            A = STRING LENGTH
01B1 - 01B3     CHECK FOR ENOUGH STRING SPACE
01B4            GET CHARACTER
01B5 - 01BB     SAVE STRING AND JUMP
01BC - 01C8     SAVE READY MESSAGE AS STRING AND RETURN
**************************************************************
01C9 - 01D2     CLS ROUTINE
**************************************************************
01D3 - 01D8     RANDOM ROUTINE
**************************************************************
01D9 - 01F7     CASSETTE ROUTINE

01D9 - 01F7     MAKE A CASSETTE PULSE AND RETURN
**************************************************************
01F8 - 01FD     CASSETTE ROUTINE

01F8 - 01FD     TURN CASSETTE OFF AND RETURN
**************************************************************
01FE - 0211     CASSETTE ROUTINE

01FE - 0211     GET DRIVE NUMBER
**************************************************************
0212 - 021D     CASSETTE ROUTINE

0212 - 021D     DEFINE DRIVE AND RETURN
**************************************************************
021E - 022B     CASSETTE ROUTINE

021E - 022B     RESET TAPE INPUT SO VIDEO DOES'T GO TO 32
                CHARACTER MODE AND RETURN
**************************************************************
022C - 0234     CASSETTE ROUTINE

022C - 0234     BLINK * DURING CASSETTE READ
**************************************************************
0235 - 0240     CASSETTE ROUTINE

0235 - 0236     SAVE BC AND HL
0237 - 0238     SET COUNTER TO READ 8 BITS
0239 - 023B     GET A BIT
023C - 023D     LOOP TILL 8 BITS HAVE BEEN READ
023E - 023F     GET HL AND BC
0240            RETURN WITH BYTE READ IN A
**************************************************************
0241 - 0260     CASSETTE ROUTINE

0241 - 0260     READ 1 BIT AND RETURN
**************************************************************
0261 - 0263     CASSETTE ROUTINE

0261 - 0263     WRITE A CLOCK PULSE
**************************************************************
0264 - 027D     CASSETTE ROUTINE

0264 - 027D     WRITE BYTE IN A AND RETURN
**************************************************************
027E - 0292     CASSETTE ROUTINE

027E - 0292     WRITE CASSETTE LEADER AND RETURN
**************************************************************
0293 - 02A8     CASSETTE ROUTINE

0293 - 02A8     READ CASSETTE LEADER AND RETURN
**************************************************************
02A9 - 0313     SYSTEM ROUTINE *ENTRY POINT IS 02B2

02A9 - 02AE     READ 2 BYTES AND SAVE THEM
02AF - 02B1     TURN CASSETTE OFF
02B2 - 02B4     CALL DOS EXIT AT 41E2
02B5 - 02B7     SET ENCODED STATEMENT POINTER
02B8 - 02BA     DISPLAY CARRIAGE RETURN
02BB - 02BF     DISPLAY *
02C0 - 02C2     GET RESPONSE
02C3 - 02C5     IF BREAK PRESSED THEN JUMP
02C6            POINT TO 1ST CHARACTER INPUT
02C7 - 02C9     IF SN ERROR THEN JUMP
02CA - 02CD     IF / THEN JUMP
02CE - 02D0     READ CASSETTE LEADER
02D1 - 02D3     READ 1 BYTE
02D4 - 02D7     IF NOT NAME HEADER THEN LOOP
02D8 - 02E6     READ NAME IF NO MATCH SKIP TO NEXT PROGAM
                ON TAPE
02E7 - 02E9     BLINK *
02EA - 02EC     READ 1 BYTE
```

*Map continues*

## NEW MODEM GAME

Presenting Terminal War. Command a starship as you battle against a human opponent, the computer, or both. Yes, the computer can control up to 28 ships. Choose from fifteen commands as you fight the alien ships and the other player. But be careful, the aliens will intelligently attack with various weapons, advancing and retreating as they try to destroy you.

Every time you play Terminal-War the scenario is different. You can also make each game as easy or as hard as you like because you control the strength and the number of ships.

### Only $29.95

**Includes**: 2 detailed manuals, 2 disks each with the 2 player modem version, 1 player version, and a special terminal program so that 16K computer users can play Terminal-War. (The terminal program is **transferable to cassette**. Use the 32K computer.)
Cassette with the terminal program already on it $1.00

Minimum system requirements

2 Player version: 1 16K TRS 80 level 2 computer with a full duplex modem and 1 32K 1 disk TRS-80 computer with a full duplex modem

1 Player version: 1 32K 1 disk TRS 80 computer

**—DEALER INQUIRIES INVITED—**

Ordering Information
*Specify Mod I or Mod III disks or one disk of each.*

We accept Money orders, Certified checks, C.O.D., and personal checks (Up to 2 weeks to clear)
Please add $2.00 shipping. NY residents add 7% sales tax

Mail orders: **Ocean Computer Products**
P.O. Box 6474
Ithaca, NY 14850

24 Hour order line (607) 273-6608

TRS 80 is a trademark of Radio Shack a Tandy Corp

525

---

PROFESSIONAL SOFTWARE **TRS-80® MODEL II & 16** NOW AVAILABLE!

**BASIC COMPILER** BY MICROSOFT • EASY TO USE **$339.00***
• TRSDOS* 2.0 COMPATIBLE • INCLUDES **M-80, CREF-80 & EDIT 80**
• UP TO 30 TIMES FASTER • AUGMENTED WITH OUR DOCUMENTATION

**DISK SORT** • YOU DON'T HAVE TO BE A PROGRAMMER TO USE IT! **$69.99***
• MENU DRIVEN • RANDOM FILES • EASY TO USE • SPEC's SAVED ON DISK
• EASILY INSERTED INTO JOB STREAM FOR NON-STOP RUNNING • FULL DISK CAPACITY
• 1,000 REC's SORTING 15 BYTES, INCLUDING ALL DISK I/O - 2 MIN, 45 SEC's.
**HARD DISK SUPPORTED**
DOC ONLY $10.00 DEDUCTIBLE ON PURCHASES

**BASIC CROSS REFERENCE** • SAVE HOURS! **$59.99***
• SEE USED LINE#s/NAMES AT A GLANCE • PRODUCES NICE PROG. LISTING • INDISPENSABLE
DOC ONLY $10.00 DED ON PURCH

**EPSON** MX-80 **$460.00** MX-100 **$700.00*** **HURRY-LIMITED TIME!**

**PAPER** 9½x11 18lb WHITE STRIPS TO 8½x11 3,000/BOX **$25.00***

**MAILING LIST** • HIGH CAPACITY • SPANS DISK • NO SORTING **$149.95***
• FAST RETRIEVAL-UPDATE • R/S CONVERSION INCLUDED • LONG/SHORT FORMATS
• ESTIMATED AVAILABILITY - DEC 1982
DOC ONLY $10.00 DED ON PURCH

*PLUS POSTAGE & HANDLING. PRICES SUBJECT TO CHANGE WITHOUT NOTICE

TRS-80 & TRSDOS ARE REGISTERED TRADEMARKS OF TANDY CORP

**GOOD-LYDDON DATA SYSTEMS**
**5486 RIVERSIDE DR., CHINO, CA 91710**
**(714) 980-4563**

186

---

## Coordinated Business Software

**D.B. Software Co.** announces a new line of coordinated business software designed for the Model III TRS-80. The BASE system consists of a **General Ledger System** with **Accounts Receivable** and **Accounts Payable**. Other modules can be purchased and added as needed.

| | |
|---|---|
| BASE System (GL-AR-AP) | $200.00 |
| Coordinated Modules: | |
| Payroll | $ 75.00 |
| Order Entry | $125.00 |
| Inventory | $129.95 |
| Purchase Order Entry | $125.00 |
| Inventory (special for Auto Parts Suppliers) | $129.95 |

*Also available:*

| | |
|---|---|
| Job Costing with GL and AP (Payroll optional) | $500.00 |
| Mailing Lister | $100.00 |

*Coming soon:*
Apartment Management System, Fixed Assets System, Manufacturing Inventory Management System

Send S.A.S.E. for more information
**D.B. Software Co.**
11840 NE Brazee, Portland, OR 97220
Phone (503) 255-7735

DEALER INQUIRIES INVITED

TRS-80 is a trademark of Tandy Corp

410

---

# LEVEL IV PRODUCTS, INC.
# TRS-80® DISCOUNT DEALER

**NO AUTHORIZED DEALER BEATS OUR PRICES ON PURE R/S EQUIPMENT**

THIS IS A FAST MOVING INDUSTRY, PRICES CHANGE FASTER THAN ADS, WE HAD A COMPLETE AD LAYOUT FOR THIS MONTH WITH PRICES, BUT BEFORE WE COULD EVEN SEND IT IN THEY CHANGED, BUT THE NEWS IS GOOD, THEY WENT DOWN AGAIN!!!

SINCE THIS AD WAS DESTROYED, AND THERE IS NO TIME TO PREPARE A NEW ONE, LET'S JUST TALK ABOUT A FEW THINGS THAT WE AT LEVEL IV THINK SHOULD BE IMPORTANT.

THERE ARE TWO KINDS OF TRS-80(R) COMPUTERS, THE "PURE RADIO SHACK(R)" VERSION AND THE "BUILT-UP" VERSION, THEY ARE BOTH AVAILABLE FROM A NUMBER OF SUPPLIERS VIA MAIL ORDER, AT PRICES BELOW RETAIL.

THE "PURE" VERSION WILL HAVE THE MANUFACTURER'S LIMITED 90 DAY WARRANTY IN EFFECT, (PROVIDED THE DEALER HANDLED THE SALE PROPERLY), BUT DON'T EXPECT THE LOCAL STORE TO GIVE YOU AN OPEN ARMS WELCOME, THOSE GUYS WORK ON COMMISSION YOU KNOW, SO YOUR CHOICE OF A DEALER HAS SOME IMPORTANCE, SOME OF THEM ARE LITTLE MORE THAN WAREHOUSES, VERY FEW HAVE EITHER SERVICE OR SUPPORT CAPABILITIES AFTER THE SALE, FOR THE USER WITH A PROBLEM. LEVEL IV OFFERS BOTH, WE HAVE A FULL TIME TECHNICAL SERVICE STAFF OF 8, AND A PROGRAMMING STAFF AS WELL, BOTH ARE AVAILABLE TO OUR CUSTOMERS DURING NORMAL BUSINESS HOURS MONDAY THRU SATURDAY, PHONE (313) 528-6200. YOUR HARDWARE PURCHASED FROM LEVEL IV ALSO HAS AN EXTENDED WARRANTY, AFTER THE ABOVE 90 DAYS SEND OR BRING IT BACK TO US FOR AN ADDITIONAL 3 MONTHS OF COVERAGE' (WARRANTY COPIES ARE AVAILABLE ON REQUEST).

YOU MUST RELY ON THE "BUILDER" OF A "BUILT-UP" UNIT TO PUT IN GOOD COMPONENTS, AND PROVIDE BOTH YOUR WARRANTY AND AFTER WARRANTY SERVICE, SELECT HIM CAREFULLY IF YOU GO THIS WAY.

LEVEL IV OFFERS BOTH "PURE" AND "BUILT-UP" UNITS, SEE OUR OTHER ADS IN THIS MAGAZINE.

- EPSON
- SMITH-CORONA
- HAYES
- MPI
- TANDON
- SOFTWARE
- SUPPLIES

RETAIL STORE NO. P-130

LEVEL IV PRODUCTS, INC.
134 SOUTH FRONT STREET
DOWAGIAC, MICHIGAN 49047
PHONE (616) 782-6600

PHONE/MAIL ORDER CENTER

LEVEL IV PRODUCTS, INC.
32429 SCHOOLCRAFT ROAD
LIVONIA, MICHIGAN 48150
PHONE: (MI) 313-525-6200
TOLL FREE: 800-521-3305

**CALL FOR OUR CURRENT LOW PRICES (MI) 313-525-6200 OTHERS 800-521-3305**
**WE ACCEPT (VISA-MC-CHECKS-COD) INQUIRE ABOUT OUR CASH PRE-PAY DISCOUNTS**

*Map continued*

```
02ED - 0313    READ PROGRAM AND JUMP
**************************************************************
0314 - 031C    CASSETTE ROUTINE

0314 - 031C    READ 2 BYTES AND PUT IN HL
**************************************************************
031D - 0329    SYSTEM ROUTINE

031D - 0329    COMPUTE JUMP ADDRESS AND GO
               *FOR ***** INPUT
**************************************************************
032A - 0347    OUTPUT ROUTINE

032A - 032B    SAVE BC AND CHARACTER TO OUTPUT
032C - 032E    CALL DOS EXIT AT 41C1
032F - 0332    CHECK FOR DEVICE TYPE
0333 - 0334    GET CHARACTER TO OUTPUT AND BC
0335 - 0337    IF CASSETTE THEN JUMP
0338 - 0339    IF PRINTER THEN JUMP
033A - 0347    DISPLAY CHARACTER ON VIDEO AND UDATE CURSOR
               THEN RETURN
               *ON ENTRY DEVICE TYPE FLAG (409C):
               -1 CASSETTE
                0 VIDEO
               +1 PRINTER
**************************************************************
0348 - 0357    VIDEO ROUTINE

0348 - 0357    CURSOR MAINTENANCE
**************************************************************
0358 - 0360    KEYBOARD ROUTINE

0358 - 035A    CALL DOS EXIT AT 41C4
035B - 0360    SCAN KEYBOARD AND RETURN WITH CHARACTER IN A
**************************************************************
0361 - 0383    INPUT ROUTINE

0361 - 0367    ZERO LAST KEY PRESSED AND CURSOR POSITION
0368 - 036A    CALL DOS EXIT AT 41AF
036B           SAVE BC
036C - 0370    GET START OF BUFFER AND LENGTH
0371 - 0373    GET INPUT
0374           SAVE FLAGS
0375 - 0378    GET INPUT LENGTH AND END OF INPUT
0379 - 037A    ZERO END OF INPUT
037B - 0380    GET BC AND HL = START OF INPUT - 1
0381           RETURN WITH CARRY IF BREAK WAS PUSHED
0382 - 0383    CLEAR FLAGS AND RETURN
**************************************************************
0384 - 038A    KEYBOARD ROUTINE

0384 - 0386    SCAN KEYBOARD
0387           CHECK FOR KEY PRESSED
0388           RETURN IF KEY PRESSED
0389   038A    LOOP TILL KEY PRESSED
**************************************************************
038B - 039B    PRINTER ROUTINE

038B - 039B    SET CARRIAGE POSITION TO START OF LINE
**************************************************************
039C - 03C1    PRINTER ROUTINE

039C - 03C1    PRINT CHARACTER IN A
**************************************************************
03C2 - 03E2    DRIVER ENTRY ROUTINE

03C2 - 03E2    JUMP TO APPROPRIATE DRIVER ADDRESS
**************************************************************
03E3 - 0457    KEYBOARD DRIVER

03E3 - 0457    SCANS KEYBOARD AND RETURNS WITH BYTE
**************************************************************
0458 - 058C    VIDEO DRIVER

0458 - 058C    DISPLAY BYTE AND RETURN
**************************************************************
058D - 05D8    PRINTER DRIVER

058D - 05D8    PRINT BYTE AND RETURN
**************************************************************
05D9 - 0673    INPUT ROUTINE

05D9 - 0673    INPUT INTO BUFFER
               *ON ENTRY HL = START OF BUFFER
               B = BUFFER SIZE
               WILL EXIT WITH CARRY IF BREAK PRESSED
**************************************************************
0674 - 06D1    INITIALIZATION ROUTINE

0674 - 0675    TURN OFF CASSETTE
0676 - 0680    SET UP RSTS, VIDEO, KEYBOARD, AND PRINTER
               DEVICE CONTROL BLOCKS IN RAM
0681 - 0684    SEND VALUE TO CASSETTE PORT
0685 - 068A    ZERO RAM LOCATIONS 4036 - 4062
068B - 0692    IF BREAK PRESSED JUMP
0693 - 0695    SET STACK TO 40D7
0696   069E    IF NON DISK JUMP
069F - 06A3    SELECT DISK 0 AS CURRENT DRIVE
06A4 - 06CB    LOAD DOS LOADER AND JUMP
06CC - 06D1    JUMP TO READY ROUTINE
**************************************************************
06D2 - 0707    ROM STORAGE LOCATION

06D2 - 0707    RST VECTORS AND DEVICE CONTROL BLOCK ARE
               STORED HERE
**************************************************************
0708   07B1    SINGLE PRECISION ADDITION ROUTINES
```

*Map continues*

# RUN BASIC PROGRAMS AT SUPER SPEED WITH ZBASIC 2.2.

## THE WORLDS FASTEST TRS-80 BASIC COMPILER from SIMUTEK

### BELIEVE IT OR NOT WE'VE ADDED MORE NEW FEATURES to the ONLY INTERACTIVE BASIC COMPILER for the TRS-80!

1. Speed increases of 10 - 100 times are typical after compilation.
2. Compiled code can be RELOCATED to run anywhere in memory. Code is even ROMable!
3. ZBASIC 2.2 NOW SUPPORTS BOTH RANDOM and SEQUENTIAL DISK I/O.
4. ZBASIC 2.2 is now a super tool for business programmers. RANDOM ACCESS FILES, and PRINT USING statements are supported as well as a HIGH PRECISION MATH package (with no rounding problems).
5. Special BUILT-IN MACHINE LANGUAGE COMMANDS to increase program operation by as much as 1000 times! Special commands are implemented for fast memory searching (CPDR, CPIR), block memory moves (LDIR, LDDR), inputting and printing HEX numbers, inserting MACHINE LANGUAGE into COMPILED CODE, disabling and enabling interrupts, inverting memory, 16 bit PEEKs and POKEs, and stack control, debug and much more.
6. ZBASIC 2.2 compiles the ENTIRE PROGRAM into to Z-80 machine language. (Not 8080 code or a combination of BASIC and machine language like some other compilers.) Clumsy LINKING LOADERS, and RUNTIME MODULES are not needed. ZBASIC 2.2 creates a ready to run MACHINE LANGUAGE program.
7. NO ROYALTIES imposed on registered ZBASIC owners.
8. Typical COMPILATION TIME is TWO SECONDS for a 4K program.
9. Use TRS-80 Basic to write ZBASIC programs!
10. Compile many existing programs with only minor changes. (Some BASIC programming experience is required.)
11. Fully compatible with both the Model I and the Model III. Mod I compiled programs work on a MODEL III, and visa-versa. ZBASIC works with NEWDOS-80, NEWDOS+, DOSPLUS, LDOS, MULTIDOS, ULTRADOS, TRSDOS etc.
12. BUILT-IN and much improved MUSIC and SOUND EFFECTS commands.
13. Improved CHAINING for disk users.
14. TIME$ now available on DISK version.
15. ZBASIC 2.2 now has an INPUT @ command (similar to PRINT @).
16. The TAB function will now tab 255 columns on a printer. (BASIC cannot tab past column 64.)
17. NEWDOS 80 2.0 USERS can use the CMD "dos command" function!
18. NEW and EASIER to use USR COMMANDS.
19. New math functions to calculate XOR and INTEGER REMAINDERS.
20. Logical STRING COMPARISONS are now supported.
21. The disk commands INSTR, MID$ ASSIGNMENT are now supported on both DISK AND TAPE ZBASIC.
22. DEFSTR is now supported.
23. Eight disk files may be opened simultaneously; random, sequential or mixed.
24. LINE INPUT#, is now supported.
25. Invoke the compiler by simply hitting these two keys: "-"
26. NEW 100+ PAGE MANUAL WITH DESCRIPTIONS AND EXAMPLE.
27. ZBASIC 2.2 Comes with CMDFILE/CMD program from MISOSYS, to allow appending or merging compiled programs and machine language programs from tape or disk.

### ZBASIC 2.2 DOES NOT SUPPORT THESE BASIC COMMANDS:

1. ATN, EXP, COS, SIN, LOG, TAN, and exponentiation. (However, subroutines are included in the manual for these functions.)
2. ERROR, ON ERROR GOTO, ERL, ERR RESUME.
3. No direct commands like AUTO, EDIT, LIST, LLIST ETC, although these commands may be used when writing programs.
4. Others NOT supported: CDBL, CINT, CSNG, DEFFN, FIX, FRE.
5. Normal CASSETTE I/O. (ZBASIC supports it's own SPECIAL CASSETTE I/O statements.)
6. SOME BASIC COMMANDS MAY DIFFER IN ZBASIC. For instance, END jumps to DOS READY, STOP jumps to BASIC READY etc.
7. MEMORY REQUIREMENTS: to approximate the largest BASIC program that can be compiled in your machine (at one time), enter BASIC and type: PRINT (MEM-6500)/2. Remember, you can merge compiled programs together to fill memory.

### ZBASIC 2.2 SPEED COMPARISON DEMO

To help give you an idea how fast compiled programs are, we have included this demo program:

### ZBASIC 2.2 DEMO PROGRAM

| | |
|---|---|
| Time to compile and run complete program | **: 0 MIN. 2 SEC.** |
| BASIC Execution speed MOD 1, LEVEL II | **: 7 MIN. 34 SEC.** |
| ZBASIC Execution speed MOD 1, LEVEL II | **: 0 MIN. 18 SEC.** |
| BASIC Program size (WITHOUT VARIABLES) | **: 895 BYTES** |
| ZBASIC Program size (WITHOUT VARIABLES) | **: 2733 BYTES** |

(Remember that the ZBASIC program includes an 1879 byte subroutine package.) Program shown exactly as compiled and run in BASIC and ZBASIC.

```
10 '========== ZBASIC 2.2 EXAMPLE PROGRAM AND TIME TEST=========
20 CLS:CLEAR100:DEFINT A-X:DEFSTR Z:DIM AA(64,24),Z(50):RANDOM
30 AA=100:BB=-1000:CC=3:DD=-3:EE=-9999:ST$="START TIME "+TIME$
40 FOR I=1TO127STEP2 :FOR J=47TO1STEP-3:XX=POINT(I,J):SET(I,J)
50 XX=(I-J)/CC*(7+I+J) :XX=ABS(INT(RND(I*J)-AA)+7) :RESET(I,J)
60 XX=PEEK(I+J) :POKE15360+I+J,J :OUT255,J AND (3*J):XX=INP(I)
70 AB$=STR$(I+J) :BA$=LEFT$(AB$,2) :AA(I/2,J/2)=VAL(BA$)+AA*3
80 BA$=BA$+RIGHT$(BA$,RND(3)) :XX=INSTR(1,BA$,"9"):XX=SQR(I*J)
90 BA$=MID$(BA$,2,2) :MID$(BA$,1,1)=Z :IF XX THEN 100 ELSE CLS
100 IF LEN(BA$))3 OR SGN(XX)=1 AND ASC(BA$)=32 THEN PRINT"+++";
110 IFPOS(0))62 THEN TRON:TROFF:PRINT ELSE XX=NOT(RND(99))+100
120 A$=INKEY$:IF A$="Y" OR A$="y" AND I)120 THEN PRINT"TRUE.."
130 RESTORE :READA,C,Z(J),D:GOSUB170:GOSUB170:GOSUB170:GOTO210
140 NEXT :PRINT"*";:NEXTI:CLS:PRINT@512,ST$,"STOP TIME ";TIME$
150 STOP'================ END OF MAIN TEST LOOP ==================
160 DATA 12345,-1,"TEST",-9999
170 ON RND(6) GOTO 180,190,200,180,190,200
180 RETURN
190 RETURN
200 RETURN
210 ON RND(9) GOSUB 180,190,200,180,190,200,180,190,200
220 GOTO140
```

NOTICE ZBASIC 2.0 OWNERS: you can upgrade your ZBASIC 2.0 for no charge. Just send us your original diskette/cassette and a S.A.S.E with your registered serial number and copy of your invoice. We will send you ZBASIC 2.2 and updates to your manual.

VISA, MASTERCARD, AMERICAN EXPRESS, C.O.D. ORDERS ONLY

**800 528-1149 order line**

| | |
|---|---|
| ZBASIC 2.2 DISK VERSION AND MANUAL | 89.95 |
| ZBASIC 2.2 TAPE VERSION AND MANUAL | 79.95 |
| ZBASIC 2.2 DISK & TAPE VERSION AND MANUAL | 99.99 |
| MANUAL ONLY (APPLIES TO PURCHASE) | 25.00 |

## SIMUTEK COMPUTER PRODUCTS INC.

**TECHNICAL QUESTIONS PLEASE CALL (602) 323-9391**
**4897 E. SPEEDWAY, TUCSON, ARIZONA 85712** ✓12

TRS-80 is tm of Radio Shack, a Tandy Corp.

*Map continued*

```
0708 - 070A    POINT HL TO .5 VALUE
070B - 070F    LOAD BCDE WITH VALUE POINTED TO BY HL AND JUMP
0710 - 0712    LOAD BCDE WITH VALUE POINTED TO BY HL
0713 - 0715    INVERTS THE VALUE IN REG1 @SUBTRACTION
0716 - 07B1    ADDS THE VALUES IN BCDE TO REG1 AND RETURNS
               IF NC OV ERROR
*****************************************************************
07B2 - 07B6    OV ERROR ROUTINE

07B2 - 07B3    E - ERROR CODE
07B4 - 07B6    JUMP TO ERROR ROUTINE
*****************************************************************
07B7 - 07C2    SYSTEM MATH ROUTINE
*****************************************************************
07C3 - 07D6    SYSTEM MATH ROUTINE
*****************************************************************
07D7 - 07F7    SYSTEM MATH ROUTINE
*****************************************************************
07F8 - 0808    SYSTEM MATH ROUTINE
*****************************************************************
0809 - 0846    REG1 = LOG(REG1) ROUTINE
*****************************************************************
0847 - 0891    REG1 = BCDE * REG1
*****************************************************************
0892 - 0896    SYSTEM MATH ROUTINE
*****************************************************************
0897 - 0954    REG1 = BCDE / REG1
*****************************************************************
0955 - 0963    SYSTEM MATH ROUTINE
*****************************************************************
0964 - 0976    SYSTEM MATH ROUTINE
*****************************************************************
0977 - 0989    REG1 = ABS(REG1)
*****************************************************************
098A - 0993    REG1 = SGN(REG1)
*****************************************************************
0994 - 09A3    SYSTEM MATH ROUTINE
*****************************************************************
09A4 - 09B0    SINGLE PRECISION REG1 TO STACK
*****************************************************************
09B1 - 09B3    LOAD VALUE POINTED TO BY HL INTO BCDE
*****************************************************************
09B4 - 09BE    REG1 = BCDE
*****************************************************************
09BF - 09CA    LOAD VALUE IN BCDE TO LOCATION POINTED TO BY HL
*****************************************************************
09CB - 09CD    MOVE REG1 TO (HL) SINGLE ONLY
*****************************************************************
09CE - 09D1    MOVE (DE) TO (HL) SINGLE ONLY
*****************************************************************
09D2           MOVE (HL) TO (DE) FOR NTF BYTES
*****************************************************************
09D3 - 09D5    MOVE (DE) TO (HL) FOR NTF BYTES
*****************************************************************
09D6           MOVE (DE) TO (HL) FOR A BYTES
*****************************************************************
09D7 - 09DE    MOVE (DE) TO (HL) FOR 6 BYTES
*****************************************************************
09DF - 09F3    SYSTEM MATH ROUTINE
*****************************************************************
09F4 - 0A0B    SYSTEM MATH ROUTINE
*****************************************************************
0A0C - 0A25    SINGLE PRECISION COMPARE REG1 - BCDE
*****************************************************************
0A26 - 0A38    SYSTEM MATH ROUTINE
*****************************************************************
0A39 - 0A48    INTEGER COMPARE HL - DE
*****************************************************************
0A49 - 0A77    SYSTEM MATH ROUTINE
*****************************************************************
0A78 - 0A7E    DOUBLE PRECISION COMPARE REG2 - REG1
*****************************************************************
0A7F - 0AA2    CINT(REG1) ROUTINE
*****************************************************************
0AA3 - 0AB0    SYSTEM MATH ROUTINE
*****************************************************************
0AB1 - 0ADA    CSNG(REG1) ROUTINE
*****************************************************************
0ADB - 0AF3    CDBL(REG1) ROUTINE
*****************************************************************
0AF4 - 0B1E    SYSTEM MATH ROUTINE
*****************************************************************
0B1F - 0B25    SYSTEM MATH ROUTINE
*****************************************************************
0B26 - 0B36    FIX(REG1) ROUTINE
*****************************************************************
0B37 - 0B58    INT(REG1) ROUTINE
*****************************************************************
0B59 - 0B9F    SYSTEM MATH ROUTINE
*****************************************************************
0BA0 - 0BA9    SYSTEM MATH ROUTINE
*****************************************************************
0BAA - 0BC6    SYSTEM MATH ROUTINE
*****************************************************************
0BC7 - 0BD1    INTEGER SUBTRACTION  HL = DE - HL
*****************************************************************
0BD2 - 0BF1    INTEGER ADDITION  HL = DE + HL
*****************************************************************
0BF2 - 0C5A    INTEGER MULTIPLICATION  HL = DE * HL
*****************************************************************
0C5B - 0C6F    SYSTEM MATH ROUTINE
*****************************************************************
0C70 - 0C76    DOUBLE PRECISION SUBTRACTION REG1 = REG1 - REG2
*****************************************************************
0C77 - 0D32    DOUBLE PRECISION ADDITION REG1 = REG1 + REG2
*****************************************************************
0D33 - 0D44    SYSTEM MATH ROUTINE
```

*Map continues*

# What Does 80micro

the magazine for TRS-80* users

# Do For You?

**IT**

- provides you with more information on your TRS-80* than any other single source
- gives you 20–30 programs in each issue
- reviews equipment and software so you know what or what not to buy
- gives you the truth about the TRS-80—its good points and its limitations (*80 Micro* is not affiliated with Tandy)
- lets you save money—lots of it—by comparison shopping within the ad pages
- clues you in on how other TRS-80 owners are using and updating their systems
- tells you what's really happening in the industry
- brings you Wayne Green's outspoken and often controversial editorials every month
- and best of all it gives you a no risk subscription offer—

**Subscribe today**—if you are not satisfied with the first issue write "cancel" across your invoice and send it back. The *80 Micro* is yours to keep

## YES . . . I WANT 80 MICRO

☐ Check enclosed for $24.97

Bill my ☐ AE ☐ MC ☐ VISA ☐ Bill me

Name ____________________

Address ____________________

City ____________ State ______ Zip ______

Card# ____________ Expire Date ____________

Signature ____________ Interbank# ______ **32DR8**

Canada and Mexico $27.97, one year only, US funds. Foreign $44.97, one year only, US funds, DRAWN ON US BANK ONLY

**80 Micro ● PO Box 981 ● Farmingdale, N.Y. 11737**

***TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.**

*Map continued*

```
****************************************************************
0D45 - 0D56     SYSTEM MATH ROUTINE
****************************************************************
0D57 - 0D68     SYSTEM MATH ROUTINE
****************************************************************
0D69 - 0DA0     SYSTEM MATH ROUTINE
****************************************************************
0DA1 - 0DE4     DOUBLE PRECISION MULTIPLICATION
                REG1 = REG1 * REG2
****************************************************************
0DE5 - 0E4C     DOUBLE PRECISION DIVISION
                REG1 = REG1 / REG2
****************************************************************
0E4D - 0E64     SYSTEM MATH ROUTINE
****************************************************************
0E65 - 0E6B     CONVERT ASCII STRING TO DOUBLE PRECISION
                *ON ENTRY HL POINTS TO STRING
****************************************************************
0E6C - 0F09     CONVERT ASCII STRING TO MINUMUM REQUIRED
                *ON ENTRY HL POINTS TO STRING
****************************************************************
0F0A - 0F28     SYSTEM MATH ROUTINE
****************************************************************
0F29 - 0F93     SYSTEM MATH ROUTINE
****************************************************************
0F94 - 0FBC     SYSTEM MATH ROUTINE
****************************************************************
0FBD - 1033     CONVERT REG1 TO ASCII STRING
                *ON EXIT HL POINTS TO STRING
****************************************************************
1034 - 13E6     VARIOUS SYSTEM ROUTINES
****************************************************************
13E7 - 1438     SQR(REG1) ROUTINE
****************************************************************
1439 - 14C8     EXP(REG1) ROUTINE
****************************************************************
14C9 - 1540     RND(REG1) ROUTINE
****************************************************************
1541 - 1546     COS(REG1) ROUTINE
****************************************************************
1547 - 158A     SIN(REG1) ROUTINE
****************************************************************
158B - 15A7     SYSTEM MATH ROUTINE
****************************************************************
15A8 - 15BC     TAN(REG1) ROUTINE
****************************************************************
15BD - 15E2     ATN(REG1) ROUTINE
****************************************************************
15E3 - 1607     SYSTEM MATH ROUTINE
****************************************************************
1608 - 18F6     CONTAINS BASIC COMMAND ADDRESS LISTS,
                THE RESERVED WORD LIST, PRECEDENCE OPERATOR
                VALUES, ARITHMETIC ROUTINES LIST, DATA
                CONVERSION ROUTINES LIST, AND ERROR MESSAGE
                LIST

RESERVED WORD LIST:

ABS     D9 0977          GOSUB   91 1EB1          READ     88 21EF
AND     D2 25FD          GOTO    8D 1EC2          REM      93 1F07
ASC     F6 2A0F          IF      8F 2039          RESET    82 0138
ATN     E4 15BD          INKEY$  C9 019D          RESTORE  90 1D91
AUTO    B7 2008          INP     DB 2AEF          RESUME   9F 1FAF
CDBL    F1 0ADB          INPUT   89 219A          RETURN   92 1EDE
CHR$    F7 2A1F          INSTR   C5 419D          RIGHT$   F9 2A91
CINT    EF 0A7F          INT     D8 0B37          RND      DE 14C9
CLEAR   B8 1E7A          KILL    AA 4191          RSET     AC 419A
CLOAD   B9 2C1F          LEFT$   F8 2A61          RUN      8E 1EA3
CLOSE   A6 4185          LEN     F3 2A03          SAVE     AD 41A0
CLS     84 01C9          LET     8C 1F21          SET      83 0135
CMD     85 4173          LINE    9C 41A3          SGN      D7 098A
CONT    B3 1DE4          LIST    B4 2B2E          SIN      E2 1547
COS     E1 1541          LLIST   B5 2B29          SQR      CD 13E7
CSAVE   BA 2BF5          LOAD    A7 4188          STEP     CC 2B01
CSNG    F0 0AB1          LOC     EA 4164          STOP     94 1DA9
CVD     E8 415E          LOF     EB 4167          STR$     F4 2836
CVI     E6 4152          LOG     DF 0809          STRING$  C4 2A2F
CVS     E7 4158          LPRINT  AF 2067          SYSTEM   AE 02B2
DATA    88 1F05          LSET    AB 4197          TAB(     BC 2137
DEF     DD 4158          MEM     C8 27C9          TAN      E3 15A8
DEFDBL  9B 1E09          MERGE   A8 418B          THEN     CA
DEFINT  99 1E03          MID$    FA 2A9A          TIME$    C7 4176
DEFSNG  9A 1E06          MKD$    EE 4170          TO       BD
DEFSTR  98 1E00          MKI$    EC 416A          TROFF    97 1DF8
DELETE  B6 2BC6          MKS$    ED 416D          TRON     96 1DF7
DIM     8A 2608          NAME    A9 418E          USING    BF 2CBD
EDIT    9D 2E60          NEW     BB 1B49          USR      C1 27FE
ELSE    95 1F07          NEXT    87 22B6          VAL      FF 2AC5
END     80 1DAE          NOT     CB 25C4          VARPTR   C0 24EB
EOF     E9 4161          ON      A1 1FC6          +        CD 249F
ERL     C2 24DD          OPEN    A2 4179          -        CE 2532
ERR     C3 24CF          OR      D3 25F7          *        CF
ERROR   9E 1FF4          OUT     A0 2AFB          /        D0
EXP     E0 1439          PEEK    E5 2CAA                   D1
FIELD   A3 417C          POINT   C6 0132          >        D4
FIX     F2 0B26          POKE    B1 2CB1          =        D5
FN      BE 4155          POS     DC 27F5          <        D6
FOR     81 1CA1          PRINT   B2 206F          &        26 4194
FRE     DA 27D4          PUT     A5 4182          '  3A 93 FB
GET     A4 4174          RANDOM  86 01D3
****************************************************************
18F7 - 1904     DIVISION SUPPORT ROUTINE
****************************************************************
1905 - 191C     BASIC VARIABLES ROM STORAGE LOCATION
****************************************************************
191D - 1923     'ERROR' MESSAGE
****************************************************************
```

*Map continues*

*Map continued*

```
1924 - 1928    ' IN ' MESSAGE
****************************************************************
1929 - 192F    'READY' MESSAGE
****************************************************************
1930 - 1935    'BREAK' MESSAGE
****************************************************************
1936 - 1954    LOCATE FOR NEXT FRAME ON STACK ROUTINE
****************************************************************
1955 - 1962    MEMORY MANAGEMENT ROUTINE
****************************************************************
1963 - 197D    MEMORY MANAGEMENT ROUTINE
****************************************************************
197E - 1985    IF STILL IN INPUT PHASE THEN JUMP
1986 - 1989    CHECK RESUME
198A - 198D    IF NR ERROR THEN JUMP
198E - 1990    JUMP TO READY ROUTINE
1991 - 1996    SAVE LINE NUMBER FOR LAST DATA STATEMENT
1997 - 1999    ENTRY POINT FOR SN ERROR
199A - 199C    ENTRY POINT FOR /0 ERROR
199D - 199F    ENTRY POINT FOR NF ERROR
19A0 - 19A1    ENTRY POINT FOR RW ERROR
19A2 - 19AA    SAVE LINE NUMBER WITH ERROR TWICE
19AB - 19AD    GET RETURN ADDRESS AFTER REINITIALIZTION
19AE - 19B0    GET STACK ADDRESS
19B1 - 19B3    JUMP TO REINITIALIZE VARIABLES
19B4           ZERO BC
19B5 - 19B9    PUT ERROR NUMBER IN A AND C ALSO SAVE IT
19BA - 19BF    SAVE LAST BYTE EXECUTED ADDRESS
19C0           PUT IT IN DE
19C1 - 19C8    IF STILL IN INPUT PHASE THEN JUMP
19C9 - 19CB    SAVE ERROR ADDRESS
19CC           GET LAST BYTE EXECUTED ADDRESS
19CD - 19CF    SAVE IT
19D0 - 19DA    IF ON ERROR SET THEN JUMP
19DB - 19DD    IF NESTED RESUME THEN JUMP
19DE - 19E2    JUMP TO EXECUTION DRIVER FOR RESUME
19E3 - 19E4    ZERO ERROR OVERIDE FLAG
19E5           GET ERROR CODE
19E6 - 19E8    GOTO NEXT LINE ON VIDEO
19E9 - 19EB    POINT TO ERROR CODE TABLE
19EC - 19EE    CALL DOS EXIT AT 41A6
19EF           ZERO D
19F0 - 19F4    DISPLAY ?
19F5           ADJUST ERROR CODE POINTER
19F6 - 19FD    DISPLAY ERROR CODE
19FE - 1A01    POINT TO 'ERROR' MESSAGE
1A02 - 1A05    SAVE LINE NUMBER WITH ERROR
1A06 - 1A08    DISPLAY 'ERROR' MESSAGE
1A09           GET LINE NUMBER WITH ERROR
1A0A - 1A10    IF INPUT PHASE THEN JUMP
1A11 - 1A17    IF LINE NUMBER NON ZERO THEN DISPLAY
1A18           GET BC
1A19 - 1A1B    SET CURRENT OUTPUT DEVICE TO VIDEO
1A1C - 1A1E    CALL DOS EXIT AT 41AC
1A1F - 1A21    TURN OFF CASSETTE
1A22 - 1A24    GOTO NEXT LINE ON VIDEO
1A25 - 1A2A    DISPLAY 'READY' MESSAGE
1A2B - 1A32    IF SN ERROR JUMP TO EDIT MODE
1A33 - 1A35    GET CURRENT LINE NUMBER
1A36 - 1A3C    TEST FOR AUTO MODE
1A3D - 1A3E    IF NON AUTO THEN JUMP
1A3F - 1A41    GET CURRENT LINE NUMBER
1A42           SAVE IT
1A43 - 1A45    OUTPUT CURRENT LINE NUMBER
1A46 - 1A4A    SEARCH BASIC PROGRAM FOR CURRENT LINE NUMBER
1A4B - 1A53    IF MATCH FOUND DISPLAY * ELSE DISPLAY SPACE
1A54 - 1A56    GET INPUT
1A57           GET CURRENT LINE NUMBER
1A58 - 1A59    IF BREAK PRESSED THEN JUMP
1A5A - 1A5D    TURN OFF AUTO MODE
1A5E - 1A5F    IF NON AUTO THEN JUMP
1A60 - 1A75    FIGURE NEXT LINE NUMBER AND JUMP
****************************************************************
1A76 - 1A7A    DISPLAY >
1A7B - 1A7D    GET INPUT
1A7E - 1A80    IF BREAK PRESSED THEN JUMP
1A81 - 1A83    GET 1ST CHARACTER AND TEST FOR END OF STATEMENT
1A84 - 1A86    IF END THEN JUMP
1A87           SAVE LINE NUMBER
1A88 - 1A97    POINT TO 1ST CHARACTER AFTER LINE NUMBER
1A98           SAVE LINE NUMBER
1A99 - 1A9B    ENCODE INPUT INTO TOKENS
1A9C           GET LINE NUMBER
1A9D           GET FLAGS
1A9E - 1AA0    SAVE ENCODED STATEMENT POINTER
1AA1 - 1AA3    CALL DOS EXIT AT 41B2
1AA4 - 1AA6    IF INPUT PHASE THEN JUMP
1AA7           SAVE LINE NUMBER
1AA8           SAVE LINE LENGTH
1AA9           ZERO A
1AAA - 1AAC    SET FLAG TO INPUT PHASE
1AAD - 1AAF    GET 1ST CHARACTER, SET FLAGS, AND SAVE BOTH
1AB0           GET LINE NUMBER
1AB1 - 1AB3    SAVE IT
1AB4           GET LINE NUMBER
1AB5 - 1AB7    SEARCH FOR MATCHING LINE NUMBER IN PROGRAM
1AB8           SAVE MATCHING ADDRESS
1AB9 - 1ABB    SHIFT CLOSET LINE IF NO MATCH TO MAKE ROOM
1ABC           GET LINE ADDRESS
1ABD           GET FLAGS
1ABE           SAVE LINE ADDRESS
1ABF - 1AC0    IF MATCHING LINE NUMBER JUMP
1AC1           GET LINE ADDRESS
1AC2 - 1AC4    GET END OF BASIC PROGRAM
1AC5 - 1AC7    FIGURE NEW END OF BASIC PROGRAM
1AC8           SAVE IT
1AC9 - 1ACB    TEST OUT OF MEMORY
1ACC           GET END OF BASIC PROGRAM
1ACD - 1ACF    GET OLD END OF BASIL PROGRAM
1AD0           GET LINE ADDRESS
1AD1 - 1AE8    MOVE THE NEW LINE INTO MEMORY
1AE9 - 1AEB    ADJUST ALL THE LINE POINTERS
1AEC - 1AEE    CALL DOS EXIT AT 41B5
1AEF - 1AF1    UPDATE POINTERS AND VARIABLES
1AF2 - 1AF4    CALL DOS EXIT AT 41B8
1AF5 - 1AF7    GO GET NEW INPUT
****************************************************************
1AF8 - 1B0F    UPDATE LINE POINTERS ROUTINE
****************************************************************
1B10 - 1B2B    LINE NUMBER ROUTINE
****************************************************************
1B2C - 1B48    SEARCH FOR MATCHING LINE NUMBER ROUTINE
****************************************************************
1B49 - 1B5C    NEW ROUTINE

1B49           RETURN IF SN ERROR
1B4A - 1B4C    CLEAR SCREEN
1B4D - 1B4F    GET START OF BASIC PROGRAM
1B50 - 1B52    TROFF
1B53 - 1B55    AUTO OFF
1B56 - 1B58    ZERO 1ST TWO BYTES OF BASIC PROGRAM
1B59           POINT TO START OF VARIABLE TABLE
1B5A - 1B5C    SAVE IT
****************************************************************
```

*Map continues*

*Map continued*

```
****************************************************************
1B5D - 1BB2     RUN ROUTINE
1B5D - 1B5F     GET START OF BASIC PROGRAM
1B60            DEC BY 1
1B61 - 1B63     SAVE IT
1B64 - 1B6D     SET ALL VARIABLES TO SINGLE PRECISION
1B6E            ZERO A
1B6F - 1B71     SET NO ERROR FOR RESUME
1B72 - 1B73     ZERO HL
1B74 - 1B7F     ZERO BASIC POINTERS
1B80 - 1B82     RESTORE DATA
1B83 - 1B8B     INITIALIZE VARIABLES POINTERS
1B8C - 1B8E     CALL DOS EXIT AT 41B8
1B8F            GET RETURN ADDRESS
1B90 - 1B94     SET START OF STRING SPACE
1B95 - 1B9A     SET STACK POINTER
1B9B - 1B9E     INITIALIZE LITERAL STRING POOL
1BA1 - 1BA6     SET OUTPUT DEVICE TO VIDEO AND TURN CASSETTE
                OFF
1BA7            ZERO A
1BA8 - 1BA9     ZERO HL
1BAA - 1BAC     CLEAR FOR FLAG
1BAD            SIGNAL RUN MODE
1BAE            SET RETURN ADDRESS
1BAF - 1BB1     SAVE ENCODED STATEMENT POINTER
1BB2            RETURN
****************************************************************
1BB3 - 1BBF     INPUT ROUTINE

1BB3 - 1BB7     DISPLAY
1BB8 - 1BBC     DISPLAY SPACE
1BBD - 1BBF     JUMP TO INPUT ROUTINE
****************************************************************
1BC0 - 1C8F     TOKENIZE STATEMENT ROUTINE
****************************************************************
1C90 - 1C95     RST 18 CODE

1C90 - 1C91     H  D
1C92            RETURN IF NOT EQUAL
1C93 - 1C94     L  E
1C95            RETURN
****************************************************************
1C96 - 1CA0     RST 08 CODE

1C96            GET VALUE TO COMPARE
1C97            POINT TO VALUE TO COMPARE AND SAVE RETURN
                ADDRESS
1C98            COMPARE THE VALUES
1C99 - 1C9A     ADJUST RETURN ADDRESS AND SAVE IT
1C9B - 1C9D     IF NO SN ERROR THEN RST 10
1C9E - 1CA0     JUMP TO SN ERROR ROUTINE
****************************************************************
1CA1 - 1D1D     FOR ROUTINE
****************************************************************
1D1E - 1D77     BASIC INTERPERTER

1D1E - 1D21     CHECK FOR SHIFT @
1D22 - 1D24     IF SHIFT @ THEN CALL ROUTINE
1D25 - 1D27     SAVE ENCODED STATEMENT POINTER
1D28 - 1D2C     GET NEXT CHARACTER
1D2D - 1D30     IF MULTIPLE STATEMENT JUMP
1D31            CHECK FOR SN ERROR
1D32 - 1D34     IF THEN JUMP TO SN ERROR ROUTINE
1D35 - 1D38     CHECK FOR END OF BASIC PROGRAM
1D39 - 1D3B     IF THEN JUMP TO READY ROUTINE
1D3C - 1D43     SAVE LINE NUMBER
1D44 - 1D49     IF TROFF THEN JUMP
1D4A            SAVE DE
1D4B - 1D4F     DISPLAY <
1D50 - 1D52     CONVERT LINE NUMBER AND DISPLAY IT
1D53 - 1D55     DISPLAY
1D56            GET DE
1D59            GET ENCODED STATEMENT POINTER
1D5A            GET NEXT CHARACTER
1D5B - 1D5E     SET RETURN ADDRESS
1D5F            IF END OF STATEMENT RETURN
1D60 - 1D61     CHECK FOR TOKEN
1D62 - 1D64     IF NOT THEN JUMP
1D65 - 1D75     FIGURE JUMP
1D76            SAVE IT
1D77            GET ENCODED STATEMENT POINTER
****************************************************************
1D78 - 1D90     RST 10 CODE

1D78            INC ENCODED STATEMENT POINTER BY 1
1D79 - 1D7C     IF ALPHA RETURN
1D7D - 1D81     IF SPACE THEN LOOP
1D82 - 1D8F     SET FLAGS FOR OTHER CHARACTERS
1D90            RETURN
****************************************************************
1D91 - 1D9A     RESTORE ROUTINE
****************************************************************
1D9B - 1DA8     KEYBOARD ROUTINE

1D9B - 1D9D     SCAN KEYBOARD
1D9E - 1D9F     IF NO KEY PRESSED RETURN
1DA0 - 1DA4     IF SHIFT @ WAIT TILL KEY PRESSED
1DA5 - 1DA7     SAVE KEY
1DA8            ADJUST KEY
****************************************************************
1DA9 - 1DAD     STOP ROUTINE
****************************************************************
1DAE - 1DE3     END ROUTINE
****************************************************************
1DE4 - 1DF6     CONT ROUTINE
****************************************************************
1DF7 - 1DFF     TRON ROUTINE
****************************************************************
1DF8 - 1DFF     TROFF ROUTINE
****************************************************************
```

```
1E00 - 1E3C     DEFSTR ROUTINE
****************************************************************
1E03 - 1E3C     DEFINT ROUTINE
****************************************************************
1E06 - 1E3C     DEFSNG ROUTINE
****************************************************************
1E09 - 1E3C     DEFDBL ROUTINE
****************************************************************
1E3D - 1E44     SYSTEM ROUTINE
****************************************************************
1E45 - 1E4D     SYSTEM ROUTINE
****************************************************************
1E4F - 1E79     ASCII TO BINARY ROUTINE
****************************************************************
1E7A - 1EA2     CLEAR ROUTINE
****************************************************************
1EA3 - 1EB0     RUN ROUTINE
****************************************************************
1EB1 - 1EC1     GOSUB ROUTINE
****************************************************************
1EC2 - 1EDD     GOTO ROUTINE
****************************************************************
1EDE - 1F04     RETURN ROUTINE
****************************************************************
1F05 - 1F1F     DATA ROUTINE
****************************************************************
1F21 - 1F6B     LET ROUTINE
****************************************************************
1F6C - 1FAE     ON ROUTINE
****************************************************************
1FAF - 1FF3     RESUME ROUTINE
****************************************************************
1FF4 - 2007     ERROR ROUTINE
****************************************************************
2008 - 2038     AUTO ROUTINE
****************************************************************
2039 - 2066     IF ROUTINE
****************************************************************
2067 - 206E     LPRINT ROUTINE
****************************************************************
206F - 2166     PRINT ROUTINE
****************************************************************
2169 - 217      SET OUTPUT DEVICE TO VIDEO ROUTINE
****************************************************************
2178 - 217E     'REDO' MESSAGE
****************************************************************
217F - 2199     REDO ROUTINE
****************************************************************
219A - 21EE     INPUT ROUTINE
****************************************************************
21EF - 2285     READ ROUTINE
****************************************************************
2286 - 2295     'EXTRA IGNORED' MESSAGE
****************************************************************
2296 - 22B5     DATA ROUTINE
****************************************************************
22B6 - 2323     NEXT ROUTINE
****************************************************************
2324 - 27C8     EXPRESSION EVALUATION ROUTINES
****************************************************************
27C9 - 27D3     MEM ROUTINE
****************************************************************
27D4 - 27F4     FRE ROUTINE
****************************************************************
27F5 - 27FD     POS ROUTINE
****************************************************************
27FE - 2818     USR ROUTINE
****************************************************************
2819 - 2A2E     EXPRESSION EVALUATION ROUTINES
****************************************************************
2A2F - 2A60     STRING$ ROUTINE
****************************************************************
2A61 - 2A90     LEFT$ ROUTINE
****************************************************************
2A91 - 2A99     RIGHT$ ROUTINE
****************************************************************
2A9A - 2AC4     MID$ ROUTINE
****************************************************************
2AC5 - 2ADE     VAL ROUTINE
****************************************************************
2ADF - 2AE6     SYNTAX CHECK ROUTINE
****************************************************************
2AE7 - 2AEE     DISK BASIC ROUTINE
****************************************************************
2AEF - 2AFA     INP ROUTINE
****************************************************************
2AFB - 2B00     OUT ROUTINE
****************************************************************
2B01 - 2B28     EVALUATE EXPRESSION ROUTINES
****************************************************************
2B29 - 2B2D     LLIST ROUTINE
****************************************************************
2B2E - 2B74     LIST ROUTINE
****************************************************************
2B75 - 2B7D     OUTPUT ROUTINE
                *OUTPUT MESSAGE POINTED TO BY HL UNTIL 0
                IS FOUND
****************************************************************
2B7E - 2BC5     DECODE ENCODED STATEMENT ROUTINE
****************************************************************
2BC6 - 2BF4     DELETE ROUTINE
****************************************************************
2BF5 - 2C1E     CSAVE ROUTINE
****************************************************************
2C1F - 2CA4     CLOAD ROUTINE
****************************************************************
2CA5 - 2CA9     'BAD' MESSAGE
****************************************************************
2CAA - 2CB0     PEEK ROUTINE
****************************************************************
```

*Map continues*

See List of Advertisers on Page 451

*Map continued*

```
2CB1 - 2CBC    POKE ROUTINE
******************************************************************
2CBD - 2E52    PRINTUSING ROUTINE
******************************************************************
2E53 - 2E5F    LINE NUMBER ROUTINE
******************************************************************
2E60 - 2FFA    EDIT ROUTINE
******************************************************************
2FFB - 2FFF    NOTHING HERE
******************************************************************
3000 - 37DD    RESERVED MEMORY
******************************************************************
37DE - 37FF    MEMORY MAPPED INPUT/OUTPUT

37DE           DOS COMMUNICATION STATUS ADDRESS
37DF           DOS COMMUNICATION DATA ADDRESS
37E0           INTERRUPT LATCH ADDRESS
37E1           DISK DRIVE SELECT LATCH ADDRESS
37E2           CASSETTE DRIVE LATCH ADDRESS
37E8           PRINTER PORT ADDRESS
37EC           FLOPPY DISK CONTROLLER ADDRESS
KEYBOARD MEMORY

BIT     0    1    2    3    4    5    6    7
3801    @    A    B    C    D    E    F    G
3802    H    I    J    K    L    M    N    O
3804    P    Q    R    S    T    U    V    W
3808    X    Y    Z
3810    0    1    2    3    4    5    6    7
3820    8    9    :    ;    ,    -    .    /
3840    CR   CLS  BRK  UA   DA   LA   RA   SP
3880    SHIFT
******************************************************************
4000 - 4014    RST JUMP VECTORS

4000 - 4002    RST 08
4003 - 4005    RST 10
4006 - 4008    RST 18
4009 - 400B    RST 20
400C   400E    RST 28
400F - 4011    RST 30
4012 - 4014    RST 38
******************************************************************
4015 - 401C    KEYBOARD DEVICE CONTROL BLOCK

4015           DEVICE TYPE
4016 - 4017    DRIVER ADDRESS
4018 - 401A    NOT USED
401B - 401C    RAM BUFFER ADDRESS
******************************************************************
401D - 4024    VIDEO DEVICE CONTROL BLOCK

401D           DEVICE TYPE
401E - 401F    DRIVER ADDRESS
4020 - 4021    CURSOR POSITION
4022           CURSOR ON/OFF FLAG
4023           CURSOR CHARACTER
4024           NOT USED
******************************************************************
4025 - 402C    PRINTER DEVICE CONTROL BLOCK

4025           DEVICE TYPE
4026 - 4027    DRIVER ADDRESS
4028           LINES PER PAGE
4029           LINES PRINTED SO FAR
402A           NOT USED
402B - 402C    RAM BUFFER ADDRESS
******************************************************************
402D - 4152    COMMUNICATIONS REGION

402D - 407F    USED BY DOS
4080 - 408D    DIVISION SUPPORT ROUTINE
408E - 408F    USER ADDRESS
4090 - 4092    RANDOM NUMBER SEED
4093 - 4095    IN A,0
4096 - 4098    OUT A,0
4099           LAST KEY PRESSED AFTER BREAK
409A           USED FOR RESUME
409B           NUMBER OF CHARACTERS IN PRINT LINE
409C           DEVICE TYPE FLAG
               -1 CASSETTE
                0 VIDEO
               +1 PRINTER
409D           SIZE OF VIDEO LINE
409E           SIZE OF PRINTER LINE
409F           RESERVED
40A0 - 40A1    START OF STRING SPACE POINTER
40A2 - 40A3    CURRENT LINE NUMBER
40A4 - 40A5    START OF BASIC PROGRAM POINTER
40A6           CURSOR POSITION
40A7 - 40A8    KEYBOARD BUFFER ADDRESS
40A9           CASSETTE INPUT FLAG
               0 IF CASSETTE ELSE NON ZERO
40AA - 40AD    RANDOM NUMBER SEED
40AE           VARIABLE FLAG
40AF           NUMBER TYPE FLAG
               2 INTEGER
               3 STRING
               4 SINGLE PRECISION
               8 DOUBLE PRECISION
40B0           USED DURING EXPRESSION EVALUATION
40B1 - 40B2    MEMORY SIZE POINTER
40B3 - 40B4    NEXT LOCATION AVAILABLE IN LITERAL STRING POOL
40B5 - 40D2    LITERAL STRING POOL
40D3 - 40D5    USED FOR STRINGS
40D6 - 40D7    NEXT LOCATION AVAILABLE IN STRING SPACE
40D8 - 40D9    MULTIPLE USES
```

*Map continues*

*Map continued*

```
4ODA - 40DB    LAST DATA STATEMENT READ LINE NUMBER
40DC           FOR FLAG
               1 IN PROGRESS
               0 NOT IN PROGRESS
40DD           USED DURING INPUT PHASE
40DE           MULTIPLE USES
40DF - 40E0    USED BY DOS
40E1           AUTO FLAG
               0 NO AUTO
40E2 - 40E3    CURRENT LINE NUMBER
40E4   40E5    AUTO INCREMENT
40E6   40E7    MULTIPLE USES
40E8   40E9    ENCODED STATEMENT POINTER
40EA - 40EB    ERROR LINE NUMBER
40EC           USED BY ERROR
40ED - 40EE    LAST BYTE EXECUTED IN ERROR LINE
40EF           POSITION OF ERROR IN LINE
40F0   40F1    ON ERROR POINTER
40F2           ERROR FLAG
40F3   40F4    MULTIPLE USES
40F5 - 40F6    LAST LINE RUN
40F7 - 40F8    USED BY ERROR
40F9 - 40FA    VARIABLES POINTER
40FB - 40FC    ARRAY POINTER
40FD - 40FE    START OF FREE SPACE
40FF - 4100    USED BY READ
4101 - 411A    VARIABLE DECLARATION TABLE
411B           TRACE FLAG
               0 TROFF
               NON ZERO TRON
411C           USED BY MATH ROUTINES
411D - 4124    REG1
               INTEGER            SINGLE             DOUBLE
411D                                                 LSB
411E                                                 LSB
411F                                                 LSB
4120                                                 LSB
4121           LSB                LSB                LSB
4122           MSB                LSB                LSB
4123                              MSB                MSB
4124                              EXP                EXP
4125 - 4126    USED BY MATH ROUTINES
4127 - 412E    REG2 SAME AS REG1
412F           NOT USED
4130 - 4149    PRINT BUFFER
414A - 4151    USED BY MATH ROUTINES
************************************************************
4152 - 41A5    DISK BASIC LINKS

4152 - 4154    CVI
4155 - 4157    FN
4158 - 415A    CVS
415B - 415D    DEF
415E - 4160    CVD
4161 - 4163    EOF
4164 - 4166    LOC
4167 - 4169    LOF
416A - 416C    MKI$
416D - 416F    MKS$
4170 - 4172    MKD$
4173 - 4175    CMD
4176 - 4178    TIME$
4179 - 417B    OPEN
417C - 417E    FIELD
417F - 4181    GET
4182 - 4184    PUT
4185 - 4187    CLOSE
4188 - 418A    LOAD
418B - 418D    MERGE
418E - 4190    NAME
4191 - 4193    KILL
4194 - 4196    &
4197 - 4199    LSET
419A - 419C    RSET
419D - 419F    INSTR
41A0 - 41A2    SAVE
41A3 - 41A5    LINE
************************************************************
41A6 - 41E4    DOS LINKS

               CALLED FROM:
41A6 - 41A8    19EC
41A9 - 41AB    27FE
41AC - 41AE    1A1C
41AF - 41B1    0368
41B2 - 41B4    1AA1
41B5 - 41B7    1AEC
41B8 - 41BA    1AF2
41BB - 41BD    1B8C
41BE - 41C0    2174
41C1   41C3    032C
41C4 - 41C6    0358
41C7 - 41C9    1EA6
41CA - 41CC    206F
41CD - 41CF    20C6
41D0 - 41D2    2103
41D3 - 41D5    2108
41D6 - 41D8    219E
41DC - 41DE    222D
41DF - 41E1    2B44
41E2 - 41E4    02B2
************************************************************
41E5 - 41E7    USED BY INPUT
************************************************************
41E8 - 42E7    INPUT BUFFER
************************************************************
42E8           ZERO
************************************************************
42E9           START OF BASIC PROGRAM MEMORY
************************************************************
```





✓ See List of Advertisers on Page 451

*Actuators and energy translation.*

# Cybernetics—Part III

*Stephen Davids, President*
*Robot Research and Development, Inc.*
*Box 541 Station Z*
*Toronto, Ontario*
*Canada M5N 2Z6*

In my previous two articles, I discussed the overall notion of a robot system and the specifics of sensors, as well as their interface to the TRS-80. This article covers actuators or output devices: their nature; modes of operation; and interface to the TRS-80. I will expand the bibliography started in Part Two of this series.

An actuator manipulates or alters the environment. The muscular system is the human's main output device, just as the nervous system is the main input device. The human system is effective but severely limited; it is capable of only sound, heat and motion output. Robots may emit light or other forms of electromagnetic radiation, or extend the range of motion, strength, or precision of the human form.

**Energy Translation**

All output from any microcomputer is in the form of electrical energy. The biological world does not generally make direct use of this electrical energy; we must first translate this energy into a more useful form (motion, light, heat, or sound).

The standard way to accomplish this translation is to interface the weak electrical output signals of the TRS-80 to an appropriate actuator giving the TRS-80 control of the output device as well as feedback on the performance of its commands.

This is a four part operation. First the TRS-80 must identify that it requires an output. During program execution, the TRS-80 encounters a conditional statement branching to an output subroutine. For example, a program input loop constantly monitors an infrared light sensitive detector diode on the robot's forehead. All night the robot waits for a high at that port address while it charges batteries. In the morning, as the first rays of light signal the start of a sunny day, the sensor drives the input port high. The program branches to the first line of an output subroutine.

The second part of this operation is the selection of an output device. Continuing the example, the robot now decides, on the basis of all information available to its sensors and the program it has been given by its creator, what to do now that the sun has arisen. Assume the robot is plugged into a wall outlet to charge. Before going about its daily chores, it must disconnect itself from the power supply. I designed my robot outlet to greatly facilitate the plugging/unplugging operation. The robot retracts its connector to free itself and terminate the charging process. This is accomplished by a single solenoid, related to port address 05. The program determines that port address 05 is to be activated and moves to the next part of the operation.

> *"This system, called distributed processing, is on the very frontier of cybernetics."*

The third part of the output operation is simple for our example, but could be complex for an analog output routine. The program must determine the correct value to be output to the actuator. The TRS-80 can drive the selected output high or low with respect to a reference voltage also on the output bus. The decoding circuit or interface must determine the state of the output port and act accordingly.

For example, if the actuator is a solenoid as above, the program could drive port address 05 high to signal the solenoid to pull the plug out from the wall.

The fourth part of the output operation is feedback. In my first article I stressed that a robot system without feedback is inherently unstable. In our example the system is unstable because the robot does not yet know that the plug has been extracted from the wall. A true servosystem compares the existing state to the desired state. The robot must determine if the cord is still attached to the wall. This may be done in many ways, such as checking the charging circuit to see if a current is still flowing, or trying to close a small trap door that contains the cord and the insert/withdrawal solenoid. If the door cannot close or the circuit is still

# TAS Adventure Series Package I

**By: Gary Heimburger and Gary Robinson**

(This adventure package is the first to be marketed that was created by The Adventure System! Look for more fantastic adventures coming soon! Each TAS adventure package includes TWO complete adventures for one low price!)

### Adventure 1: Isles of Doom - Part I

The Scenario: A small South-sea island chain. Unfortunate navigation left you stranded on one island.

The Mission: (if you can get through the shark-infested waters to the island!) The island was a scene of activity during WWII. Scattered over the island are components that will allow you to fix an old radio transmitter. Can you fix it and tap out your SOS before falling prey to the elements of doom?

### Adventure 2: Lost City of Gold

The Scenario: You're on a mountain climbing expedition in South America (in Peru, near Lake Titicaca). After straying from the beaten path, you discover a cave that eventually leads to a forgotten part of the lost Incan city. Grab your pitons.

The Mission: There are plenty of treasures if you can safely collect them and retrace your steps to the simple mountain hut. Be forwarned that most Incan cities contain many perils, especially for the uninitiated...seems Incan priests were rather paranoid. Requires 32K.

**TAS Adventure Packages are $15.95 on tape and $19.95 on disk. All media is unprotected! TAS Adventures allow you to SAVE up to 10 versions of the game in progress! Please specify Model I or III.**

•••

**Get in on the action! The Adventure System editor is now completely in Z80 and allows easy interactive debugging. Requires 48K. Specify Tape or Disk, please. Only $39.95 for the complete system AND three adventures!**

# LDOS Users!

A special utility package just for YOU:

**DRIVE-5**

Four drives not enough? DRIVE-5 is a memory based disk emulator which allows you to configure a fifth drive IN MEMORY! No moving parts, so access is FAST! In a two drive system, FORMAT and BACKUP can be placed on DRIVE-5 (with /SYS modules) to allow multiple backups with less disk swapping.

**LOWDUMP**

LOWDUMP allows you to DUMP files from anywhere in memory, including ROM and video RAM! LOWDUMP allows all parameters as the LDOS DUMP command, including ASCII, for dumping files without loader information present.

**SYSGEN**

SYSGEN/CMD allows you to save one or more SYSGEN files as normal /CMD files. Keep several configurations on the same disk and reload as necessary by simply typing the filename. Especially useful when the preferred configuration consumes so much memory that large programs will not execute properly. SYSGEN/CMD may also be executed from a running BASIC program. When the resultant /CMD file is executed, the BASIC program will continue with the next statement, all program variables intact!

**UNREPAIR**

Model I TRSDOS cannot read LDOS disks due to the F8 DAM used for directory sectors. UNREPAIR will rewrite the directory track of a single density disk using the FA DAM that will be recognized by Model I TRSDOS. UNREPAIR should run with any Model I DOS and the Radio Shack E/I.

An extra BONUS! The TAS LDOS Utility pack includes Z80 source code (on disk) of most of the above utilities (the most informative). The LDOS Utility Pak is $49.95. This package was written by Les Mikesell.

# BRAND NEW!


**By Bruce Hansen**

An exciting new arcade game! Threshold-Nine is easy, but only the experienced reach Threshold-Zero. As you progress to the faster levels, the playing field decreases! Excellent graphics and sound, joystick compatible, all Z80 programming! Supplied in a "back-uppable" format (disk users will need to copy to a TRSDOS system diskette). High scores are saved on the disk version. Includes a special "pause" feature (for nature calls and other interrupt processing). Works on Models I and III cassette ($15.95) and disk ($19.95).

FREE to the first 100 customers who order Threshold-Zero: A mini-poster of the Threshold-Zero Cover Art! Order today and find out why Threshold-Zero is already one of our best selling games!

THE ALTERNATE SOURCE

69

# STOPPER!

## The BASIC Breakpointer
## By Roxton Baker
### Author of "TRAKCESS"

Stopper is a unique machine language utility that allows interactive program debugging when using Level II or Disk BASIC on the Model I and III. Stopper was designed by Roxton to help debug a new version of TRAKCESS that will be announced shortly, because there wasn't anything on the market that would provide the features needed.

Some of Stopper's commands: Set a breakpoint with a "hit" count, either when a line has been executed "n" times or a variable is equal a certain value; or not equal a certain value; breakpoints and variables may be viewed and modified at any time; selective execution of program statements and lines; display only the portion of the line where the error occurred; TRON functions are improved and may be routed to printer; set a "tolerance" on single and double precision numbers (to accommodate the inaccurate ROM routines and more. Naturally single stepping is fully supported.

With STOPPER you will know exactly where the error is and why it occurred. Program execution is under YOUR control!

A special offer from the author and TAS: Stopper is $20 on tape or disk, BUT you are licensed to make up to four additional copies, making the cost for this valuable utility only $4. Support co-op purchases by ordering today. BASIC debugging will never be easier, nor a better utility any cheaper. Extra manuals and Z80 source code are available.

# EFFECTIVE COMMUNICATION

is easy with Modem80. There is absolutely no need to pay any more for a top-quality communication package. Modem80 can help your TRS-80 Model I and III talk to a wide variety of machines. The Modem80 translation table is easily modified to accommodate unusual protocol. Modem80 allows file uploading and downloading. XON and XOFF are supported to allow transmission of files larger than memory. A special protocol (compatible with the CP/M public domain program MODEM) allows error-free transmission without converting to ASCII. XMODEM, a special version of MODEM is included so that you may give your friends a copy to take advantage of faster error-free transmission. A host program is included at no extra charge to allow accessing your unattended TRS-80 from a remote location. All "local" parameters (baud rate, stop bits, parity, duplex and more) can be changed quickly and easily with a minimum of keystrokes. For specialized applications, one file may be transmitted while a different file is simultaneously being received. DOS commands may be executed while within Modem80. Special patches insure compatibility with all popular Model I and III operating systems. The previous screen is restored after DOS commands. Modem80 also allows disabling screen scroll to permit transmission speeds at up to 9600 baud (under direct connect situations). A special patch sheet is available for people wanting to use the Micronet Dow Jones service. Also available, on request, is a LYNX modem version that allows operation on a Model III even if RS-232 is installed. All this and many more features are what's making Modem80 one of our most popular programs! Oh, yes, the price helps too: $39.95 complete including documentation in a three-ring binder. Modem80 is authored by Les Mikesell.

# Order Form

**Please send me the following:**

| | |
|---|---|
| ☐ TAS Adventure Package I (tape) | $15.95 |
| ☐ TAS Adventure Package I (disk) | $19.95 |
| ☐ The Adventure System (tape) | $39.95 |
| ☐ The Adventure System (disk) | $39.95 |
| ☐ LDOS Enhancement Package | $49.95 |
| ☐ Threshold-Zero (tape) | $15.95 |
| ☐ Threshold-Zero (disk) | $19.95 |
| ☐ Stopper | $20.00 |
| ☐ Modem80 | $39.95 |

☐ **TAS Subscription .. $24.00**

**Please add $3 to your total order for shipping (except TAS subscribers). We ship UPS unless you specify otherwise.**

Name: ____________________

Address: ____________________

Address: ____________________

City: ____________________

State, Zip: ____________________

**Mail order to:**

The Alternate Source
704 N. Pennsylvania Ave.
Lansing, Michigan 48906
Ph. (800) 248-0284
Ph. (517) 482-8270

Master Card, Visa and COD ($1.75 extra) are welcome!

See List of Advertisers on Page 451

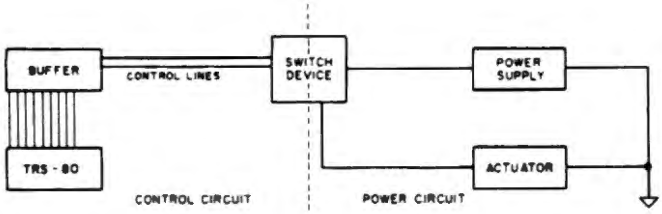



*Fig. 1*

functional, the program should jump to an error correction subroutine or try to extract the plug once more.

**Circuits**

In any device that comprises both decision logic and working circuits, there are voltages and currents potentially harmful to the more sensitive electronic parts. Actuators amplify the output signal of the microcomputer from logic levels to useful working levels. These working levels may range from 5 volts at several hundred milliamps to hundreds of volts at 40 or 50 amperes. Though the latter conditions are rare, it is common to switch 12 volts dc at 5 or 10 amperes, enough to run small and medium dc motors. These conditions are sufficient to fry delicate electronic equipment like the TRS-80. With this in mind, before you hook up your TRS-80 to the vacuum cleaner, let's discuss some basic robotics.

There are two subcircuits in a robot output device. The *control* circuit includes the first output stage and the decision-base, in our case the TRS-80. The control circuit controls the behavior of the rest of the output device using the logic level voltages I have mentioned.

The *power* circuit is the business end of the system. The actuators and all other output devices are part of these circuits. A power circuit includes a power source and an actuator or effector. This circuit is a slave circuit, dependent upon the control circuit for commands (see Fig. 1).

When the control circuit receives an output command in the form of a varying voltage relative to an acknowledged ground, the command simultaneously appears in the power circuit to be amplified and interpreted by the whole process. The end result is a change in the actuator status.

Returning to my example, the small electrical signals on the output bus of the TRS-80 cannot operate the solenoid directly. This is where the power stage comes in. A 12 volt dc power supply forms a circuit with the solenoid and some type of switching device. When the switch is closed, current flows through the solenoid coil pulling the metal plunger due to magnetic action similar to a speaker cone (see Figs. 2 and 3). The plunger is attached to the plug leading to the wall outlet so it extends when the solenoid is activated and is retracted by the action of an integral spring when the power circuit is open and the solenoid deactivated (see Fig. 4).

The voltage and current operating the solenoid are typical of levels dangerous to the TRS-80. An excellent design practice is to opto-isolate the control and power circuits (Fig. 5). An opto-isolator is an infrared LED emitter/detector pair in a single IC package. The theory of operation is simple. The LED emitter, part of the control circuit, is switched on and off by the buffered output from the TRS-80 at the expansion port. The detec-

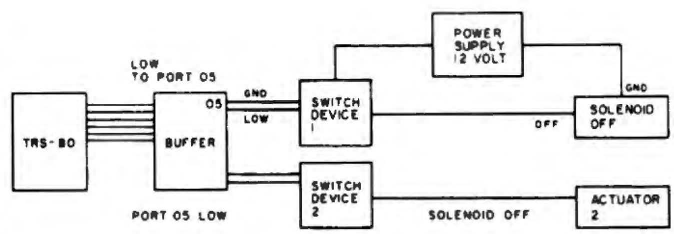



*Fig. 2*

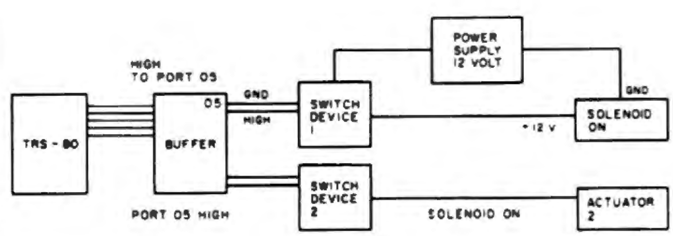



*Fig. 3*

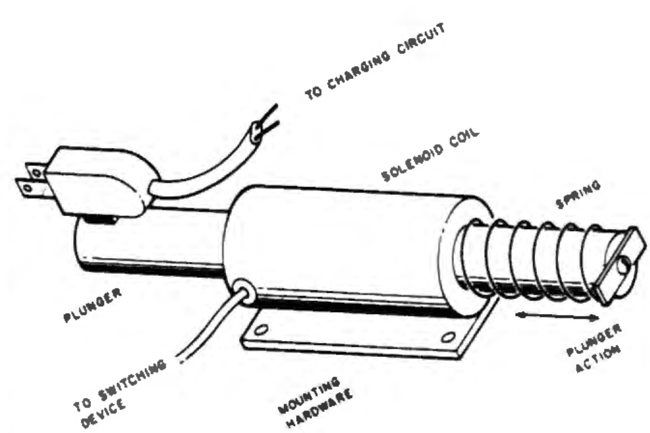



*Fig. 4*

tor acts as a switch electrically isolated from the emitter to about 3750 volts. When the emitter diode is on, the detector phototransistor turns on, allowing current flow to the actuator (see Fig. 6). If anything ever goes wrong electrically, the most expensive component, the TRS-80, is not destroyed.

I have assumed that the robot you are building has electrical actuators or effectors (these include motors, solenoids, lights, and amplifiers). Advantages to hydraulic and pneumatic systems are increased strength and speed. Indeed, most industrial robots are hydraulic at this time.

In a hydraulic or pneumatic robot the power circuit usually includes a control valve rather than the actuator itself. The best of these, called servovalves, monitor precisely the real state of the actuator through optical or electromagnetic transducer feedback loops. These feedback devices provide a very high precision control of the effector and an excellent repeatability of operation. This robot is excessively expensive to design and build and is outside the scope of this series of articles.

The motor is probably the most useful actuator available to the robot builder. When inter-

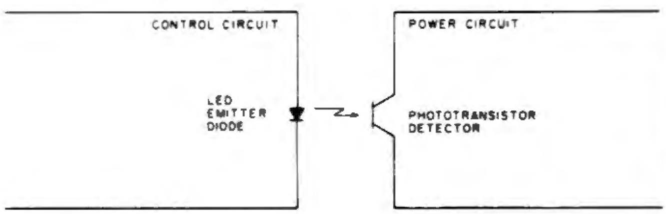



*Fig. 5*

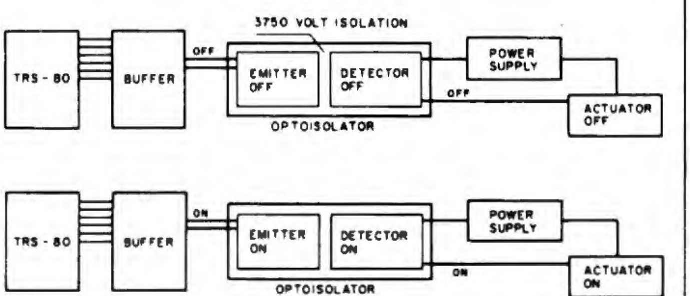



*Fig. 6*

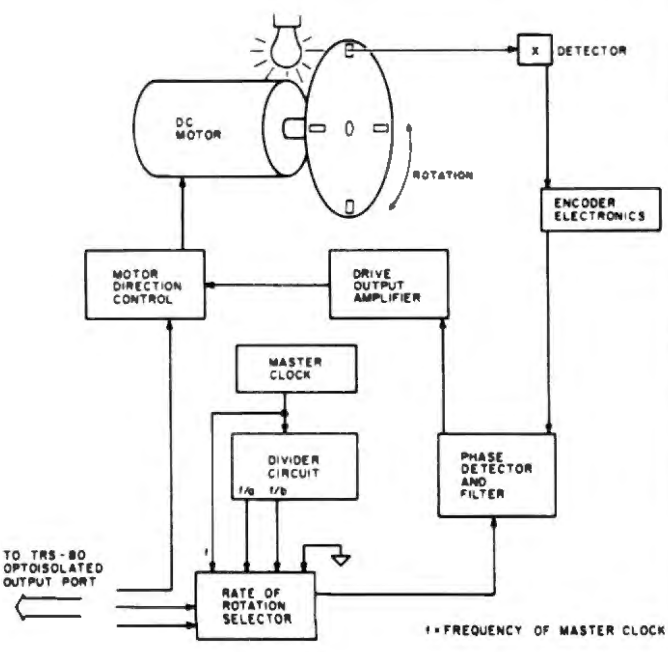



*Fig. 7*



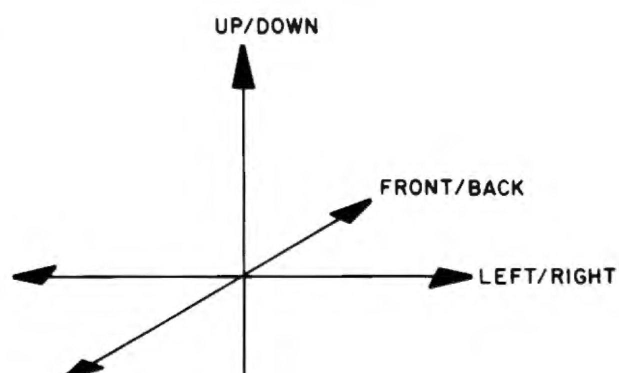
✓See List of Advertisers on Page 451

facing a motor to the TRS-80 first consider what function that motor is to serve. This will help decide the type and size of motor and speed and voltage requirements. Sometimes the controlling speed is critical, as in the main drive motors responsible for propulsion. You can build a phase locked servosystem that contains feedback from a slotted disk rotating in synchronization with the motor output shaft (see Fig. 7). For a more complete discussion of phase locked servosystems, refer to the list of references.

**TRS-80 Limitations**

We must face the fact that the TRS-80 is not the fastest microcomputer designed; while it is functional as a robot brain, we should not challenge it with too many important tasks at any one instant. Clever programming of the TRS-80 allows a small number of simultaneous motions such as traversing the living room floor while playing the national anthem and waving a small flag. The programmer must develop an internal list of priorities to ensure that the main processor is mainly reading inputs rather than controlling the behavior of the outputs. One way to achieve this is to put the burden of counting repetitive motions or monitoring speed on an external circuit. The TRS-80 signals the external circuit, called an output controller, as to the required behavior of its actuator. The output controller directly drives the actuator and monitors the actuator's behavior, looking for the completion of the task assigned by the TRS-80 or a failure. In either case the output controller signals the TRS-80 and awaits the next instruction. This way the TRS-80 has much less to do with each actuator, and can handle a greater number of tasks than if it directly controlled every function. This system, called distributed processing, is on the very frontier of cybernetics.

Another method of reducing the demand placed on the TRS-80 by complex motion of an actuator such as a manipulator arm is to reduce the required motion into three single axis motion vectors. If an arm manipulator (actuator) must move from a start position to a finish position possessing different coordinates in three dimensional space, the path trajectory requires an integration formula that would weaken the knees of the brightest mathematical geniuses. Most robot builders would consider early retirement. The solution is to consider each motion as being composed of three separate straight line motions: left or right; forward or back; and up or down (Fig. 8). To truly simplify the motion, we can design a manipulator on these same straight lines




VECTOR COMPONENTS OF THREE DIMENSIONAL MOTION

*Fig. 8*

# Heart of TEXAS COMPUTER SYSTEMS

**TCS**

## LEADER in MAIL ORDER DISCOUNTS!

**800 433-5184**

**Texas 817/274-5625**

## TANDON DRIVES

**BARE DRIVES ONLY**

| | |
|---|---|
| 40 track single sided | $199 |
| 80 track (dual sided 40) | $269 |
| 160 track (dual sided 80) | $389 |
| 5 ¼ inch THINLINE | $Call |
| 8 inch THINLINE | $Call |
| Winchester Hard Drive | $Call |

WE WILL NOT BE UNDERSOLD!

**1 DRIVE / Single Cabinet**

| | |
|---|---|
| 40 track single sided | $249 |
| 80 track (dual sided 40 track) | $329 |
| 160 track (dual sided 80 track) | $449 |

**1 DRIVE / Double Cabinet**

| | |
|---|---|
| 40 track single sided | $289 |
| 80 track (dual sided 40 track) | $369 |
| 160 track (dual sided 80 track) | $499 |

**2 DRIVES / Double Cabinet**

| | |
|---|---|
| 40 track single sided | $489 |
| 80 track (dual sided 40 track) | $639 |
| 160 track (dual sided 80 track) | $849 |

Drives in cabinets come assembled/tested with power supply. Order cable separately.

## CORVUS HARD DISK

**CORVUS HARD DISKS complete from $2695**

Add 5, 10 or 20 megabytes of storage to your TRW80, Apple, Atari, Heath, Zenith, IBM, Intertec, S-100 and many others. One or several computers can share a hard disk. Get simultaneous access to data for multiple users. Available now at SUPER SAVINGS.

## PRINTERS

TCS has the LOWEST PRICES on IN STOCK PRINTERS!

**MATRIX PRINTERS**

| | |
|---|---|
| STAR MICRONICS | $399 |
| NEC 8023A | $495 |
| C. ITOH 8510 / TEC | $479 |
| ANADEX 9501 w/2k buffer | $1295 |
| LINE PRINTER (RS) | $Call |
| LINE PRINTER VI (RS) | $988 |
| LINE PRINTER VII (RS) | $649 |
| CENTRONICS 352 (200 cps) | $1795 |
| CENTRONICS 353 (dual mode) | $2495 |
| OKIDATA 82A | $429 |
| OKIDATA 82A w/tractor | $479 |
| OKIDATA 83A | $685 |
| OKIDATA 84A (serial) | $1169 |
| OKIDATA 84A (parallel) | $1069 |

**HI SPEED BAND PRINTERS**

| | |
|---|---|
| CENTRONICS 6080 (standard) | $CALL |
| CENTRONICS (quietized) | $CALL |

**LETTER QUALITY PRINTERS**

| | |
|---|---|
| C.ITOH F-10 (40 cps) | $1595 |
| DAISY WHEEL II (RS) | $1695 |
| SMITH CORONA TP-1 | $699 |
| BROTHER / COMREX | $829 |
| NEC 7730 | $2375 |
| NEC 3530 | $1795 |

**EPSON PRINTERS**

| | |
|---|---|
| EPSON MX80 | $449 |
| EPSON MX80 FT | $524 |
| EPSON MX100 FT | $685 |

GRAPHTRAX PLUS comes free in Epsons!!

**EPSON ACCESSORIES**

| | |
|---|---|
| GRAPHTRAX PLUS (bit image) | $79 |
| MODEL I interface | $29 |
| MODEL I interface/cable to keyboard | $85 |
| MODEL III cable | $29 |
| COLOR Computer interface/cable | $155 |
| APPLE interface/cable | $69 |
| ATARI cable (must have 850 int) | $36 |
| IBM Personal Computer cable | $45 |
| OSBORNE cable | $36 |
| Serial RS232 2k buffered int card | $109 |

**EPSON PACK II**

Screen oriented bit image graphics generator for the Model III. Create your own characters, symbols, etc. Documentation.

| | |
|---|---|
| EPSON PACK II bought w/printer | $24.95 |
| EPSON PACK II bought seperately | $34.95 |
| Update for EPSON PACK I | $7.50 |

## B-TREE CUSTOM SOFTWARE FROM TCS

**SCRATCHPAD**

The BTREE Executive Reference Scratchpad allows the user to store all your phone numbers, business cards and what ever in a multi-key data base for immediate access by as many keys as you assign. Extremely easy to use. Allows 3 fields of Misc data.

$39.95

**LIBRARY**

The BTREE Library program allows over 2000 books to be stored on a single double density data diskette in a efficient BTREE structured file for immediate access to your data by Author. Includes 4 build in reports. No sorting necessary for reports.

$39.95

**MAIL LIST**

The BTREE Mail List program allows over 2000 names to be stored on a single double density data diskette. Labels can be generated in zipcode order or by range of zipcodes and also by a code restriction if you wish.

$49.95

## TEXAS COMPUTER SYSTEMS

P.O. Box 1327 Arlington, Texas 76004-1327 ~59

TECHNICAL ASSISTANCE 817/274-9221
ORDER STATUS 817/277-1913
TELEX/TWX/Easylink ELN 62100790

★ **800 433-5184**

No tax out of state. Texans add 5%. Prices subject to change at any time.

**DEALER INQUIRIES INVITED**

## TRS-80

## TCS MODEL III 48k 2 DISK

Systems come with 180 day TCS limited warranty.


**$1695**

With standard 40 track double density drives. Over 340,000 bytes. Includes TDOS

**$1995**

With 2 dual headed 40 track dbl.density drives. Over 730,000 bytes. Includes TDOS

Fully assembled and tested systems that are software compatible and functionally identical to Radio Shack units sold at computer stores for $hundreds more

- CONTROLLER BOARDS are high quality double sided epoxy boards with gold plated contacts.
- POWER SUPPLY is the finest switching type available
- MOUNTING HARDWARE includes power and data cables
- DISK DRIVES are Tandon, the same ones used by Radio Shack ... 40 track, double density, with a 5 millisecond stepping rate

## TCS MODEL III DISK EXPANSION KITS

| | | |
|---|---|---|
| 1 | Controller, Power Supply, Mounting Hardware & Instructions | $279 |
| 2 | Controller, Power Supply, Hardware & one 40 track Tandon drive | $478 |
| 3 | Controller, Power Supply, Hardware & two 40 track Tandon drives | $677 |
| 3A | Kit 3 but with two 80 track Tandon drives (dual sided 40s) | $879 |
| 3B | Kit 3 but with two 160 track Tandon drives (dual sided 80s) | $1099 |

16k of high quality memory chips ... $16 32k of high quality memory chips ... $32

**MODEL III SYSTEMS**

Original 90 day manufacturer's warranty

| | |
|---|---|
| MODEL III 4k Level 1 | $599 |
| MODEL III 16k | $818 |
| MODEL III 32k | $896 |
| MODEL III 48k | $968 |
| MODEL III 48k 2Dr/RS232 | $1944 |

TCS MODEL III Systems use original RS hardware and quality TCS memory. 180 day limited warranty

| | |
|---|---|
| TCS MODEL III 16k | $799 |
| TCS MODEL III 32k | $818 |
| TCS MODEL III 48k | $848 |

**COLOR COMPUTER**

Original 90 day manufacturer's warranty

| | |
|---|---|
| 16k Level 1 | $308 |
| 16k Extended Basic | $398 |
| 32k Extended Basic | $CALL |
| Color Disk 0 $479 | Disk 1 $349 |

TCS COLOR COMPUTERS use original RS hardware & TCS memory. 180 day warranty

| | |
|---|---|
| TCS 32k Extended Basic | $499 |
| TCS 32k Memory Upgrade | $79 |
| TCS Color Disk Drive 0 | $449 |
| TCS Color Disk Drive 1/2/3 | $249 |

## Model II . . Model 16 . . Accessories . . $CALL

TCS is an authorized TRS-80 dealer F701 in Brady, Texas

## WIZARD'S TOUCH

Regularly $199. Introductory Price 12/31/82

**Program Development System $99**

BELIEVE IT OR NOT, YOU CAN DEVELOP YOUR OWN PROGRAM IN 15 MINUTES!!!

The WIZARD'S TOUCH Program Development System is a superior program generator that can provide you with the program you3ve been looking for. It comes with it's own Disk Operating System, providing unlimited backups.

With WIZARD'S TOUCH, here's what you can do:

- Custom design your own special form for maximum ease of entering your data. You have total control over defining the number of characters assigned to each field and any other entry restrictions.
- Fully view and edit all fields at all times. No need to retype an entire field when editing. Just type over the mistakes, or insert or delete them.
- Create a BTREE structured file, allowing you to access your stored data rapidly and without sorting. Duplicate keys are allowed, as well as sub-keys.
- Enter data at full typing speed.
- Define your own special self-help or prompt information that will be displayed for any given field, by just touching one key.
- Expand your program to support up to 8 calculations for each data field. Results can be displayed in any field or in any defined display area. Results can also be saved to disk along with regular data fields. Data such as dates or prices can be formatted as desired.
- Generate your own reports with the built-in pre-defined or custom defined report generator. Just a few keystrokes can get you a report that can be alphabetized or ordered by any field. And it's all automatic.

The WIZARD'S TOUCH is so fast and easy to use. It makes other development systems of the market look like amateurs. No royalty necessary for resale of programs you develop on WIZARD'S TOUCH. It creates a stand-alone, fully commented, BASIC program, which you can easily modify.

Your purchase includes a free subscription to our WIZARD'S TOUCH newsletter, a quarterly publication with free programs, ideas for applications, updates and more.

Now available for MODEL I/III. Coming soon for MODEL II, IBM and APPLE.

and avoid the time-consuming mathematics.

There are several ways to translate the rotational motion of the motor into a linear motion. Linear (or translational) motion is extremely useful in push/pull applications. One manufacturer makes a ball bearing device accomplishing this within the price range of the hobbyist (see list of references).

The stepper motor is a special type of motor whose speed, direction of rotation and degree of rotation can be controlled by commercially available controllers interfaced to the logic level signals at the TRS-80 edge connector output bus. Manufacturers of computer peripherals such as disk drives and printers commonly use these expensive stepper motors. If your robot budget can handle this expense, I highly recommend the investment.

The final schematic may be found in the *TRS-80 Microcomputer Technical Reference Handbook* on pages 84–89, published by Radio Shack, for the interface buffer required for actuator interface. I have not used this circuit before. By all means try it yourself and inform me of your successes with this or any other robot circuit. ■

*Stephen Davids has designed and developed some of the world's most fascinating robot systems, including talking vending machines, robot waiters and K-9, the robot dog.*

### References

*Craig, John. "Digital Speed Control of DC Motors."* Robotics Age, *Summer, 1980.*
*Dorf.* Modern Control Systems. *Addison-Wesley, 1974.*
*Gupton, James. "Unicorn-1 Robot."* Radio-Electronics, *August to January, 1980.*
*Reichardt, Jasia.* Robots, Fact, Fiction and Prediction. *New York; Penguin Books, 1978.*

*Convert Fortran and muMath and learn some tricks.*

# Model III Machine-Language Modifications

*Richard Koch*
*2740 Washington Street*
*Eugene, OR 97405*

With a push from the Federal Communications Commission, Radio Shack launched the Model III in September, 1980. At that time, the company announced that "virtually all our software and most non-Radio Shack software for the Model I is compatible with the Model III." It gradually became clear last year that this statement applies only to programs written in Basic; machine language programs usually require modification.

I will report here on a project to modify two Model I programs, muMATH from the Soft Warehouse by way of Microsoft, and Fortran, from Microsoft by way of Radio Shack. Model III versions of these programs should be available soon from the suppliers. Although I give explicit details for modifying the programs, the main purpose of this article is to describe the type of code which can fail on the Model III.

If you want to make the modifications without reading about the details, refer to the end of this article where I have collected the required changes. I have a confession to make: Not all bugs in Fortran have been removed. I need help from readers with one problem I will explain later.

***The Key Box***

**Basic Level II**
**Model III**
**48K RAM**
**TRSDOS 1.3**
**1 Disk Drive**

### Preliminaries

I modified Scripsit, the Editor/Assembler, APL80, and MMSFORTH before modifying Fortran. Unhappily, most lessons learned from these programs were irrelevant for Fortran, but the lessons may interest you.

The authors of many of these programs changed the keyboard routine. They provided a flashing cursor or debounce or special purpose keys. Typically, they loaded location 4016 in the keyboard device control block (DCB) with the new keyboard driver address (all addresses in this article will appear in hex). Since the Model III keyboard DCB has not been changed, the new keyboard routines would have worked had they not contained surprises.

```
program name:  INIT
name for search:  $INIT
name for search:  $EC
name for search:  $IOFLG
define size of data:  02 00
define program size:  15 00
set loading counter:  00 00
set loading counter:  01 00
set loading counter:  02 00
set loading counter:  00 00
AF
32
data relative:  00 00
32
data relative:  01 00
21
program relative:  14 00
22
00
00
2A
49
40
F9
...
```

*Figure 1*

Most new routines only contain code for the beginning of the routine. Then they jump into the middle of the old ROM code. Addresses in the *middle* of ROM routines are different on the Model III, and such code will not work.

This kind of difficulty was easy to repair. First I searched the code for the address 4016. The instruction containing this number was usually LD (4016), HL and the address of the new driver then became clear. If this driver contained jumps back into ROM, I would use the regular Model III driver. When that failed, I would compare the Model I and Model III keyboard ROM routines and find the address to jump into on Model III.

The keyboard driver problem was the most serious difficulty I encountered in earlier modification projects. Fortran just uses the Model III ROM routine.

The printer routine often caused trouble. The documented printer routine at 3B has not changed, but many software authors wrote their own routines instead. In that case I

# Turn your color computer on to the power of FLEX and OS-9

**NOW FROM THE WORLDS LARGEST SUPPLIER OF SOFTWARE FOR FLEX AND OS-9 COMES FHL COLOR FLEX AND COLOR OS-9, JUST LOOK AT THESE FEATURES:**

## OS-9 ONLY $200
### OS-9 WITH BASIC09 $299

- Unix*-based, multitasking, modular, and versatile
- Over 40 utility commands
- Friendly "Shell" command interpretor
- Tree structured multilevel file directories
- Full time sharing support with log in and file security
- Fast, secure, random and sequential access files
- Comprehensive English language error messages
- Compact real-time multitasking executive
- BASIC09 is the fastest and most comprehensive full Basic language available for the 6809. It combines standard Basic with the best features of Pascal. It's a unique interactive compiler that combines compiler speed, interpreter friendliness, and superlative debugging facilities.

## FLEX NOW ONLY $99

- HI-RES screen formats
- 16 × 32 and 24 × 51, upper and lower case characters
- 24 × 64 and 32 × 64 upper case
- Full ASCII keyboards
- Easy start-up—just type RUN "FLEX"
- Online assistance just type HELP
- Optionally use a standard terminal and printer
- Advance disk I/O and terminal capabilities
- NO additional hardware required
- We have supported FLEX with more software than anyone else in the world for more than 2 years!

## SPECIAL

1. DBASIC, RS Disk Basic under FLEX with a utility to copy RS to FLEX disk $30.
2. ED/ASM line and screen editor and macro assembler, both more powerful than TSC's, and at the same cost, only $100.
3. UTILITIES, a set of 12 utilities especially designed for FHL Color FLEX $50.


**Get on our mailing list, call or send for our complete catalog of over 100 products for FLEX and OS-9. We're doing exciting things with your color computer!**

✓ 262

# FRANK HOGG LABORATORY

THE REGENCY TOWER • 770 JAMES ST. • SYRACUSE, NY 13203 TELEPHONE (315) 474-7856

*Unix is a trademark of Bell Labs. *OS-9 & BASIC09 are trademarks of Microware & Motorola Inc. *FLEX is a trademark of Technical Systems Consultants Inc.

| Location | New Code | Old Code |
|---|---|---|
| 6C66 | 000000 | 321543 |
| 87B6 | 000000 | 321543 |
| 6C61 | 000000 | 221343 |
| 87C5 | 000000 | 221343 |
| 6FF4 | 4B44 | 7344 |
| 87DA | 21FFFE | 2A4940 |
| 6FA6 | 00FF | C05F |
| 6FAE | 00FF | C05F |
| 72DB | 00FF | C05F |
| 72F8 | 00FF | C05F |
| 7307 | 00FF | C05F |
| 730D | 00FF | C05F |
| 7318 | 00FF | C05F |
| 736E | 00FF | C05F |
| 73A1 | 00FF | C05F |
| 772B | 00FF | C05F |
| 7741 | 00FF | C05F |
| 7759 | 00FF | C05F |
| 7770 | 00FF | C05F |
| 7E58 | 00FF | C05F |
| 7FB0 | 00FF | C05F |
| 7215 | 05FF | C55F |
| 72CA | 0BFF | CB5F |
| 721B | 782004FE8018 | CA2A7278E67F |
| 7221 | 08B720031B18 | C2287278EE80 |
| 7227 | 05FE81380113 | 471B1B7887D2 |
| 722D | 000000 | 307213 |
| 784D | 2100000909B7 | E67F7E21FEFF |
| 7853 | 20032B1805FE | CA5878EE8087 |
| 7859 | 813801230000 | DA5D78230909 |
| 7F74 | 2811FE1C37C2 | CA7A7FFE1C37 |
| 7F7A | AC673AA85FE6 | C2AC67E521AA |
| 7F80 | 7F4F06000936 | 5F5E2356237E |
| 7F86 | 1AD1C1E1C9 | 23666FCDF6 |
| 5200 | C30061 | C3B587 |
| 6100 | 222061213061 | 000000000000 |
| 6106 | 1140FF013800 | 000000000000 |
| 610C | EDB02A2061C3 | 000000000000 |
| 6112 | B587 | 0000 |
| 6130 | 215EFF7EB736 | 000000000000 |
| 6136 | 00211E5FC23B | 000000000000 |
| 613C | 79E1C31079E5 | 000000000000 |
| 6142 | 215EFF36FFE1 | 000000000000 |
| 6148 | E3E1E3C3F278 | 000000000000 |
| 614E | 0000CDE07F21 | 000000000000 |
| 6154 | 4AFF36C3C9E5 | 000000000000 |
| 615A | 214AFF36C221 | 000000000000 |
| 6160 | 5EFF3600E1C3 | 000000000000 |
| 6166 | 0277 | 0000 |
| 62C3 | 69FF | 0277 |
| 72FA | CD60FF | CDE07F |
| 78EF | C351FF | E3E1E3 |
| 7938 | C340FF | 211E5F |

*Table 2. Changes for EDIT/CMD*

| Location | Old Code | New Code |
|---|---|---|
| 63D4 | 916F | 2542 |
| 63DD | 916F | 2542 |
| 64B8 | 916F | 2542 |
| 6367 | E06F | 6042 |
| 6406 | E06F | 6042 |
| 640F | E06F | 6042 |
| 642D | E06F | 6042 |

*Table 1. Changes for muMATH (either 32K or 48K version)*

changed the instruction LD (37E8),A for printing the contents of the accumulator to the equivalent new instruction OUT (0F8H),A. Fortran uses the ROM routine and needs no change.

When I began this game, I was frightened most by programs with extensive calls to the arithmetical routines in ROM. For example, APL80 uses the Basic ROM to add numbers, multiply them, compute square roots, calculate sines, and so on. The floating point package for MMSFORTH also calls ROM routines. The good news is that the arithmetical ROM routines have not changed locations in Model III, and programs which use them are as easy to modify as other programs. Fortran does not use the arithmetical ROM routines.

Finally, the routine which sets the top of memory is usually incorrect. There are four techniques here. Some programs like Checkers-80 simply use 16K and have no need for additional memory. Other programs come on a disk containing several versions for different memory capabilities; for example, the muMATH package contains a 32K version and a 48K version. Some programs find the top of memory by directly testing location FFFF and working down until a live cell is found; APL80 and Scripsit work like that. (Such programs wipe out protected code at the top of memory, so readers with serial printer drivers or other routines there will have to make changes.) The most sophisticated technique for finding the highest memory location is to ask the DOS. Model I DOS stores the highest available memory at location 4049 and Model III DOS stores the information at 4411.

Fortran examines location 4049 to find the top of memory. I will begin the modification at that point.

**Review of Fortran**

The Fortran disk contains three programs and an additional library file. The programs are called EDIT/CMD, F80/CMD, and L80/CMD; the additional file is called FORLIB/REL. Microsoft constructed the package to accommodate writing and running very large Fortran programs. Consequently, at any moment the computer memory contains only the code to do the task at hand.

Edit is an editor to simplify typing text into the computer; it works like the Basic editor or the editor portion of EDTASM. When a program is complete, it is written to the disk and the Edit program is erased.

Unfortunately, Edit is a complicated program. It caused most of the trouble in the project. Microsoft has written Edit to use with programs too large for the computer memory. Old programs are not read back into memory for reediting. Instead, the computer stores only changes and additional text in memory and reads from disk when it refers to earlier material. Thus, Edit must be able to work with two files simultaneously when writing a reedited program back to disk. Even a third file can complicate this process, for when a large file is written to the disk, Edit also writes a small "index file" of vital information about the large file for faster processing.

After you write a program, the compiler F80 converts it to code. This program also juggles three files at once. If you call your program EXERCISE, you must save it from the editor with the extension /FOR. The compiler pro-

| Location | Old Code | New Code |
|---|---|---|
| 78C5 | 4940 | 1144 |
| 575B | 321543 | 000000 |
| 7C09 | 321543 | 000000 |
| 5767 | 221343 | 000000 |
| 7C0F | 221343 | 000000 |
| 795C | 7344 | 4B44 |
| 797F | 7344 | 4B44 |
| 7999 | 7344 | 4B44 |

*Table 3. F80/CMD Changes*

| Location | Old Code | New Code |
|---|---|---|
| 5214 | 4940 | 1144 |
| 5204 | 321543 | 000000 |
| 582A | 321543 | 000000 |
| 5210 | 221343 | 000000 |
| 5830 | 221343 | 000000 |
| 5D0C | 7344 | 4B44 |

*Table 4. L80/CMD Changes*

duces a machine language version of the program on a file called EXERCISE/REL. It also produces an Assembly language version of the code which you can list and inspect; this file is called EXERCISE/LST.

The program EXERCISE/REL is not in a form which can run directly from the computer. One of Fortran's most important features is its ability to handle precompiled subroutines. Suppose you often need a routine called Matrix (no need to explain what it does). You can write the program Matrix only once and compile it to MATRIX/REL code. When the program EXERCISE/FOR uses Matrix, include the line CALL MATRIX in the text. Even though Matrix appears nowhere else in the program, the compiler will not complain. But you must merge or link the file MATRIX/REL together with EXERCISE/REL before you can run the program. L80 does this linking. It converts EXERCISE/REL and any routines it uses into a single runnable file called EXERCISE/CMD. Many commonly used routines are included in the file FORLIB/REL; the linker must always merge at least these programs with the EXERCISE/REL file.

In summary, during a Fortran session with Exercise the disk may contain EXERCISE/FOR, EXERCISE/ZFO (the index file), EXERCISE/REL, EXERCISE/LST, and EXERCISE/CMD. The Fortran package juggles them, and every routine doing the juggling can cause trouble on the Model III.

> *"The Fortran system tracks related files by giving them the same first name with different extension names."*

**First Steps**

I am using version 1.0 of Fortran. Use DOS 1.3 during the modification; the "end of file" protocol in earlier versions of DOS is incompatible with the modifications we will make.

The first change is easy. The three programs Edit, F80, and L80 find the top of memory by examining location 4049; we must change this number to 4411. Use Patch to change 4940 to 1144 at location 87DB in EDIT/CMD, at location 78C5 in F80/CMD, and at location 5214 in L80/CMD.

These changes do not end our "top of memory" problems. Fortran programs created by the system still use location 4049 to find the top of memory. Solving this problem is much more difficult (and much more fun); it will be the last step of our modification.

If you did not own a Model I you may never have heard of the Break key's idiosyncracies. On the Model I, this Break key is "activated" by turning on Debug. After leaving Debug, you can run programs as on the Model III. During these programs, a (perhaps accidental) press of the Break key causes you to fall back into Debug. This dismaying response can, of course, be deactivated.

Location 4315 controls the action of the Break key. It contains C3 when the Break key is activated and 00 when it is inactive. All three of our packages contain code which sets this location to 00 at the start of the program and to C3 when it ends. We must eliminate this code. Replace 321543 by 000000 at locations 6C66 and 87B6 in EDIT/CMD, at locations 575B and 7C09 in F80/CMD, and at locations 5204 and 582A in L80/CMD.

Our programs also temporarily replace the contents of location 4313 on startup and return these contents to normal before ending. Documentation I own claims this location is a "jump director for positive DOS commands." The Model III has different code near 4313. So we eliminate manipulation of address 4313 by changing 221343 to 000000 at locations 6C61 and 87C5 of EDIT/CMD, at locations 5767 and 7C0F of F80/CMD, and at locations 5210 and 5830 of L80/CMD.

The Fortran system tracks related files by giving them the same first name with different extension names. The package handles these extensions automatically. If you wish to compile Exercise, you need type only these eight letters; the compiler automatically looks for EXERCISE/FOR and calls the resulting code file EXERCISE/REL. The compiler does this using a DOS routine called $PUTEXT. This routine was not documented on the Model I, and Radio Shack changed its location. It has been documented on

# THE LEMON™ SOURS SURGES


Our crop-The Lemon™, The Lime™, and The Orange™ are designed to eliminate undetected submicrosecond overvoltage transients from electrical circuits. Commonly referred to as "spikes", or "glitches", these transients can cause hardware and software damage to unprotected circuits.

Today's electronic products are often microprocessor controlled - mini and micro computers, televisions, video cassette recorders - to name a few. Each of these products is sensitive to fluctuations in electrical power lines. Power switching devices such as refrigerators coming on and off or air conditioners starting up can be responsible for a momentary surge or spike of electricity in a circuit. Even your local utility stepping-up transformers to add power at peak load times or an electrical storm passing through can trigger surges. Such surges can cause equipment to falter at times, not to work at peak performance or fail completely. An entire data base can be lost.

Now you can prevent this from happening to you with an AC Surge Protector from Electronic Protection Devices. Each Protector is a solid state clamping device with 6 outlets utilizing modern high speed semiconductor technology. Using our Protectors is as simple as plugging it into any standard three wire duplex outlet then plugging what needs protection into it. Each Protector exceeds the IEEE 587-1980 Guide for Surge Voltages in Low Voltage AC Power Circuits.

When you compare the cost of computer hardware, software and your time with the price of a Protector (from \$59.95 to \$139.95), you'll want to sour your surges with one of the AC Surge Protectors from EPD, which are available through your local dealer.

*Electronic Protection Devices*
5 Central Avenue
Waltham, Massachusetts 02154

In Massachusetts Call:
(617) 891-6602

Outside Massachusetts Call:
1-800-343-1813

Dealer Inquiries Invited

the Model III. The routine begins at 4473 on the Model I and at 444B on the Model III. Consequently we change 7344 to 4B44 at location 6FF4 in EDIT/CMD, at locations 795C, 797F and 7999 in F80/CMD, and at locations 5D0C in L80/CMD.

This last change has a dramatic effect. F80 and L80 now work perfectly. At first glance, Edit seems to work too. But as soon as we try to reedit a file, the editor becomes very sick and refuses to reload the file.

**The DCB**

Now we come to the most annoying of all Model I to Model III changes. Radio Shack lengthened the DCB for disk files from 32 bytes to 50 bytes.

When you use a file, the file name is placed in an area of memory called the device control block or DCB, and the address of this block is given to a DOS routine called Open. The routine locates the file in the directory and replaces its name in the DCB with technical information about it. While you manipulate the file, this DCB becomes a sort of scratchpad for the DOS. When you close the file, the name of the file is returned to the DCB.

The computer always reads or writes 256 bytes at a time. For example, when Patch changes a single byte of code, the computer actually reads in 256 bytes, changes that one byte, and then writes back all 256 bytes. The 256 byte memory area from which the computer reads and writes is called the buffer; the location of this buffer is given the system when a file is opened, and becomes part of the technical information in the DCB.

Many programmers arrange memory allocation so the file buffer immediately follows the DCB. How does this arrangement affect the change in the length of the DCB? Imagine that we are going to read from a file. First we place the name of the file in the DCB. This name takes less than 32 bytes and the rest of the DCB contains irrelevant information. Next, we open the file. The Model III scoops up the name and replaces it with technical information. This information is 18 bytes long (for a fairly short file) followed by 24 bytes of FF. Forty-two bytes in memory are changed. Since the Model I DCB is only 32 bytes long, an extra 10 bytes in the file buffer are filled with FF garbage.

This garbage is irrelevant because the file buffer contains nothing interesting yet. Next, we read the first record of the file. The beginning bytes of this record overwrite the last 10 bytes of FF in the technical information contained in the DCB. The technical information at the end of the DCB is information on file extents; you can change the last few bytes unless the file has a large number of extents.

Most programmers use the information in the file buffer only when the file is open. When the buffer is changed during the Open and Close routines, no harm is done. In summary, when the file buffer immediately follows the DCB in a Model I program, the program will probably work on the Model III.

Programs such as F80 and L80 can cause trouble on a Model III. But this problem will only come up once or twice in a lifetime, when you try to write a file on a dirty disk and force a very large number of extents. If you are bothered by this possibility, feel free to change the DCBs in these two programs.

Not all programs are arranged in the above manner. Sometimes the DCB occurs in the middle of a program, followed immediately by other important code. It will be disastrous to use the disk during the operation of such programs. It turns out that Edit and muMATH fall in this class.

**MuMATH**

MuMATH is produced by the Software House in Hawaii. The program is written in Lisp and can perform algebraic manipulations. It expands polynomials, differentiates any reasonable function, and even integrates using the method of substitution. The program happily handles very large numbers; for instance it can calculate 100! down to the last digit and then convert the result to hex. Some people claim that the very first computer program was one written by Ada Byron for Babbage's machine (but never run); the program computed Bernoulli numbers. Bernoulli numbers are very large rational numbers and the whole point is to know their numerators and denominators exactly; approximations are not interesting. MuMATH is the only system available for the TRS-80 which can run Ada's program without resorting to tricks.

When I first converted muMATH to the Model III, I was happy to find that it worked without change. Later I discovered a curious flaw. The authors of muMATH included a large demonstration file on the disk. This file asks muMATH a series of questions and muMATH stops to answer each question before reading further. Of course, muMATH can answer quite rapidly; you must freeze the screen if you wish to examine the answers. When I first tried this program on Model III, the questions and responses scrolled rapidly across the screen; I neglected to look at the answers because I had seen them all before. But soon a strange answer caught my eye as it floated off the screen and after freezing the display I discovered that muMATH had forgotten everything it once knew.

MuMATH repeats questions it cannot answer. If you ask it to integrate SIN(X), it replies -COS(X), but if you ask it to integrate SIN(X*X*X), it just replies INT(SIN(X*X*X),X). But in the demonstration program, muMATH replied 1+2 when asked to sum one and two.

The difficulty here is the DCB. In muMATH, it is hidden in the middle of the code. As soon as we try to read the demonstration disk, a small part of muMATH dies.

Unfortunately, muMATH uses up the complete 32K or 48K of computer memory and leaves no obvious spot to place a larger DCB. However, the DCB can be located safely in a buffer area on page 42 of the DOS. There actually are two DCBs, one at 6F91 for reading and one at 6FE0 for writing. So to patch either version of muMATH, change 916F to 2542 at 63D4, 63DD and 64B8. Also change E06F to 6042 at 6406, 640F, 642D and 6367.

**Fixing the Edit DCB**

Several techniques work to find the DCBs for Edit. Perhaps the easiest is to look for calls to the INIT routine at 4420 and the Open routine at 4424. Before calling these routines, the computer loads DE with the address of the DCB. Experiments like these show that the Edit program has DCBs at 5FA0 and 5FC0. Moreover, there is a temporary storage area for the contents of these blocks at 5FE0.

When the 5FA0 DCB is used, the computer overwrites the first part of the 5FC0 DCB. We are going to move this 5FC0 DCB to the address FF00; later on we will use other pieces of this page of memory as a patch space. If you have only 32K, you will have to change various pieces of the modification which follow (buy another 16K instead!).

Before using this area, we have to protect it from Edit by revising the reference to the top of memory. At 87DA we find the command LD HL,(4411). (The number 4411 was 4049 before we changed it a moment ago.) Change this command to LD HL, 0FEFFH. Use Patch to change 2A1144 to 21FFFE at 87DA.

Next find all references in the code to the number 5FC0. Unfortunately, many programs are savvy enough to use the technical information stored in the DCB; Edit is one of these. Consequently you also must find references to 5FC5 and 5FCB in the code.

Once you find these references, the final step is easy; use Patch to change C05F to 00FF at 6FA6, 6FAE, 72DB, 72F8, 7307, 730D, 7318, 736E, 73A1, 772B, 7741, 7759, 7770, 7E58 and 7FB0. Then change C55F to 05FF at 7215. Finally change CB5F to 0BFF at 72CA.

Now test the result. Edit can read in and change a short program. But nasty surprises are ahead. When Edit reads a program longer than eight lines, it starts listing lines out of order, forgets pieces of lines, and ignores the end of the program entirely.

The new problem does not occur with DOS 1.1. Reaching into the bag of Radio Shack changes, we have just discovered the "end of file" revision.

*"The routine locates the file . . . and replaces its name in the DCB with technical information."*

**The EOF**

The DOS stores certain technical information in the DCB while a file is open. Call the address of the first byte in the DCB "location zero." Then locations 8 and 12 contain end of file information. To understand this information, you must know a little more about how DOS operates.

On the disk, a file consists of a series of "physical records", each 256 bytes long. These records are numbered record 0, record 1, record 2, . . . The records may be scattered all over the disk, but the operating system handles all of that.

Sometimes it is inconvenient to deal with blocks of data 256 bytes long. The DOS allows users to write files with record lengths other than 256. For instance, Edit files have records of length 128. The length of the records in a file is called the "logical record length" or LRL.

However, this logical record length is fictitious. When records are written to the disk, they have length 256. The DOS converts the logical records in the programmer's mind into blocks of code exactly 256 bytes long. When a program asks that a record be written to disk, the record is actually written into the 256 byte buffer we discussed earlier. But the buffer is written to disk only when it is full.

For instance, imagine that you are working with records of length 100 and you want to write four such records to disk. When you record 0, the operating system writes it into the buffer, but nothing goes to disk. The next record also moves only to the buffer, filling spots 100-199. The first part of record two moves to the buffer, filling spots 200-255, and then this buffer is written to disk. Next the buffer is cleared and the end of record two is written into its first 44 spots. Finally record number three is written from spot 44 to spot 143. The contents of the buffer, however, will not be written to disk until further logical records are written or you close the file.

The fictitious records of length LRL which compose a file are called "logical records." The actual 256 byte blocks of code on the disk are called "physical records." Thus in the above example, logical record number three is in physical record number one.

Now I can explain locations 8 and 12 in the DCB. In DOS 1.3, the information in these locations tells where to place the *next* byte to extend the file. If you closed the file described in the last paragraph after the four records were written, on the disk the file would consist of two records, each with 256 bytes of code. The first record would be called record zero and the second record one. In record one, only bytes 0-143 would be significant. When we reopen this file and fill the DCB with technical information, location 8 will contain 144 and location 12 will contain 1, because the next byte to be written would go to spot 144 in record one. Notice that the information in locations 8 and 12 is about physical records, not logical records.

An important special case occurs when a record ends at the boundary of a physical record. Suppose that a file contains exactly three physical records called record zero, record one and record two. Then location 8 will contain 0 and location 12 will contain 3 because the next byte to be added would be inserted at the start of physical record 3.

Locations 5 and 10 in the DCB contain similar information. Suppose that we are reading through a file. Each time we read a logical record, locations 5 and 10 are updated to point to the *next* byte to be read. For instance, if location 5 contains 52 and location 10 contains 3, then the *next* read command will start by reading spot 52 on physical record 3 of the file on disk.

All this is getting very complicated. Moreover, the story is different for DOS 1.1 and for the Model I. If a program uses the technical information in the DCB, it probably must be altered for DOS 1.3.

**Fortran Record Number Calculations**

In the Edit program two routines perform calculations on the above DCB information. Both routines calculate which 128 byte logical record was the last one read from disk. Let X equal the contents of location 5 and Y equal the contents of location 10. The required record number is $2^*Y-1$ if $X=0$, $2^*Y$ if $0<X\leqslant 80$, and $2^*Y+1$ if $80<X$. But the routines give a different answer: $2^*Y-2$ if $0<X\leqslant 80$ and $2^*Y-1$ if $X=0$ or $80<X$.

Incidentally, I found these routines by searching for the number 4442; the routine at 4442 positions a file to read the logical record number stored in BC. Once we *find* the error it is easy to fix. Starting at 721B, insert the code 78 20 04 FE 80 18 08 B7 20 03 1B 18 05 FE 81 38 01 13 (the old code here reads CA 2A 72 78 E6 7F C2 28 72 78 EE 80 47 1B 1B 78 87 D2 30 72 13). Then starting at 784D, insert the code 21 00 00 09 09 B7 20 03 2B 18 05 FE 81 38 01 23 00 00 (the old code reads E6 7F 7E 21 FE FF CA 58 78 EE 80 87 DA 5D 78 23 09 09).

This change improves the operation of Edit dramatically. Unfortunately, there are still two difficulties. When Edit attempts to read or write a file whose length is a multiple of 128 bytes, it fails. Writing such a file causes a catastrophic failure and reboot.

At 7F6C in Edit, a routine reads 128 bytes of code from the disk and stores them beginning at the location contained in HL. After reading, the routine checks to see if the record read from the disk was the last record in the file; if so, it puts 1A in memory immediately following the last byte of code from the file.

There is a tricky special case. If the last record in the file contains exactly 128 bytes, there is no spot in memory for 1A. The routine is supposed to wait until Read is called again, when it should place 1A at the first available memory spot.

As an interesting exercise, try to write such a routine using the technical information provided. You should know that when the Read routine reads a logical record and some of the bytes read are not in the file, it sets the zero flag and places 1C in the accumulator. This happens even when only a few of the bytes read from disk do not belong to the file, but it does not happen if the file ends exactly at the boundary of the record. The required changes in code are easily implemented. Starting at 7F74, insert the code 28 11 FE 1C 37 C2 AC 67 3A A8 5F E6 7F 4F 06 00 09 36 1A D1 C1 E1 C9. (It will replace CA 7A 7F FE 1C 37 C2 AC 67 E5 21 AA 5F 5E 23 56 23 7E 23 66 6F CD F6.)

Edit cannot write files whose length is an even multiple of 128. I can solve the problem, but I do not know exactly why it came about.

The Write routine occurs at location 7917. After writing, the computer checks several memory locations. If conditions are exactly right, it tricks the stack by loading its last four locations with data from 5270–5277. For weeks I tried to understand this code; from the beginning, experiments disclosed that it operated only in the exceptional case causing trouble. It is easy to find the code's "mirror image"; near 78E8 a routine saves the last four locations of the stack to 5270-5277. The basic problem on the Model III is that this "mirror image code" sometimes does not run before the section of code causing trouble.

The solution I have invented runs like this. We modify the mirror image code so it sets a flag when it runs. Then we allow the dangerous code to run only if this flag has been previously set. The modified code is allowed to affect computer operations only during the writing of the main file (the index file causes no trouble).

Place these patches on the last page of memory. Change location 5200 from C3 B5 87 to C3 00 61; starting at 6100, replace the 00s with 22 20 61 21 30 61 11 40 FF 01 38 00 ED B0 2A 20 61 C3 B5 87. Then starting at 6130, replace the 00s with 21 5E FF 7E B7 36 00 21 1E 5F C2 3B 79 E1 C3 10 79 E5 21 5E FF 36 FF E1 E3 E1 E3 C3 F2 78 00 00 CD E0 7F 21 4A FF 36 C3 C9 E5 21 4A FF 36 C2 21 5E FF 36 00 E1 C3 02 77. Incidentally, 6100-6200 is Edit's write buffer; we are using it here temporarily to store patches which will reside up near FF00.

Connect these patches to the main program by changing 02 77 to 69 FF at 62C3, CD E0 7F to CD 60 FF at 72FA, E3 E1 E3 to C3 51 FF at 78EF, and 21 1E 5F to C3 40 FF at 7938.

Edit, F80, and L80 now work!

### Top of Memory

The instruction manual explains that each Fortran program begins the same way: LD BC,$$L; JP $INIT. It warns that $INIT sets the stack pointer to the top of memory. How is this location determined? Since L80 creates each Fortran program, I hoped that L80 set the top of memory. No such luck; the program examines location 4049.

The $INIT routine is contained in the library FORLIB/REL; we must find 4049 in this library and change it to 4411. Unfortunately the library is in code—not Z80 code, but Microsoft linker code. The manual explains that to save space in REL files, the Fortran system constructs these files as long chains of zeroes and ones; significant information is not aligned with eight bit boundaries. Therefore it is necessary to convert FORLIB/REL (which is a very large file) into a gigantic chain of zeroes and ones and then search through this chain for the bit pattern corresponding to 4049. (The coding method introduces an extra zero bit between each of these numbers.)

There might be hope for doing this by hand if $INIT occurred near the start of the file (and I expected to find it there); actually $INIT is hidden in the middle of things. Therefore I wrote an Assembly language program to search the file automatically.

Using the results of the search I can explain how to change the code. Use the F command of Debug to display the file FORLIB/REL. Use the plus key to leaf through this file until you come to a page which begins 8C 00 C9 11 50 E3 2A 00 . . . Find the two bytes 49 20 starting at location 5C of this file. Use the M command to change these bytes to 11 22 and then push Enter. All modifications are now complete.

I did not actually search for 4049, though it would have been the logical thing to do. Instead I looked for the bit pattern corresponding to INIT. In advance, I knew that the $INIT routine reads:

```
XOR   A
LD    (ADD1),A
LD    (ADD2),A
LD    HL,NUMBER
LD    (ADD3),HL
LD    HL,(4049H)
LD    SP,HL
LD    H,B
LD    L,C
JP    (HL)
```

Once I found INIT, I began decoding the bit pattern by hand using the rules in a very obscure part of the manual. I gradually wrote down the information shown in Fig. 1.

### A Problem for the Reader

After making all these changes Fortran programs work well on the Model III *unless* they write data to the disk. Files like FORT06/DAT created on disk are incomplete and out of order. I do not know how to fix this problem. Can someone search for the routine in the FORLIB/REL file that reads and writes data files and explain how to patch it?

**Summary of Required Changes**

Tables 1–4 list the changes you must make to muMATH, Edit, F80, and FORLIB/REL for the Model III. In each case, I list the starting location, old code, and new code. Use the Patch command to make these changes.

Consider, for example, the first change. It asks that location 63D4 of muMATH be changed from 916F to 2542. To do this, type PATCH MATH48/CMD (ADD = 63D4, FIND = 916F, CHG = 2542), and press Enter. The disk will turn on briefly and then the computer will print "Patch Made". If it ever prints "String NOT Found", you made a typing mistake and you should issue the command again. Replace MATH48/CMD by the correct program name in other cases: MATH32/CMD, EDIT/CMD, F80/CMD, or L80/CMD.

Finally, you must modify FORLIB/ REL in a different manner. Run Debug and issue the command F. The computer will ask for a filespec; type FORLIB/REL and press Enter. The disk will run and the screen will show a display of numbers. Each time you press the plus key, the disk will run and the display will change. Keep pressing plus until you see 8C 00 C9 11 50 E3 2A . . . along the top line of the display. Then press M. Immediately a small square will appear at the top left of the screen. Move the square down to line 5 using the down arrow key, and then over 12 spaces using the right arrow key until it lands on top of 49. Type 11 22; the two entries 49 20 will change to 11 22. Now press Enter. The disk will turn on briefly. Press the Break key twice and you will be back in DOS. ■

*Richard Koch is Associate Professor of Mathematics at the University of Oregon.*



✓See List of Advertisers on Page 451

*Find a solution to your waiting line problems with the TRS-80.*

# Queue Up!

*David E. Clapp, Ph.D.*
*1769 Kingsway Court*
*Cincinnati, OH 45230*

Waiting lines are everywhere in modern society. Nearly all services available today involve waiting. The cost of providing instant service is usually high, and is only justified in a few rare situations. There is normally a tradeoff between cost of service and cost of waiting. With limited servers, a line develops with demand for the service. Typically, there is a cost of providing a waiting area to accommodate the waiting customers (or jobs). There is a penalty for customers who leave the line when waiting seems too long.

Most firms or institutions which provide a service spend some time in waiting line analysis aided by a computer program. Following is a TRS-80 program for analyzing simple waiting line problems. While the analysis is limited to fairly basic models, the level achieved is surprising.

**Definitions**

This program is limited to Poisson waiting lines, relatively unconstrained, random arrivals and departures. The Poisson classification is a mathematical characterization of a truly random arrival and departure waiting line discipline. In most cases, the Poisson behavior may be assumed; even in those cases which are not theoretically correct, the computer program provides useful rough results. The formal conditions for Poisson waiting lines are described in most operations research textbooks; the readers who wish to rigorously classify their waiting line models should consult such a text. In any analysis effort, it is the analyst's responsibility to validate results for the system under study.

The Kendall notation is used in the following descriptions of examples of waiting line models, to describe the model parameters. This convenient shorthand notation has the form (M/M/c)-(GD/K/N). These descriptors completely define the parameters of a Poisson waiting line. The first two letters (M) signify Poisson arrival and departure rates, while the c designates the number of servers in the system. The last three letters describe the properties of the calling customers. GD (general discipline) implies a first-come first-served scheme. If other service schemes are in use, this model will not work.

K and N specify available waiting space and numbers of calling customers respectively. As an example, consider the notation (M/M/1)-(GD /9999/9999) indicates that customers arrive and are serviced according to the Poisson model (M/M). The customers file through one server (c = 1) in a first-come first-served fashion (GD). There is unlimited space for waiting (9999) and the calling population is theoretically unlimited (9999). This is the classical (M/M/1) waiting line model where both arrivals and departures follow a random pattern.

The computer analysis of this model is in Fig. 1. This printout is in response to the query "Input arr. rate, svc. rate, no. servers, waiting spaces, customer

**The Key Box**

**Basic Level II**
**Model I**
**16K RAM**

```
(M/M/ 1 )-(GD/ 9999 / 9999 )

ARR. RATE = 20                  EFF. ARR. RATE = 20
SVC. RATE = 30                  TRAFFIC INTENSITY = .666667

AVG. TIME IN SYS. = .1          AVG. TIME IN LINE = .0666667
AVG. NO. CUST. IN SYS. = 2      AVG. NO. CUST. IN LINE = 1 33333

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 25 ) CUSTOMERS IN THE SYSTEM.
.333333 .222222 .148148 .0987654 .0658436 .0438958 .0292638 .0195092 .010062
8.67077E-03 5.78051E-03 3.85367E-03 2.56912E-03 1.71274E-03 1.14183E-03
7.6122E-04 3.3832E-04 2.25547E-04 1.50364E-04 1.00243E-04 6.68286E-05
4.45524E-05 2.97016E-05 1.98011E-05 1.32007E-05
```

*Fig. 1*

```
(M/M/ 1 )-(GD/ 9999 / 9999 )

ARR. RATE = 25                  EFF. ARR. RATE = 25
SVC. RATE = 30                  TRAFFIC INTENSITY = .833333

AVG. TIME IN SYS. = .2          AVG. TIME IN LINE = .166667
AVG. NO. CUST. IN SYS. = 5      AVG. NO. CUST. IN LINE = 4 16667

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 53 ) CUSTOMERS IN THE SYSTEM.
.166667 .138889 .115741 .0964506 .0803755 .0669796 .0558163 .0465136 .0387613
.0323011 .0269176 .0224313 .0186928 .0155773 .0129811 .0108176 9.01465E-03
7.51221E-03 6.26017E-03 5.21681E-03 4.34734E-03 3.62278E-03 3.01899E-03
2.51582E-03 2.09652E-03
Note Output terminated to conserve space
```

*Fig. 2*

```
(M/M/ 1 )-(GD/ 10 / 9999 )

ARR. RATE = 20                  EFF. ARR. RATE = 19.883
SVC. RATE = 30                  TRAFFIC INTENSITY = .666667

AVG. TIME IN SYS. = .0941175    AVG. TIME IN LINE = .0607841
AVG. NO. CUST. IN SYS. = 1.87134  AVG. NO. CUST. IN LINE = 1.20857

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 10 ) CUSTOMERS IN THE SYSTEM.
.337232 .224821 .149881 .0999206 .0666138 .0444092 .0296061 .0197374 .0131583
8.77218E-03 5.84812E-03
```

*Fig. 3*

population." For this example, the input was 20,30,1,9999,9999 where the arrival rate (20 per hour) and the departure rate (30 per hour) were arbitrarily chosen. The printout echoes this input data. In an actual problem, these rates must be determined by studying the system and tabulating actual arrival and departure data.

The first analytical results are the effective arrival rate and the traffic intensity. Effective arrival rate is only of interest in problems where the calling population is not unlimited. Traffic intensity is the ratio of arrival rate over departure rate, providing an estimate of how busy the server is in the system. For an unlimited system, this value must be less than one, or the waiting line will grow infinitely. If the intensity exceeds one, arrivals occur faster than they can be served.

The next two output values concern the average time customers wait in the system and in the line. In this example, the server is quite prompt and customers only spend six minutes in the system, four minutes of which are in line. The service rate is 30 customers per hour, or two minutes per customer, and six minus two equals four.

The service speed is further illustrated by the line statistics in the printout. There is an average of 1.3 persons in line or a total of two persons in the system at any one time. These results are ideal for the customer, but the server is not very busy.

The last section of data indicates the probability of exactly N customers on the system, beginning with zero, after the system reaches steady state. These values are computed by the program until the probabilities become less than 0.00001 (an arbitrary limit which can be changed in the program). In Fig. 1, N varies from zero to 25 before the probability values become insignificantly small. The analyst concludes that the likelihood of no one in the system (N = 0) is 0.33, or that the server is idle 33 percent of the time. The possibility of more than two persons in the system rapidly falls off, indicating the expected "decaying exponential" pattern typical in Poisson waiting line problems.

### Examples

The ideas in the previous section can be illustrated with a few examples of simple waiting line models. The first example is a modification of the model in the previous section. Because the server was idle a considerable amount of the time, activity is stimulated and more customers assigned to that server. The results of an increased arrival rate are shown in Fig. 2.

The traffic intensity increases, the server is busy 84 percent of the time, and there is a likelihood of up to 53 customers in the system. If waiting area becomes a problem, the analyst can add probability values up to an estimate of system saturation. If 10 persons in the system are a problem, the analyst adds probabilities for nine persons or less, yielding 0.84. The probability of 10 or more is 0.16 (1 minus 0.84 equals 0.16). Use this type of estimate to evaluate the need for more waiting area.

```
10 CLS:M=200:DIM P(200)
20 PRINT"INPUT ARR. RATE, SVC.RATE, NO.SERVERS, WAITING SPACES, CUSTOMER POPULATION"
25 INPUT XL,XM,C,XN,XK
30 CLS:IT=1:NC=C:N=XN:K=XK:RH=XL/XM:EL=XL
80 LPRINT TAB(15)"(M/M/";NC;")-(GD/";N;"/";K;")"
81 LPRINT
90 IF K<9999 AND K > M AND N >M GOTO1090
95 IF M<N THEN 110
97 IF N>K THEN 120
100 NN=N:GOTO 130
110 IF M>K THEN 120
115 NN=M:GOTO 130
120 NN=K
130 RC=RH/C
140 PZ=1
150 FOR I=1 TO NN
153 P(I)=0
155 NEXT I
160 IF NC > 1 THEN 540
170 IF K<9999 THEN 290
180 IF RC >=1 AND N= 9999 THEN1090
190 PZ=PZ-RH
200 IF N<9999 THEN PZ=PZ/(1-RH :(N+1))
210 P(1)=PZ*RH
220 FOR I=2 TO NN
230 P(I)=P(I-1)*RH
240 IF P(I))=1E-5 THEN 250
242 IF IT=0 THEN 960
244 IF IT<>0 THEN 260
250 IT=0
260 NEXT I
270 I=NN+1
280 GOTO 960
290 P(1)=XK*RH
300 PZ=PZ+P(1)
310 FOR I=2 TO NN
320 X=K-I+1
330 P(I)=P(I-1)*X*RH
335 PZ=PZ+P(I)
340 NEXT I
350 PZ=1/PZ
360 FOR I=1 TO NN
370 P(I)=PZ*P(I)
380 IF P(I))=1E-5 THEN 390
382 IF IT=0 THEN 420
384 IF IT<>0 THEN 400
390 IT =0
400 NEXT I
410 I=NN+1
420 LL=I-1
430 QL=0
440 FOR J=NC TO LL
450 X=J-NC
460 QL=QL+X*P(J)
465 NEXT J
470 R=C*PZ
480 FOR J= 1 TO NC
490 X=NC-J
500 R=R+X*P(J)
510 NEXT J
520 EL=XM*(C-R)
530 GOTO 990
540 IF NC< 9999 GOTO 660
550 PZ=EXP(-RH)
560 P(1)=PZ*RH
570 FOR I=2 TO NN
580 X=I
590 P(I)=P(I-1)*RH/X
600 IF P(I))=1E-5 THEN 610
602 IF IT=0 THEN 640
604 IF IT<>0 THEN 620
610 IT=0
620 NEXT I
630 I=NN+1
640 QL=0
650 GOTO 990
660 IF K<9999 GOTO  880
670 IF RC >=1 AND N=9999 THEN1090
680 P(1)=RH
690 FOR I= 2 TO NC
700 X=I
710 P(I)=P(I-1)*RH/X
720 PZ=PZ+P(I-1)
730 NEXT I
740 X=P(NC)/(1-RC)
750 IF N<9999 THEN X=X*(1-RC:(N-NC+1))
760 PZ=PZ+X
770 PZ=1/PZ
780 FOR I=1 TO NC
790 P(I)=PZ*P(I)
800 NEXT I
810 FOR I=NC TO NN
820 P(I)=P(I-1)*RC
830 IF P(I))=1E-5 THEN 840
832 IF IT=0 THEN 960
834 IF IT <>0 THEN 850
840 IT =0
850 NEXT I
860 I=NN+1
870 GOTO 960
880 P(1)=XK*RH
882 FOR I=2 TO NC
883 X=I
885 Y=K-I+1
887 P(I)=P(I-1)*RH*Y/X
890 PZ=PZ+P(I-1)
892 NEXT I
900 FOR I=NC TO NN
910 X=K-I+1
920 P(I)=P(I-1)*RC*X
930 PZ=PZ+P(I)
940 NEXT I
950 GOTO 350
960 QL=RC*P(NC)/(1-RC):2
970 IF N<9999 THEN QL=QL*(1-RC:(N-NC)-(XN-C)*RC:(N-NC)*(1-RC))
980 EL=XL*(1-P(NN))
990 SL=QL+EL/XM
1000 WS=SL/EL
1010 WQ=QL/EL
1020 MX=I-1
1030 LPRINT "ARR.RATE=";XL;TAB(30)"EFF.ARR. RATE=";EL
1035 LPRINT "SVC. RATE=";XM;TAB(30)"TRAFFIC INTENSITY=";RH
1037 LPRINT
1040 LPRINT "AVG. TIME IN SYS.=";WS;TAB(35)"AVG. TIME IN LINE=";WQ
1045 LPRINT "AVG. NO. CUST. IN SYS.=";SL;TAB(35)"AVG. NO. CUST. IN LINE=";QL
1047 LPRINT
1050 LPRINT "LIKELIHOOD OF EXACTLY N (FOR N=0 TO";MX;") CUSTOMERS IN THE SYSTEM,"
1055 LPRINT PZ;
1059 K=0
1060 FOR I=1 TO MX
1061 K=K+1
1070 LPRINT P(I);
1071 IF K=6 GOTO 1075
1072 GOTO 1080
1075 LPRINT CHR$(9):K=0
1080 NEXT I
1081 LPRINT:LPRINT:LPRINT
1085 GOTO 20
1090 PRINT "ARR. RATE=";XL;TAB(30) " EFF. ARR. RATE=";EL
1091 PRINT "SVC. RATE=";XM;TAB(30) "TRAFFIC INTENSITY=";RH
1092 PRINT:PRINT "INVALID DATA"
1100 PRINT
1110 GOTO 20
1120 END
```

*Program Listing*

The results are identical if the service rate is reduced instead of increasing the arrival rate (by using a less skilled server or by adding tasks to the service function). For example, if the arrival rate remains at 20 per hour as in Fig. 1, and the service rate is reduced to 24 per hour, the traffic intensity will be 0.833—exactly as shown in Fig. 2. The remaining results, as well, are identical.

Fig. 3 shows statistics if only 10 customers can wait. The effective arrival rate reflects the fact that customers are turned away or leave if waiting area is not available. The results in Fig. 3 are close to the results in Fig. 1; the likelihood of ten or more persons calculated previously was quite small (0.16). The analyst would conclude that planning a waiting area for 10 customers is quite conservative, but accommodates nearly all likely conditions.

Fig. 4 displays results for a service rate of only 10 customers per hour. The traffic intensity exceeds one, only possible in a waiting line model where waiting area or calling population is limited. Customers in this system spend a far greater amount of time waiting and there is almost no chance of finding the server idle. The effective arrival rate indicates a high number of customers turned away due to inadequate waiting area.

Consider the case where the calling population is limited. The results in Fig. 5 indicate that with a calling population of 100, the service is so slow customers must nearly always wait. Continuing this analysis with reduced values for the calling population does not change the results significantly. This lack of change is due to the high arrival rate even with a small calling population.

A final example illustrates the case of added servers. Using the data in Fig. 5, but adding a server, yields the data in Fig. 6. There is a reduction in waiting time, but the line remains long. Clearly an improvement in service time is necessary to produce meaningful change in the results.

Further investigation is limited only by the analyst's imagination. An interesting model is the (M/M/N)-(GD/N/N) case where there is a server for each customer, as in a self-service facility.

Other analyses may be found in any basic operations research text.

The text by Saaty (see References) is a classical treatment of the theory of waiting lines. ■


*References*

*Saaty, T.*, Elements of Queuing Theory, *New York: McGraw-Hill, 1961.*

*Shamblin, J.E. and G.T. Stevens, Jr.*, Operations Research, a Fundamental Approach, *New York: McGraw-Hill, 1974.*

*Taha, H.A.*, Operations Research, an Introduction, *New York: MacMillan, 1971.*


```
(M/M/ 1 )-(GD/ 10 / 100 )

ARR. RATE = 20                        EFF. ARR. RATE = 10
SVC. RATE = 10                        TRAFFIC INTENSITY = 2

AVG. TIME IN SYS. = .999448           AVG. TIME IN LINE = .899448
AVG. NO. CUST. IN SYS. = 9.99448      AVG. NO. CUST. IN LINE = 8.99448

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 10 ) CUSTOMERS IN THE SYSTEM.
1.54611E-23 3.09222E-21 6.12259E-19 1.20003E-16 2.32805E-14 4.46986E-12
8.49273E-10 1.59663E-07 2.96974E-05 5.46432E-03 .994506
```

*Fig. 5*

```
(M/M/ 1 )-(GD/ 10 / 9999 )

ARR. RATE = 20                        EFF. ARR. RATE = 9.99512
SVC. RATE = 10                        TRAFFIC INTENSITY = 2

AVG. TIME IN SYS. = .900978           AVG. TIME IN LINE = .800978
AVG. NO. CUST. IN SYS. = 9.00537      AVG. NO. CUST. IN LINE = 8.00586

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 10 ) CUSTOMERS IN THE SYSTEM.
4.8852E-04 9.7704E-04 1.95408E-03 3.90816E-03 7.81632E-03 .0156326 .0312653
.0625305 .125061 .250122 .500244
```

*Fig. 4*

```
(M/M/ 2 )-(GD/ 10 / 100 )

ARR. RATE = 20                        EFF. ARR. RATE = 20
SVC. RATE = 10                        TRAFFIC INTENSITY = 2

AVG. TIME IN SYS. = .499445           AVG. TIME IN LINE = .399445
AVG. NO. CUST. IN SYS. = 9.98889      AVG. NO. CUST. IN LINE = 7.98889

LIKELIHOOD OF EXACTLY N (FOR N = 0 TO 10 ) CUSTOMERS IN THE SYSTEM.
7.87235E-21 1.57447E-18 1.55872E-16 1.52755E-14 1.48172E-12 1.42245E-10
1.35133E-08 1.27025E-06 1.18133E-04 .0108683 .989012
```

*Fig. 6*

# ANNOUNCING

the

# UCSD p-System[*]

for your

# TRS-80 Model II or III[§]

Now you can write programs in

Pascal FORTRAN BASIC

and run them on

Altos, Apple, Commodore, Cromemco, DEC, IBM, Intertec, Philips, Ohio Scientific, Radio Shack, Terak, Texas Instruments, Vector Graphic, Xerox, Zenith, and many more...without change![†]

**Package includes operating system, compiler, file handler, interpreter, editors, assembler, linker and documentation for $650.00. TRS-80 Model II p-System, Pascal compiler and Corvus (5, 10, 20 Mb) hard disk software for $850.00. Free catalog of p-System application software available. Contact us for details.**

**PCD Systems**
**P.O. Box 143**
**Penn Yan, N.Y. 14527**
**(315)-536-7428**

*trademark of the Regents of the University of California
§trademark of Tandy Corporation
†PCD Supports Systems Software on DEC LSI-II, Radio Shack Models II & III, Altos & Terak only.

✓433

# PLEASE ANSWER THESE 5 QUESTIONS

| | YES | NO |
|---|---|---|
| 1. Do you have information that must be kept organized and accessible? | ☐ | ☐ |
| 2. Do you ever need to perform statistical analyses? | ☐ | ☐ |
| 3. Would you like to have a tool that will allow you greater flexibility in managing your own or your company's money? | ☐ | |
| 4. Could you use a mail list program that will be easy to use, maintain an unlimited number of names, allow you great sorting flexibility, and even interact with a word processor? | | |
| 5. Do you own or have access to a TRS-80 microcomputer? | | |

If you answered "yes" to any of these questions, we can be of assistance to you. Our Maxi Series of applications programs are designed to give you maximum versatility with a minimum of hassle. These programs were created for business use, but you'll find yourself using them for personal applications as well.

**Thorough support** — Of course, each program comes with in-depth, user-oriented documentation, and is menu-driven to make it easy to use. When necessary, the Maxi programs are compatible with each other, and, whenever pertinent, are interactive with the major word processing and spreadsheet programs published by other manufacturers. Also, we maintain a telephone support line to provide you with any assistance you might require.

## Maxi Manager *by Dale Kubler*

Maxi Manager is a remarkable data base manager. Its fast machine language sort complements its large data storage capacity. The sophistication of its data entry, management, and printing capabilities makes Maxi Manager a versatile tool for many applications.

The program now includes Maxi Utility, which allows you to rescue files on diskettes that have been damaged by excessive wear or misuse and lets you expand, add, or delete fields from an existing data base. Let the unmatched capabilities of Maxi Manager handle your data management!

**Model I & Model III. Minimum 1 disk drive required . 012-0196 $149.99**

## Maxi Cras *by Dale Kubler*

(Check Register Accounting System)

Maxi Cras is a system that will computerize check writing, recording and analysis for business and personal finance. The system features 223 income and expense accounts, each of which will handle an unlimited number of transactions. Extensive register and report printing capabilities make Maxi CRAS an indispensible tool for managing money effectively.

**Model I & Model III. Minimum 2 drives required ..... 012-0145 $99.95**

## Maxi Mail *by Dale Kubler*

Maxi Mail is a powerful mail list management system that:

1. Is easy to use.
2. Has virtually unlimited storage capacity.
3. Interfaces with the major word processors to generate form letters and other text.
4. Prints mailing labels up to four-across in any format desired.
5. Has virtually unlimited coding capabilities with thirteen fields of information for each record.

Maxi Mail is the most sophisticated user-oriented mail list program available.

**TRS-80 Model III only. 2 disk drives required ....... 012-0148 $99.95**

## Maxi Stat *by David Walonick*

Maxi Stat is the most useful statistical analysis package on the market today. It was developed to allow maximum flexibility in designing customized analysis. Maxi Stat handles the three main components of statistical analysis:

1. Complete menu-driven codebook creation and editing.
2. Menu-assisted data entry.
3. User-created control files to describe the statistical analyses to be performed and printed out on the variables of your choice.

**Model I & Model III. Minimum 2 drives required .... 012-0153 $199.95**

A Division of Scott Adams, Inc.

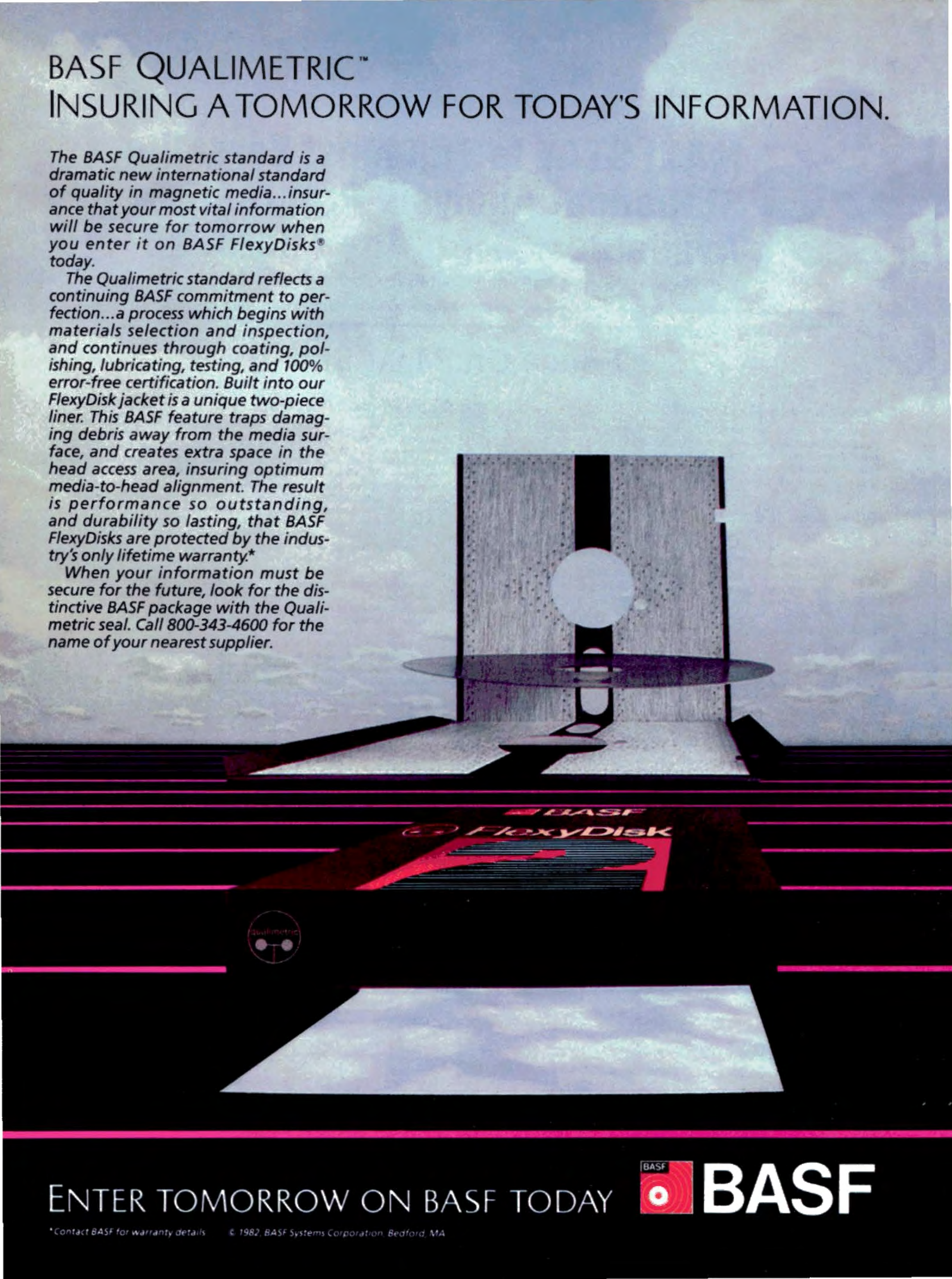
THE BUSINESS DIVISION

Send $1.00 for our 16 page booklet "Getting The Most From Your Micro" All 16 pages are packed with indepth explanations and printout samples from the Maxi Series of applications programs.

**THE BUSINESS DIVISION**
BOX 3435
LONGWOOD, FL 32750
(305) 830-8194

⸙20

*See List of Advertisers on Page 451*


BASF QUALIMETRIC™
INSURING A TOMORROW FOR TODAY'S INFORMATION.
The BASF Qualimetric standard is a dramatic new international standard of quality in magnetic media...insurance that your most vital information will be secure for tomorrow when you enter it on BASF FlexyDisks® today.
The Qualimetric standard reflects a continuing BASF commitment to perfection...a process which begins with materials selection and inspection, and continues through coating, polishing, lubricating, testing, and 100% error-free certification. Built into our FlexyDisk jacket is a unique two-piece liner. This BASF feature traps damaging debris away from the media surface, and creates extra space in the head access area, insuring optimum media-to-head alignment. The result is performance so outstanding, and durability so lasting, that BASF FlexyDisks are protected by the industry's only lifetime warranty.*
When your information must be secure for the future, look for the distinctive BASF package with the Qualimetric seal. Call 800-343-4600 for the name of your nearest supplier.
BASF
FlexyDisk
ENTER TOMORROW ON BASF TODAY
BASF
*Contact BASF for warranty details
© 1982, BASF Systems Corporation, Bedford, MA

LOAD 80

*A marathon for mice.*

# Maze XIX

*Gary Teter*
*6692 Shay Lane*
*Paradise, CA 95969*

Have you ever wondered how a mouse running a maze feels? Now you have a chance to find out: Put on your mouse ears and get ready to play Maze XIX!

**To Play**

After entering the program in the listing, hook up an amplifier (or a modified tape recorder) for sound. After the Hall of Fame display press Enter to begin the game.

The object of Maze XIX is to guide a moving dot through the maze and back as quickly as possible. The arrow keys direct the dot. Running into the wall adds 20 points to your score.

If you need to leave the computer while the game is running, press Clear to halt the game. To continue press any arrow key. To end the game, hold down the Spacebar.

**The Program**

The program is fairly straightforward. The sound routine, by Leo Christopherson (*80-US*, May/June 1980), is POKEd into S$ in line 950.

I included a checksum routine, making it almost impossible to bomb the program with a typo in a data statement. Line 930 checks the length of S$ to make sure it is 74. The only way I can think of to bomb the program is to exchange a couple of numbers in the data statements (please do not take that as a challenge!).

All constants in the dot-move routine (lines 40 and 50) are defined as variables. This makes the game run faster because variables are placed into a table. Those at the top (the ones defined first) are referenced faster than those at the bottom (and also faster than constants). For this reason I defined the variables used in lines 40 and 50 at line 910.

The program should run on other DOSes but I have not tested it on any. For all you program modifiers, Table 1 is a listing of each section, along with what each part does. ■

*Gary is a 15-year-old high school junior. He is a founding member (and presently general manager) of the Ficticious GLM Corporation.*

***The Key Box***

**Basic Level II**
**Model I**
**16K to 48K RAM**
**TRSDOS 2.3**

| | |
|---|---|
| 10-20 | Credit given where credit is due |
| 30 | Call initialize variables routine |
| 40-50 | Dot move routine, and check for wall hit |
| 60-70 | Change direction and make bounce sound |
| 80 | Delay until arrow key pressed |
| 90-160 | Ask for another game |
| 170 | Separate the subs from the program |
| 180 | Show titles and make starting noise |
| 190-210 | Draw maze frame |
| 220-280 | Create the openings |
| 290-310 | Reset variables, make sound and start game |
| 320 | Hole routine; choose start of opening |
| 330-360 | Check for conflicts—force player to turn |
| 370-380 | Make the hole and return |
| 390-480 | Display "Play Maze XIX" titles |
| 490-500 | Loop until Enter key is pressed |
| 510-530 | Draw the maze frame |
| 540-560 | Open the paths |
| 580-640 | Blinking cursor routine. Returns Z$ |
| 650 | Center title and print slowly |
| 660-670 | Sound routines |
| 680-700 | Display high scores and wait until Enter |
| 710-830 | Set high score, enter initials |
| 840-1050 | Initialize all variables |
| 9999 | Save updates on disk |

*Table 1*

*Program Listing*

```
10 REM Maze XIX by Gary M. Teter - Copyright (c)1918 by the
20 REM GLM corporation (no rights reserved, free to all)
30 GOTO840
40 S=S+1:IFX>63ANDY<7THENF1=-1ELSEIFX<63ANDY<7ANDF1THEN90ELSEC=PEEK(P):IFCA
NDL,E=-3:F=0ELSEIFCANDR,E=3:F=0ELSEIFCANDU,F=-1:E=0ELSEIFCANDD,F=1:E=0ELSEI
FCANDGGOTO80ELSEIFCANDHTHEN110
50 X=X+E:Y=Y+F:IFPOINT(X,Y)THEN60ELSERESET(X-E,Y-F):SET(X,Y):GOTO40
60 S=S+20:W=W+1:E=-E:F=-F:X=X+E:Y=Y+F:POKES0+6,1:POKES0+16,RND(50)+100
70 O=USR(0):GOTO40
80 C=PEEK(P):IFCANDUORCANDRORCANDLORCANDDTHEN40ELSE80
90 CLS:GOSUB670:PRINTCHR$(23):PRINT:PRINT:FORIX=1TO10:IFS<S(IX),GOSUB710:GO
TO110ELSENEXT
100 PRINT"Your score was"SCHR$(8)".":PRINT"You crashed"W"time";:IFW<>1,PRIN
```

*Listing continues*

*Listing continued*

```
T"s."ELSEPRINT"."
110 IFCANDHCLS:PRINTCHR$(23)
120 PRINT:PRINT:S=0:W=0:PRINT"Play again? ";
130 GOSUB580:IFZ$="N"ORZ$="n"THEN180
140 IFZ$="Y"ORZ$="y"THENPRINTZ$;:ELSEGOTO130
150 PRINT:PRINT"Use the same maze? ";
160 GOSUB580:IFZ$="Y"ORZ$="y"THENPRINTZ$;:GOSUB510:GOSUB290:GOTO40ELSEIFZ$=
"N"ORZ$="n"THENPRINTZ$;:GOSUB190:GOTO40ELSE160
170 END
180 GOSUB680:GOSUB670:FORIY=1TO10:GOSUB660:NEXT
190 CLS:PRINT@64,B$;
200 FORIX=2TO14:PRINT@IX*64,Z1$;:NEXT
210 PRINT@0,X$;:PRINT@960,Y$;:POKE16383,191
220 A=0:B=63:Y=4:FORZ=1TO13:Y=Y+3
230 GOSUB320
240 NEXT
250 POKE16351,176:POKE16287,143:POKE16352,176:POKE16287,143:POKE16288,141
260 A=64:B=125:Y=4:FORZ=14TO27
270 Y=Y+3:GOSUB320
280 NEXT
290 X=3:Y=3:E=3:F=0:F1=0
300 POKES0+6,2:POKES0+34,4:O=USR(0)
310 GOTO40
320 IFA=0THENPL=RND(49)+2ELSEPL=RND(47)+66
330 IFPL=PVTHEN320ELSEIFPL+8=PV+8THEN320ELSEIFPL<PVANDPL+9>PVTHEN320
340 IFPL>PVANDPL+8<PV+8THEN320ELSEIFPL>PV+8THEN360
350 IFPL>PVTHEN320
360 IFPL+8>B-3THEN320
370 PV=PL:FORX=PLTOPL+8STEP2:RESET(X+1,Y):RESET(X,Y):NEXT:PL(Z)=PL
380 RETURN
390 GOTO680
400 CLS:P$="Play":PRINTCHR$(23):GOSUB650
410 IFINKEY$=CHR$(13)THEN500
420 P$="Maze XIX":PRINT:GOSUB650
430 IFINKEY$=CHR$(13)THEN500
440 P$="* 1 quarter - 1 play *":PRINT:PRINT:GOSUB650
450 IFINKEY$=CHR$(13)THEN500
460 P$="* 2 quarters - 3 plays *":PRINT:GOSUB650
470 IFINKEY$=CHR$(13)THEN500
480 PRINT:PRINT:P$="Press -ENTER- to begin":PRINT:PRINT:PRINT:GOSUB650
490 IFINKEY$=CHR$(13)THEN500ELSEIX=IX+1:IFIX>200THENIX=0:GOTO680ELSE490
500 RETURN
510 CLS
520 PRINT@64,B$;:FORIX=2TO14:PRINT@64*IX,Z1$;:NEXT:PRINT@0,X$;
530 PRINT@960,Y$;:POKE 16383,191
540 A=0:B=63:Y=4:FORIY=1TO13:Y=Y+3:GOSUB570:NEXT
550 POKE16351,176:POKE16287,143:POKE16352,176:POKE16287,143:POKE16288,141
560 A=65:B=125:Y=4:FORIY=14TO27:Y=Y+3:GOSUB570:NEXT:RETURN
570 FORIX=PL(IY)TOPL(IY)+8:RESET(IX,Y):NEXT:RETURN
580 IX=0:ZZ$=CHR$(143)+CHR$(24):ZA$=" "+CHR$(24)
590 IX=IX+1:PRINTZZ$;:IFIX>10THENIX=0:GOTO620
600 Z$=INKEY$:IFZ$<>""Z=ASC(Z$):IF(Z>64ANDZ<91)OR(Z>47ANDZ<58)OR(Z>96ANDZ<1
23)ORZ=32ORZ=8ORZ=13THENPRINTZA$;:RETURN
610 GOTO 590
620 IX=IX+1:PRINTZA$;:IFIX>10THEN580
630 Z$=INKEY$:IFZ$<>""Z=ASC(Z$):IF(Z>64ANDZ<91)OR(Z>47ANDZ<58)OR(Z>96ANDZ<1
23)ORZ=32ORZ=8ORZ=13THENRETURN
640 GOTO 620
650 PRINTTAB(16-LEN(P$)/2);:FORK=1TOLEN(P$):PRINTMID$(P$,K,1);:FORTD=1TO45:
NEXT:NEXT:PRINT:RETURN
660 POKES0+6,3:POKES0+55,1:O=USR(0):RETURN
670 POKES0+6,2:POKES0+34,4:O=USR(0):RETURN
680 CLS:PRINTCHR$(23);:PRINTTAB(4)"The Maze XIX Hall of Fame"
690 PRINT:FORIX=1TO9:PRINTTAB(9);IX;CHR$(8)". ";S(IX);TAB(19)"- "S$(IX):NEX
T:PRINTTAB(8);IX;CHR$(8)". "S(IX);TAB(19)"- "S$(IX):PRINT:PRINTTAB(4)"Copyr
ight (c)1910 by the":PRINTTAB(2)"G L M  C o r p o r a t i o n"
700 IFINKEY$=CHR$(13)THENRETURNELSEHJ=HJ+1:IFHJ>275,HJ=0:GOTO400ELSE700
710 PRINT@896,TAB(3)"You have set a high score."
720 PRINTTAB(3)"Please enter your initials"
730 FORIS=9TOIXSTEP-1:S(IS+1)=S(IS):S$(IS+1)=S$(IS):NEXT:S(IX)=S:IT=IX
740 PRINT@0,"";:PRINTTAB(4)"The Maze XIX Hall of Fame":PRINT:FORIY=1TO9:PRI
NTTAB(9);IY;CHR$(8)". ";S(IY);TAB(19)"- "S$(IY):NEXT:PRINTTAB(8);IY;CHR$(8)
". "S(IY);TAB(19)"- "S$(IY);
750 PRINT@64*IT+106,STRING$(3,132);:PRINT@64*IT+106,"";
760 GOSUB580:IF(Z>64ANDZ<91)ORZ=32OR(Z>96ANDZ<123)THEN770ELSE760
770 PRINTZ$;:H$=Z$
780 GOSUB580:IF(Z>64ANDZ<91)ORZ=32OR(Z>96ANDZ<123)THEN790ELSEIFZ=8THENPRINT
CHR$(8);:H$=LEFT$(H$,1):GOTO760ELSE780
790 PRINTZ$;:H$=H$+Z$
800 GOSUB580:IF(Z>64ANDZ<91)ORZ=32OR(Z>96ANDZ<123)THEN810ELSEIFZ=8THENPRINT
CHR$(8);:H$=LEFT$(H$,2):GOTO780ELSE800
810 PRINTZ$;:H$=H$+Z$
820 GOSUB580:IFZ=8THENH$=LEFT$(H$,2):PRINTZ$;:GOTO800ELSEIFZ=13THEN830ELSE8
20
830 S$(IT)=H$:PRINT@768,CHR$(31);:IFLEN(H$)<>3THENH$="":GOTO750ELSERETURN
840 DATA 33,1,2,14,255,62,2,254,2,40,21,254,3,40,38,17
850 DATA 13,100,237,97,67,16,254,237,105,67,16,254,21
860 DATA 32,243,201,17,1,5,123,237,97,71,16,254,237,105
870 DATA 71,16,254,60,32,243,21,32,239,201,17,100,2,123
880 DATA 237,97,71,16,254,237,105,71,16,254,61,32,243,21
890 DATA 32,239,201,34
900 CLEAR500:DEFINTA-Z:RANDOM:DIMPL(40)
910 S=0:X=5:Y=3:C=0:P=15100:E=0:F=0:L=32:R=64:U=8:D=16:G=2:H=128:F1=0
920 S$="12345678901234567890123456789012345678901234567890123456789012345678901234"
930 IFLEN(S$)<>74THENPRINT"* Length of S$ in line 940 is not correct - shou
ld be 74 *":STOP
940 S1=PEEK(VARPTR(S$)+1):S2=PEEK(VARPTR(S$)+2):S0=S2*256+S1
950 RESTORE:FORN=0TO73:READD1:POKES0+N,D1:C=C+D1:NEXT:IFC<>7759THENPRINT"*
Checksum error - check data in lines 850 to 900 *":STOP
960 IFPEEK(16396)=201POKE16526,S1:POKE16527,S2ELSEDEFUSR0=S0:CMD"T"
970 A$=CHR$(191)+CHR$(149):B$=A$+STRING$(29,32)+A$+STRING$(29,32)+CHR$(170)
+CHR$(191)
980 C$=CHR$(191)+CHR$(151)
990 X$=C$+STRING$(29,131)+C$+STRING$(29,131)+CHR$(171)+CHR$(191)
1000 Y$=CHR$(191)+CHR$(181)+STRING$(29,176)+CHR$(191)+CHR$(181)+STRING$(29,
176)+CHR$(186)
1010 ZA$=CHR$(191)+CHR$(157)
1020 Z1$=ZA$+STRING$(29,140)+ZA$+STRING$(29,140)+CHR$(174)+CHR$(191)
1030 FORIX=1TO10:S$(IX)=STRING$(3,132):S(IX)=9999:NEXT
1040 GOTO100
1050 END
9999 PRINT"Saving updates.":PRINT"MAZE19/BAS":SAVE"MAZE19/BAS":END
```

*Save money by reloading ribbon cartridges.*

# Open Cartridge Surgery

*Bill Grout*
*380 Channing Way #171*
*San Rafael, CA 94903*

I have a Daisy Wheel Printer II from Radio Shack. It's a wonderful machine, but the ribbon cartridges are longer and flatter than those for other machines, so you can't substitute any other cartridge. These cartridges seem too valuable to throw out when the ribbon runs dry after only four or five hours of printing.

If you have a printer that uses expensive and hard-to-find ribbon cartridges, it is possible to reload them with the ribbons from cheaper cartridges.

First, check the ribbon width of your cartridge. (In my cartridge it was ¼ inch wide.) Once you know the size and kind of ribbon you want, go to the office supply store. I bought a smaller Qume cartridge with the right ribbon, though different in shape from my cartridge. If you are selective you may be able to find a cartridge for dollars less.

It would be easier to buy a spool of ribbon to put into the old cartridge, but I found none available. So I was forced to buy a foreign cartridge and then transfer the ribbon to my old cartridge.

Take the new and old cartridges into the kitchen. On your way, get a table knife and a roll of Scotch tape.

Carefully open your old ribbon cartridge. Remove the rubber band from the knob and axle; the rubber band turns a spool that winds up the used ribbon. Hold the cartridge in your hand and find the seam where the two halves of the plastic case are snapped together. They aren't glued, so slide the knife blade into the seam and work your way around the case until you can lift the top half off.

Examine the inside of the cartridge. (See Fig. 1.) Find the spool of used ribbon on the right. Running back from the spool, the ribbon threads through two gears, around a post, and out the front of the cartridge. Notice how the ribbon threads around the different gears and rollers so you can replace the ribbon correctly. Tracing the ribbon back, you'll see that it goes into the other arm of the cartridge, around a plastic guide with a piece of sponge and a spring attached to it, and then around two rollers in an S fashion. Finally it goes to the empty spool on the left side. You'll put in a new ribbon where the empty spool is, then thread it back to the other spool that holds the used ribbon.

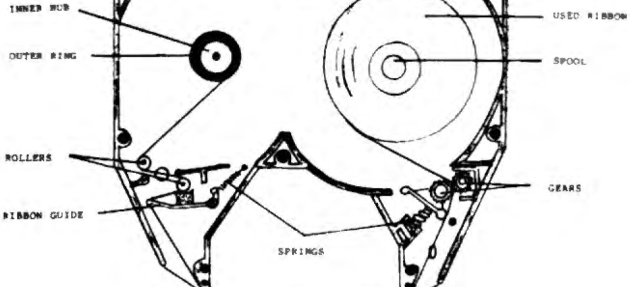



*Figure 1*

Remove the old ribbon and throw it away. Make sure you save the two center hubs or spools (you'll need them later). You have now successfully gutted your cartridge.

Open the new cartridge and detach the ribbon from the other empty spool. Then place the new ribbon onto the left axle of the old cartridge.

Here is where the first problem may arise.

The center hole in the spool may be a different size than the axle in your old cartridge. In my case the hole was too large. Examine the old spool, which fits the axle (it consists of a black outer ring and an inner hub). The outer ring slides off, and the inner hub fits into the hole of the new spool.

The second time I tried this the old inner hub didn't fit. You have two alternatives; you can either wrap a piece of paper around the axle (make sure the spool turns freely) or just let it rattle.

When the ribbon is on the axle, make sure it is unrolling in the right direction. As the spool unwinds, it should turn clockwise. Rethread the ribbon through the rollers and the guide, then out the arm and into the other arm. Fit the ribbon between the two gears that are pressed together by a spring by depressing the gear with the spring to make a small gap between the gears. I found it easier to remove one of the gears completely, position the ribbon and then replace the gear.

All that remains is to attach the loose end of the ribbon to the old spool. Make sure the ribbon is not twisted. When the ribbon winds up, the spool turns clockwise. Use a small piece of tape to attach the ribbon to the left side of the spool, and make sure the ribbon is properly aligned. Now fit the top half of the cartridge back on, and press the sides together tightly. This is important at the ends of the two arms, for if they are not firmly together, the ribbon may slide between the seams, catch and snap. With the top of the cartridge back on, replace the rubber band on the knob and axle.

Pull the ribbon to see if it unrolls freely. If it doesn't, the top half of the cartridge may be pressing down on the spool. Take the table knife and slip it into the seam again to free the spool inside. Once the ribbon comes out with a gentle pull, you're ready to replace the cartridge on your printer. ■

*Here's an explanation of their functions.*

# NEWDOS80 Files

*Randy M. Resnick*
*162 rue de la Convention*
*75015 Paris, France*

Remember your innocent days when above all you preferred to read comic books? The styles varied drastically between Superman and Donald Duck, but all comics shared one convention. When the villain was foiled or Dewey slipped on a banana peel, they shook their fists and yelled *#%&!#!!!

I suspect that when you got to the end of the NEWDOS80 manual and started looking at the documentation for the five new file types, many of you shook your fists and uttered even more sophisticated oaths. Assuming that you have at least glanced over Appendix A in the manual or Section 8 on Disk Basic I/O enhancements, I propose to lead you to the light at the end of the tunnel.

It is not the fault of the authors of this fine operating system that standard ASCII just does not have enough unique characters for all of NEWDOS 80's functions.

In Basic Level II and up, the four characters %, !, # and $ are variable type declarations. Most of you know and use the $ suffix for a string variable as in A$, C1$ or P$(1,4). Likewise, % designates an integer variable, ! means single precision, and # means double precision. You do not always see all these symbols in a program because they can also be defined at the beginning by a line like this:

```
10 DEFINT A-F:DEFSTR R-Z
```

You knew all that. So did I and so did Apparat. They had little choice, however, but to assign additional uses to these same characters. What do you make of this, for example:

```
70 GET 1,#,,(10)#,L%,R#;
```

Not only do these symbols have more than one meaning, but they can appear three times in one line and represent three different things.

When I first examined the examples in the book, I found them very confusing. With a bit of study and practice, however, the different meanings, in context, are as obvious as the difference between two, too and to in spoken conversation.

### GETting to the Point

Some file types require that you use a declaration symbol for all variables, and in some you must indicate the exact length of a string. Unlike Basic, there is no default variable. (With added power comes added responsibility.)

In line 70 above L% and R# are simply the two variables, integer and double precision respectively, that the statement wants to GET.

What's between the GET 1 (1 is the buffer number, File Area Number in NEWDOS80 terms) and the variables? We'll get to that in a moment. First, look at this elegant model:

```
10 OPEN "0", 1, "FILE", "FI"
20 PUT 1,,,A%;
30 CLOSE
```

The FI file type, threateningly referred to as "fixed-item, user-segmented," must be hard since it's the last thing in the book, right? Wrong! All five new file types require that you inform the system of the type you are using. So, lines 10–30 opened an FI file, wrote one integer value to it, and closed it. We could have written several values in a statement such as this one, and many thousands with the proper loop surrounding it. No fielding is involved. We wrote two bytes of data in LSB-MSB order (also known as internal format because this is how an integer is stored in memory) to the disk.

If A% equaled −12345 and was written to disk using the PRINT#1 statement, it would use seven bytes of disk storage. The minimum number of bytes that a PRINT#1 statement will use for an integer variable is three: one for the sign, one per digit and a trailing blank. Yet, all integers in an FI file use two

> ***The Key Box***
>
> **Basic Level II**
> **Model I**
> **32K RAM**
> **One disk drive**
> **NEWDOS80**

bytes. It is easy to see that if you wrote 1000 integers to a regular sequential file, you could waste an average of two full tracks (4096 bytes) of disk space.

I rewrote a program that stored a 3 by 1024 element integer array to disk, changing the file type from PRINT#/INPUT# sequential to FI by modifying four statements. The original file this program created was 36 sectors long and the new one with the same data took 24 sectors. Most data values in the array were zero; all were limited to three digits. If the array were full of two to five-digit integers, the difference could be more than 300 percent—better than triple density!

Examining line 20 of my program gives us the syntax for the GET and PUT commands for NEWDOS80 files. After the GET command and its buffer number and comma are two more commas, the variable name and a semicolon. These last two items have a name: IGEL (Item Group Expression List). An IGEL is much easier to tame than a Field statement. It is simply a list of variables, each type-declared, separated by commas and ending with a semicolon. An IGEL can also exist by itself in a line or several lines such as:

```
145 GOTO 155
150 A%, B%, G$:' three integers
151 L$(X,1), R#;' string and numeric
155' program continues . . .
```

Your program *must* branch around the items in the IGEL. See the manual for more examples and how to use comments in the lines of an IGEL to enhance readability. If the IGEL has its own line number, it is called a free-standing IGEL. To use one in a program, the syntax is: 20 GET 1,,150. There is no ending semicolon, and only one comma is used after the comma that comes with GET 1. You will often use the same IGEL for GET and PUT statements in the same program. This version of line 20 would read the disk data of the file and assign it to the variables named in lines 150 and 151 above, assuming the data is written with exactly the same types of variables in the same order.

If you want to write a sequential file of the FI-type numbers only, and use it like the PRINT#/INPUT# type files, you have the tools to do so. Program Listing 1 is an example.

The first four lines fill the array with 1278 useless numbers between −10000 and +10000. I encourage you to substitute meaningful data for this section, such as the 426 highest scores in Chess Master Version 3.0, the dates they were achieved, and whether they increased in the last game.

The other odd-numbered lines are remarks. Using a free-standing IGEL, the program is easy to write and debug. Compared to a PRINT#/INPUT# type file program, it accomplishes the same task about 28 percent faster, and uses ⅓ the amount of disk space to store the same data.

To use strings in such a file you must state their lengths in the IGEL such as (23)A$ or (5)Q$(X,Y). This number creates a space in the file for the string. If the actual string is longer, it is truncated and, if shorter, the remainder of the space is filled with binary zeros (nulls).

You can randomly access these new file types as well! The MF and FF type files are similar to TRSDOS's random files, except that you decide the record length (1–4096) and you do not need Field or LSET and RSET statements. The system keeps track of record numbers for you. For other file types, you must know where the record you want is located in the file (called its RBA). Further, if you have already worked with random files, you will appreciate the fact that some NEWDOS80 file types can manipulate a part of a record without re-writing the whole record. This can be complex, and I would not try it on payroll files during working hours! Even so, if you work with long records, partial I/O can save access time, and in some cases can avoid the loss of important records.

### File Positioning

If you dream in Assembly language, if you can multiply 16-bit hex numbers (signed or unsigned) in your head, or if you have read this far, you will have no trouble understanding file positioning.

```
100 OPEN "R",1,"FILE","FI"
```

Where are we? At the first byte in the file. If no other control information is specified for a GET or PUT statement executed just after an OPEN statement in modes I, O, R or D, NEWDOS80 processes the data from the first byte of the file. If we asked for data after the EOF we would get an error and program execution would be broken, so the first necessity is a test for the end of the file. (Note: Program abort by error does *not* close the file nor does it necessarily write any data. This is great for troubleshooting, but you should proceed with caution!)

NEWDOS80 has extended the LOC( ) statement to five separate functions. The LOC( )$ function, used with an FI file, returns a zero when it occurs before the EOF (end of file), and a minus one when it is positioned at or beyond the EOF. A test for EOF could be as simple as that of Program Listing 2.

Since LOC( )$ is always zero except at or beyond the EOF, line 120 will fall through the test until

```
3   DIMA%(2,425)
5   FOR X=0 TO 425:FOR Y=0 TO 2
7   A%(Y,X)=RND(20000)−10000:NEXT Y,X
9   '—sequential output of array –
10  OPEN"0",1,"FILE","FI":GOTO 30:' branch around IGEL
20  A%(0,X),A%(1,X),A%(2,X);:'— IGEL —
30  FOR X=0 TO 425:PUT1,,20:NEXT:CLOSE
41  ' program continues . . . .
49  ' — sequential input to array –
50  OPEN"I",1,"FILE","FI"
60  FOR X=0 TO 425:GET1,,20:NEXT:CLOSE
61' No need to test for EOF.
63' we know the file has 426 rec s
```

*Program Listing 1. Storing integers in an FI file*



✓See List of Advertisers on Page 451

```
100  OPEN"I",1,"FILE","FI",C=1
110  GET 1,,,Z%;
120  IF LOC(1)$ THEN 140
130  C=C+1:GOTO 110
140  CLOSE
150  PRINT"Last Item Processed, Rec ";C
```

*Program Listing 2. EOF test in FI file*

```
1180  OPEN"R",1,"FILE","FF",12
1190  A$="hello test":B%=100
1200  PUT 1,,,(10)A$,B%;
1210  GET 1,#,,(5)D$;
1220  GET 1,•,,(5)L$;
1230  GET 1,•,,U%;
1240  CLOSE
```

*Program Listing 3. File positioning within a record*

it becomes true. The exact point where LOC( )$ becomes true depends on the file type and several other functions, including:

LOC( )!—This returns the Relative Byte Address (RBA) of the next item (the one about to be processed) in the file. This item is the next record in record segmented files or the next byte in non-segmented files. The data at this location will be processed by the next access. This value is zero before the first access.

LOC( )#—This is the start of the last record processed. It is called the REMRA (REMembered Record Address). If LOC( )! is zero there is nothing to remember, so this value is marked invalid by the system and asking for LOC( )# will create an error.

LOC( )%—This is the current EOF address of the file. When extending files in the R or E modes, this value and the data added to the file may not be written to the disk yet, so in the case of a crash, it is possible to lose all new data. Close updates the EOF on the disk. To force this update without a Close, the statement PUT 1,&,,; ensures sensitive data is written before further processing. It has no effect on file positioning. The form PUT 1,!%,,; positions a pointer to the current EOF from anywhere in the open file.

These last three functions of LOC( ) return byte addresses relative to the first byte of the open file; there are some differences in their functions as well as that of LOC( )$ among the five file types. These differences relate to whether the file is segmented into records. If it is segmented, these RBA and REMBA values indicate the beginning of a record; in unsegmented files, they indicate the next byte in the file. This is the trickiest part of playing with these LOC functions, and requires experimentation. Keep in mind that you will see a lot of error messages. The manual is very explicit about the LOC functions, so take another look at it if you have problems.

**The Easy Way**

If you open an MF or FF file in any of the random modes, you must state the record length in the Open statement. These two file types can be accessed randomly by record number (as in GET 1,45,50 or PUT 1,45,,A%; where you will be moving data at record number 45, either into the IGEL with the line number of 50 or from the IGEL within the same line as in the PUT). The system does not know the record numbers in the other three file types, but you do.

If your data is segmented so all records in the file have the same length, you can easily compute the position of a record just like the system does. Other than making your listings more complicated, there is little reason to do this, since you can let the system handle it, by using MF or FF files. It is good to know the formula to better understand the process and for debugging purposes. Multiplying the number of bytes in the record by the number minus one of the record you wish to obtain gives the relative byte address (RBA) of that record. You subtract one because the first record number is one, but the lowest byte address is zero. That is, in a file of 10-byte records, Record 1 is at (1-1)•10 or zero, Record 2 starts at (2-1)•10 or 10, Record 234 starts at 2330 and so on. This Relative Byte Address (RBA) is designated by the symbol !. Instead of asking NEWDOS80 for LOC(1)! for example, you tell it:

```
230 GET 1,!2330,,;
```

If you asked for LOC(1)!, NEWDOS80 would answer 2330 for an unsegmented file, and in the FF file discussed above (which has a stated record length of 10) would answer 2340, the next record's RBA. Note that this GET did not GET anything. It only positioned the file, because it has a null IGEL in it. The same line with a non-null IGEL or IGELSN (that is IGEL plus SN for Statement Number) would perform a normal GET function. PUT 1,!2330,,A%; would indeed write the two byte form of the value of A% starting at the relative byte 2330.

If your data structure is not this simple, you can build an index file while you are writing the data. The first item written will always be at !0, the first byte in the file. After the first PUT, you could ask for the value of LOC(1)! and store it in an array. When the entire file is written, your array contains the RBA or Relative Byte Address for each record, regardless of the varying lengths of these records. You can then store this array to disk as an index file which you can use later for sequential or random access or sorting purposes. Picture the efficiency of loading an array with the record positions from an index file, accessing a single value in a very long record, sorting the values, rearranging the index numbers in this new sorted order and writing them to a new index file. The speed and space gained will amaze you! (See the partial


**SPECIALLY TARGETED.** Works in harmony with your TRS80 Model I or III. No more floppy shuffle. Fast one pass compiler that doesn't generate unnecessary intermediate files which require further assembly. Compatible with TRSDOS, LDOS, DOSPLUS, NEWDOS.

**PROFESSIONALLY DESIGNED.** Crafted by a project team with over 15 years of Pascal programming experience. A powerful Pascal that includes all the features necessary for professional program development including random files and dynamic strings.

**INCREDIBLE SPEED.** Resultant stand alone programs execute 10-20 times faster than the equivalent Basic or up to 4-1/2 times faster than U.C.S.D. programs.

**POWERFUL TEXT EDITOR.** Has full screen capability for easy program or document preparation.

**EXCEPTIONAL DOCUMENTATION.** Includes 250 page documentation pkg. that guides you step by step.

**OPTIONAL ADVANCED DEVELOPMENT PKG.** Includes Optimizer and Native Code Generator for even faster executing programs. ADP: **$125** + shipping.

**ALCOR PASCAL:** A new standard of excellence from the professionals at Alcor Systems.

**CALL OR WRITE** for a free 20 page technical brochure with all the facts. 214-226-4476. Pascal System: **$199** + shipping.

ALCOR SYSTEMS

Alcor Systems
800 W. Garland Ave.
Garland, Texas 75040

| | TRADEMARK |
|---|---|
| TRS 80 | Tandy Corporation |
| UCSD | University Calif San Diego |
| Alcor Pascal | Alcor Systems |

**Dealer Inquiries Invited**

# M I C R O T E R M

More and more hardware and communications services are allowing speeds up to 1200 baud. Soon, some may be going faster than that. Today's terminal software simply can't keep up. But now there is an alternative. Micro-Systems Software introduces MicroTerm, the high speed terminal.

Model III MicroTerm will communicate, without insertion of null characters, at 4800 baud. Guaranteed. No cop-outs, no question. MicroTerm is so fast that you can exit from the terminal to the main menu, adjust video width, open the buffer, turn on the printer, or any one of dozens of other functions, and return to the terminal model **without missing a thing!**

MicroTerm continues to input from the RS232, even while at the main menu. This is the only terminal capable of such an astounding feat. MicroTerm offers you most of the features that "Brand X" smart terminals have, plus it gives you: • Ultra high baud rate operation (up to 9600 in certain cases). • Input while at menu. • Easy to use translation tables. • Easy to use phone number listings. • Maximum auto dial support — most major brands. • Direct file transfer companion program included at no exta cost (compatible with DFT). • DOS commands from menu without exiting program. • Over 34K of capture buffer (in a 48K TRS-80). • Can be set to automatically dial telephone and transmit buffer at preset time without **any** operator intervention.

And many, many more great features. MicroTerm is so fast you must see it to believe it. The various menus are displayed so fast, they seem to jump out at you. Status of various functions can be displayed and altered in split seconds.

For the computerist who wants the ultimate, state-of-the-art terminal software, there **is** no other choice.

MicroTerm retails for $79.95, but registered DOSPLUS owners can purchase it for only $59.95. $20.00 off the retail price! MicroTerm comes complete with the terminal program, the direct file transfer program, some standard translation tables, and documentation.

Don't delay, order yours today! Specify when ordering: Model I or III and whether you want it on 40 or 80 track media. Requires a 16K TRS-80 with one disk drive. We recommend 48K for serious communications work. MicroTerm will be available beginning June 30, 1982.

**MICRO-SYSTEMS SOFTWARE, INC.**

4301-18 Oak Circle
Boca Raton, FL 33431
Telephone: (305) 983-3390 ✓384

record I/O in the manual.)

Regardless of where you are in a file, GET or PUT without explicit positioning will process the next record (if the system knows the record length or the file is marked) or the next item if the file is unmarked. We'll get to marked items in a moment, but first there are a few other symbols to clear up. Refer to Program Listing 3.

Here, the file opened is an FF type file and the stated record length is 12. Do you care to guess the values of D$, L$ and U%? In the first record we wrote a 10-letter string and an integer value. The # in line 1210 sent Basic back to the REMRA or last record processed instead of advancing to the next one, and read the five characters Hello into D$. The * in line 1220 forced the action to remain in the same record and to start at the next byte to read the "test" part of the string. The same command in line 1230 read 100 into U%. If the system is not told the record length the * symbol does not accomplish anything. To read or modify a single item in a record, or to change the structure of the record requires that you know exactly how data is organized. Other partial record techniques include:

- GET 1,,,A%,,C%; The extra comma between variables skips the variable between A and C in marked-item files.
- GET 1,,,(7)$,A%; In fixed-item files, the number of bytes in parentheses is skipped from the current position and A% is read.
- PUT 1,,,(10)#,A%; In fixed-item files, the number of bytes in parentheses is nulled and A% is written. Handle this with care!

**Marked-Item Files**

After all this work, let's let the system handle things for a change. Of course, the system is paid in bytes and marked-items make it work overtime. Another general point: In marked-item files, the PUT statement IGEL can use expressions, like 3•A!/234.3-20 and MID$(D$,2,X + 5), or literals and constants as in PUT 1,,,"abc",12.3;.

The code in Program Listing 4 will create an MU file of five records, each consisting of a string entered from the keyboard followed by a numeric expression related to the string length. The computer prints the record number and waits for you to enter some data. As soon as you hit enter the fifth time, it places this important data on the disk.

After creating the file, examine it with Superzap. The marking bytes within the file are well defined in the manual. In the MU type file, each record

```
1000  CLEAR 1000
1001  CLS:FOR X = 1 TO 5
1002  PRINTTAB(25)"REC";X;:PRINT:LINEINPUTA$(X)
1003  NEXT
1004  OPEN"O",1,"TEST/MU","MU"
1005  FOR X = 1 TO 5;'—— WRITE FIVE RECS TO FILE
1006  PUT1,,,A$(X),5•LEN(A$(X)) + 2;
1007  NEXT
1008  CLOSE
```

*Program Listing 4. Create and write MU file of strings and integers*

```
2000  DIM I(100):'dimension array to max. possible records.
2002  CLS:L = 1:OPEN"I",1,"TEST/MU","MU"
2004  GET1,,,;:I(L) = LOC(1)!:' no data transfer
2006  IFLOC(1)$THEN2010
2008  L = L + 1:GOTO2004
2010  CLOSE:PRINTL;" RECORDS IN FILE"
2012  PRINT:PRINT"REC  RBA":PRINT
2014  FOR X = 0 to L – 1:PRINT X + 1;"    ";I(X):NEXT
2016  PRINT" = = =  EOF = = = = >";I(L)
```

*Program Listing 5. Create index array I( ) with RBA for each record*





✓See List of Advertisers on Page 451

begins with a start-of-record mark (SOR). This adds one byte of overhead for each record, but in working with variable length strings, the strings can be read in order, because the system knows where each one starts. A marking byte is also added to each numerical item in the file to tell the system whether it is integer, single or double precision, and to each string to tell the system its length. Strings exceeding 127 bytes use two marking bytes.

MU type files are easy to work with and flexible but use extra disk space and can be slow. Random processing is possible using the RBA file positioning techniques discussed earlier. The MU file, as the manual states, was intended to replace PRINT#/INPUT# file types and is excellent for data that is widely varying in nature, such as strings 0 to 255 characters long mixed with numbers of varying precision.

Program Listing 5 is a quick demonstration of how to build an index array from the MU file. You could save the index as another file, or use it for processing during program execution. Try it on the five-record MU test file, with the array I( ) containing the RBA values. GET 1,!I(0),,A$,B; obtains the first record, and record N would be accessed using GET 1,!I(N – 1),,;. This allows true random access to a file containing records of varying lengths. The extra marking bytes eliminate the space wasted in field item files. The MU file can be updated as well, but the new record must fit into the same space as the one it replaced.

If you can use strings that are of consistent length (e.g. phone numbers), there are more appropriate file types. For example, if a file contained only strings of seven characters mixed with integers, you might use an MF file setting the record length at eight for the string plus three bytes for each integer, five for single precision, and nine for double precision numbers. You can process this type of file by record number. Again, the more you think out your data format, the easier it will be to choose the right file type for fast access and disk space economy.

If the data is mixed but in an iron-clad format, you can use the MI type file. This file type can not be updated randomly, but it can be read using the intra-record techniques described earlier.

**The Structure of File Types**

Here is the order of how much work the system does, from almost none of it to nearly all the housekeeping:

FI: This file is a continuous stream of data bytes. All control is in the programmer's hands. If you ask for a number and are positioned at a string, the system will calmly return a completely meaningless value. The system can not help you recover. It has no way to detect such an error, unless you try to input past the EOF. These files are very compact but require that you be aware of every facet of the data structure.

FF: All records have the same length, which you must state in the open statement. Expressions are not allowed in the IGELs. The system will handle I/O by record number. Again, the files are very compact, and other than the record length, the system is ignorant of the data structure. An important attribute of this file type is its ability to process a single item in a record without reading or writing the whole record.

MI: The system adds bytes to the data in the file to mark their types. The file can only be written sequentially (or extended), but can be read randomly by file positioning.

MF: This file is like MI but you can update it randomly and access it by record number or file positioning. Expressions are allowed in IGELs. You specify record length in the Open statement.

MU: The system marks all items and the start of each record. This type adds the most extra bytes, but is very flexible with records and strings of varying lengths. Used with an index file containing the RBA of each file SOR, strings of up to 255 bytes can be rapidly accessed randomly and sorted or manipu-

# NEWDOS/80 Version 2.0
# The Support Keeps Coming.


Apparat's newest disk operating system for the TRS-80®, NEWDOS/80 Version 2.0, has added many new enhancements and features to make your Model I or III computer more powerful. We've kept one thing the same. Our support.

Version 2.0 is our second upgrade of our original NEWDOS for the TRS-80. Each version builds and improves on the capabilities of the preceeding versions. Just as important, Apparat's commitment to supporting our products makes a good product even better. By providing our customers with zaps on an ongoing basis, we're continually making NEWDOS/80 Version 2.0 a more powerful tool.

**Version 2.0...**
**High Performance DOS**

NEWDOS/80 Version 2.0 builds even more performance into NEWDOS/80. The versatility and sophistication of Version 2.0 includes features like:

- Double density support on the Model I
- Enhanced compatability between Model I and III
- Triples directory size
- Dynamically merge in basic (also allows merging of non ASC II format files)
- Selective variable clearing
- Can display basic listings page by page
- Automatic repeat function key
- Routing for peripheral handling
- Enhanced disassembler
- Command chaining
- Superzap to scan files
- Fast sort function in basic

These new features, added to the existing capabilities of NEWDOS/80, makes it one of the most powerful additions you can make to your system. And Apparat's commitment to support assures that you've purchased a superior product, both today and tomorrow. At just $149.00 it could be the best investment you will make for your TRS-80.

If you're thinking about upgrading your system, call Apparat today. Dealer inquiries welcome.

**303-741-1778**

TRS-80 is a registered trademark of Tandy Corp.

VISA MasterCard

~40

**Apparat, Inc.**

4401 So. Tamarac Parkway, Denver, CO 80237 (303) 741-1778

*"ON GOING SUPPORT FOR MICROCOMPUTERS"*

lated using this index file. The MU system is very easy to use and understand.

**Final Thoughts**

Hundreds of other details, tricks and tidbits can be learned by experimentation or gleaned from the NEWDOS80 documentation. The program modules included here could be compressed considerably, and there is a lot of room for sophisticated programming techniques once you have mastered these fundamental concepts. I have found them to be well worth the time spent studying them.

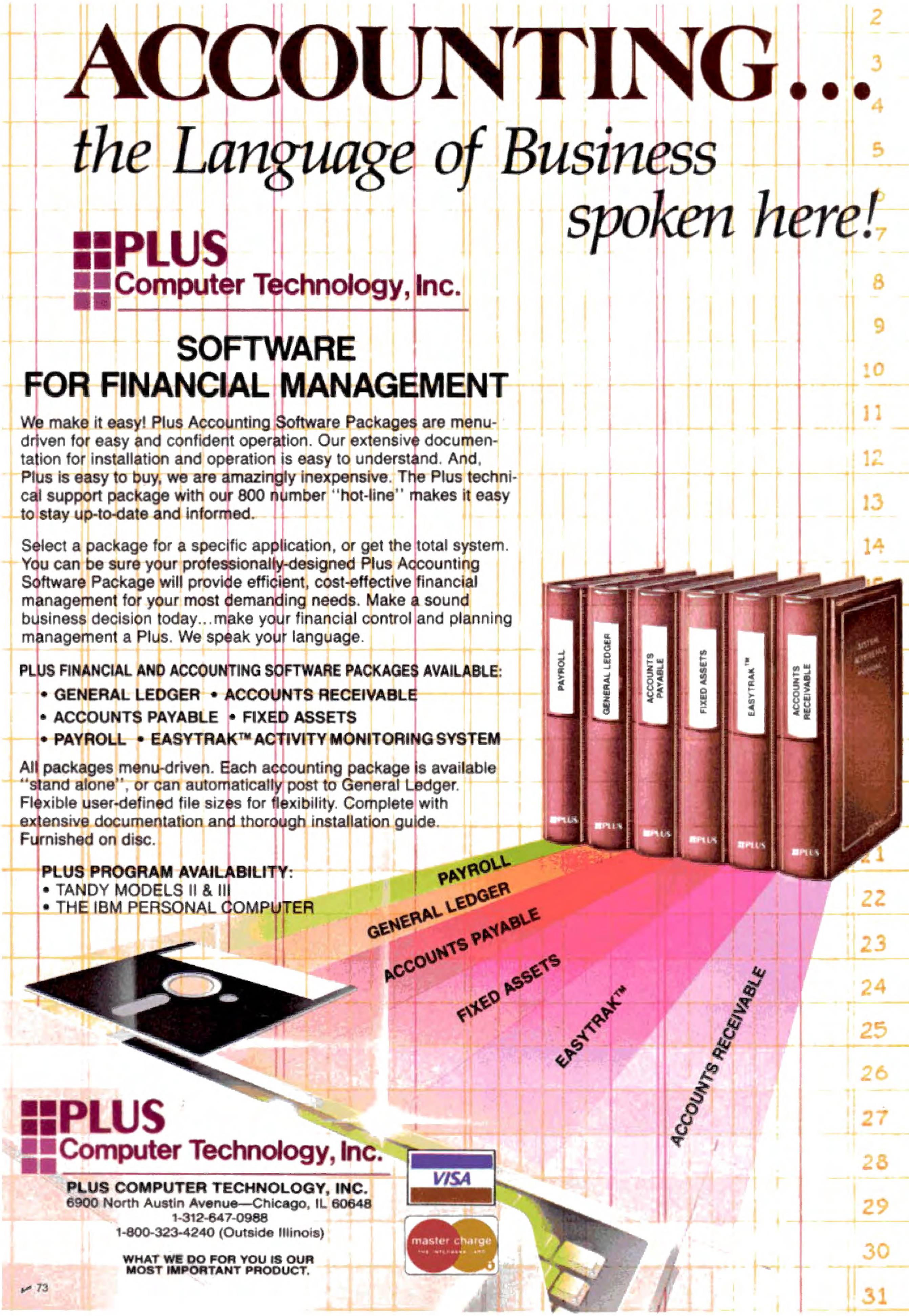
One way to picture the entire file situation in NEWDOS80 is as an analogy to computer languages. In Basic and other high-level languages, you enter words and numbers and manipulate them, without knowing how or where they are stored. The operating system makes the rules, and sometimes uses time analyzing data and worrying about where to put it in memory. Some of us rebel in the face of such a dictatorship. The need to be independent of the operating system's tyranny gives us the desire to direct the computer and run things our own way. To do this we learn to use Assembly or machine language. My first Assembly language program stored and analyzed one string. I had to write the program this way, since that string happened to be 43,548 characters long!

NEWDOS80's new file types give you the power to create your own data structures and use your imagination more freely than do the older (still fully supported) file types. At the very least, you can save disk space and time. The rules are still in the hands of the operating system, but the choice is so varied you hardly notice! ■

*Randy Resnick is a musician living in Paris. He is currently working on laser-graphic systems.*

```
A$ = "THIS IS A STRING IN THE FILE"
B% = 0      C! = -123456      D# = 3.5759458

Writing the above data creates the following disk images:

00  5448  4953  2049  5320  4120  5354  5249  4E47  THIS.IS.A.STRING
10  2049  4E20  5448  4520  4649  4C45  0000  0020  .IN.THE.FILE....
20  F191  0A30  D1C5  4BDC  6482  ----  ----  ----  ...0..K.d.
                      FF and FI Files

00  9C54  4849  5320  4953  2041  2053  5452  494E  .THIS.IS.A.STRIN
10  4720  494E  2054  4845  2046  494C  4572  0000  G.IN.THE.FILEr..
20  7300  20F1  9174  0A30  D1C5  4BDC  6482  ----  s....t.0..K.d.
                      MI and MF Files

00  709C  5448  4953  2049  5320  4120  5354  5249  p.THIS.IS.A.STRI
10  4E47  2049  4E20  5448  4520  4649  4C45  7200  NG.IN.THE.FILEr.
20  0073  0020  F191  740A  30D1  C54B  DC64  82--  .s....t.0..K.d.
                      MU File

00  5448  4953  2049  5320  4120  5354  5249  4E47  THIS.IS.A.STRING
10  2049  4E20  5448  4520  4649  4C45  2C20  3020  .IN.THE.FILE,.0.
20  2D31  3233  3435  3620  2033  2E35  3735  3934  -123456..3.57594
30  3538  200D  ----  ----  ----  ----  ----  ----  58..
                      PRINT# File
```

*Fig. 1. Comparison of the various file formats*



ACCOUNTING...
the Language of Business
spoken here!
PLUS Computer Technology, Inc.
SOFTWARE FOR FINANCIAL MANAGEMENT
We make it easy! Plus Accounting Software Packages are menu-driven for easy and confident operation. Our extensive documentation for installation and operation is easy to understand. And, Plus is easy to buy, we are amazingly inexpensive. The Plus technical support package with our 800 number "hot-line" makes it easy to stay up-to-date and informed.
Select a package for a specific application, or get the total system. You can be sure your professionally-designed Plus Accounting Software Package will provide efficient, cost-effective financial management for your most demanding needs. Make a sound business decision today...make your financial control and planning management a Plus. We speak your language.
PLUS FINANCIAL AND ACCOUNTING SOFTWARE PACKAGES AVAILABLE:
• GENERAL LEDGER • ACCOUNTS RECEIVABLE
• ACCOUNTS PAYABLE • FIXED ASSETS
• PAYROLL • EASYTRAK™ ACTIVITY MONITORING SYSTEM
All packages menu-driven. Each accounting package is available "stand alone", or can automatically post to General Ledger. Flexible user-defined file sizes for flexibility. Complete with extensive documentation and thorough installation guide. Furnished on disc.
PLUS PROGRAM AVAILABILITY:
• TANDY MODELS II & III
• THE IBM PERSONAL COMPUTER
PAYROLL
GENERAL LEDGER
ACCOUNTS PAYABLE
FIXED ASSETS
EASYTRAK™
ACCOUNTS RECEIVABLE
PLUS Computer Technology, Inc.
PLUS COMPUTER TECHNOLOGY, INC.
6900 North Austin Avenue—Chicago, IL 60648
1-312-647-0988
1-800-323-4240 (Outside Illinois)
WHAT WE DO FOR YOU IS OUR MOST IMPORTANT PRODUCT.
VISA
master charge
73

# Color Diskdump

**by Ken Knecht**

**Model I users had it; now Color Computer owners can have it, too. So examine those files.**

I have written a Basic program for the TRS-80 Color Computer that lets you see what is stored in a disk file. The program also reads any area of memory.

Since the Color Computer displays 32 characters on each line, many lines are needed to display a single disk sector of 256 bytes. This program displays the decimal codes of each character in the file with their ASCII equivalent above them—if it is a printable ASCII character. If the ASCII character is a space then no character is shown. If the character is not printable then a period is displayed. You can choose to print the program's output on your printer if you have one.

When you start the program, you are asked for the name of the file. You can also select a display of memory. The file can be any type: direct, sequential, data, Basic program, machine language and so on. Next, you are asked which sector you wish to begin with. The program tells you if this sector is beyond the length of the file. If the sector is valid, as the program continues you have frequent opportunities (every 32 bytes) to restart the program and look at another file or another sector of the same file. When each sector has been displayed, you can restart the program, go to the next sector, or display any sector.

The location of the bytes displayed within the sector is displayed above every eight-byte line together with the sector number. You are always aware of where you are in a file. The sector count begins with one, and the byte count within the sector with zero.

If you choose to read memory, the display is the same, except you see the contents of memory. You are asked for the starting memory address. After each four lines you can continue incrementing memory addresses, start at a new address, or end the program. ■

*Ken Knecht can be reached at 1340 W. 3rd St. #130, Yuma, AZ 85364.*

| *The Key Box* |
| --- |
| **Color Computer** |
| **16K RAM** |
| **Disk Basic** |
| **One Disk Drive** |

```
10 CLEAR1000:P=0:CLS
20 INPUT"OUTPUT TO PRINTER (Y/N)";Z$:IF Z$="Y" THEN P=-2
25 INPUT"READ FROM DISK OR MEMORY (M/D)";Z$:IF Z$="M" THEN 230
30 F$="":PRINT"FILE NAME (PRESS ENTER TO":INPUT"END PROGRAM)";F$
:IF F$="" THEN END
40 INPUT"STARTING SECTOR";S
50 OPEN"D",1,F$:IF S>LOF(1) THEN 210
60 FIELD#1,128 AS A$,128 AS B$:GET #1,S:V=0:SK=0
70 FOR X=0 TO 96 STEP 32:FOR Y=0 TO 31 STEP 8:PRINT#P,"SECTOR";S
;"BYTES";V;"TO";V+7
80 FOR Z=1 TO 8
90 L$=MID$(A$,X+Y+Z,1):L=ASC(L$):IF L<32 OR L>122 THEN PRINT#P,"
  . ";ELSE PRINT#P,"  "+L$+" ";
100 NEXT Z:IF P=-2 THEN PRINT#P," "
110 FOR Z=1 TO 8
120 L=ASC(MID$(A$,X+Y+Z,1)):IF L<10 THEN PRINT#P," "+STR$(L)+" "
;ELSE IF L<100 THEN PRINT#P,STR$(L)+" ";ELSE PRINT#P,STR$(L);
130 NEXT Z:IF P=-2 THEN PRINT#P," "
135 V=V+8:NEXT Y
140 INPUT"CONTINUE (Y/N)";Z$:IF Z$="Y" THEN 150 ELSE 220
150 NEXT X:IF SK=0 THEN SK=1:A$=B$:GOTO 70
160 PRINT"INPUT -1 TO END, 0 FOR NEXT":PRINT"SECTOR, OR THE SECT
OR NUMBER":INPUT"FOR A SPECIFIC SECTOR";N
170 IF N=-1 THEN 220
180 IF N=0 THEN S=S+1:IF S>LOF(1) THEN 210 ELSE 60
190 S=N
200 IF S<=LOF(1) THEN 60
210 PRINT"NO MORE DATA IN FILE"
220 CLOSE:GOTO 30
230 B$="":PRINT"START AT WHAT MEMORY ADDRESS":PRINT"(PRESS ENTER
 TO QUIT, A":PRINT"A POSITIVE NUMBER FOR ADDRESS":INPUT B$:IF B$
="" THEN END ELSE B=VAL(B$)
240 FOR Y=1 TO 4:PRINT#P,"BYTES";B;"TO";B+7:FOR Z=B TOB+7:L=PEEK
(Z)
250 IF L<32 OR L>122 THEN PRINT#P,"  . ";ELSE PRINT#P,"  "+CHR$(
L)+" ";
260 NEXT Z:IF P=-2 THEN PRINT#P," "
265 FOR Z=B TO B+7:L=PEEK(Z)
270 IF L<10 THEN PRINT#P," "+STR$(L)+" ";ELSE IF L<100 THEN PRIN
T#P,STR$(L)+" ";ELSE PRINT#P,STR$(L);
280 NEXT Z:IF P=-2 THEN PRINT#P," "
285 B=B+8:NEXT Y
290 INPUT"CONTINUE (Y/N)";Z$:IF Z$="Y" THEN 240 ELSE 230
```

*Program Listing*

# Forms Maker

by Charles Perelman

Here's an easy-to-use utility that creates forms, such as invoices, on the Model II screen. With the arrow, tab, and repeat keys you can move rapidly around the screen placing keyboard characters where you want them. You can erase or type over any entry, copy the form directly to the printer, or save to disk for future use.

Although written in Basic, the program takes advantage of the Model II supervisor calls with machine-language USR routines. The program was written with TRSDOS 2.0. Twenty-two lines are available for your forms. Instructions appear at the bottom of the screen. Use valid TRSDOS file names when saving forms to disk.

**With USR routines and this Basic utility, you can create business forms on your Model II.**

### Program Overview

The program is modular and self-documenting. USR routines are identified with U variables. Lines 220 and 230 load a Basic patch to implement PEEK for reading the screen. Additional machine language is read in lines 240–320 for the screen-to-printer routine and to prepare registers for the screen-to-RAM and screen-to-disk routines. Data line 12020 contains the PEEK instructions, and line 12030 contains the screen-print routine.

Movement of the cursor with the arrow keys is nondestructive. From the top line, the up arrow moves the cursor to the bottom of the screen in the same column and vice versa with the down arrow from the bottom line. From the right margin, the cursor moves down to the beginning of the next line. Moving backwards, it goes up to the end of the previous line. From the right edge on the bottom line, the cursor moves to the beginning of the same line. Tab erases any characters it passes over.

The error traps go to error-handling routines beginning at 10270. When working with a form, after pressing <CTR L> <P> for printout you must turn on the printer to continue with the program.

Forms Maker lets you make master forms as needed for reproduction on a copier or with a stencil. Revisions are simple and efficient with the masters on disk.■

*Program Listing*

```
10 '******************************
20 'FORMS MAKER
30 'BY CHARLES R. PERELMAN, A PROFESSIONAL CORPORATION
40 '9777 WILSHIRE, BEVERLY HILLS, CA. 90212, 1982
100 '******************************
110 'INITIAL SETUP
120 CLEAR 2000
130 DEFINT L,M,S,U-Y
140 DIM U1(23),U3(12),S(959)
200 '******************************
210 'IMPLEMENT PEEK AND SCREEN PRINTER
220 FOR X=0 TO 23:READ U1(X):NEXT
230  V=0:DEFUSR=VARPTR(U1(0)):V=USR(0)
240 FOR X=0 TO 12:READ U3(X):NEXT
300 '********************************
310 'SET UP REGISTERS FOR USR CALLS
320 U2(0)=15872:U2(2)=8448:U2(4)=4352:U2(6)=256:U2(8)=-13873
500 '*******************************
510 'INSTRUCTIONS
520 CLS:PRINT"MODEL II REQUIRES FILES TO BE OPENED WHEN ENTERING
 BASIC.  IF YOU":PRINT"NEGLECTED TO DO THIS, PRESS <BREAK>, TYPE
 'SYSTEM', PRESS <ENTER>.":PRINT"THEN TYPE 'BASIC FORMSMKR -F:1'
 AND PRESS <ENTER> TO RESTART."
530 PRINT:PRINT"   IF YOU HAVE OPENED A FILE, PRESS <ENTER> TO C
ONTINUE.";
540 Z$=INKEY$:IF Z$="" THEN 540 ELSE IF ASC(Z$)<>13 THEN 540
550 CLS:PRINT@25,"FORMS MAKER":PRINT:PRINT"   AFTER MAKING CHOIC
E FROM MENU, USE ARROW KEYS AND TAB TO POSITION.":PRINT"CURSOR O
N SCREEN.  TYPE IN ANY KEYBOARD CHARACTER ON TOP 22 LINES."
560 PRINT"CHANGE BY OVERTYPING OR ERASE WITH BACKSPACE OR SPACE
BAR.":PRINT"   PRESSING THE <CRTL> KEY AND <X> WILL RETURN YOU T
O THE MENU.":PRINT"<CRTL><D> SAVES THE SCREEN TO DISK, AND <CRTL
```

*Listing continues*

*Charles R. Perelman (9777 Wilshire Blvd., Beverly Hills, CA 90212) is an attorney, a CPA, and a certified tax specialist.*

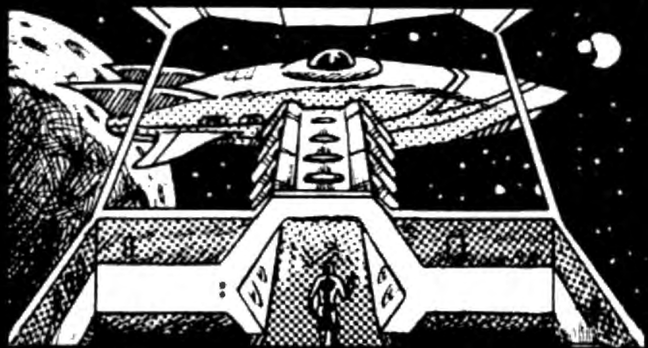
*The Key Box*

**Model II**
**TRSDOS 2.0**
**Model II Basic**

# THE COLORQUEST EXPERIENCE

## For the TRS-80C and the TDP System 100 Color Computers

Written by Kevin Herrboldt & Tim Nelson



**BEYOND THE CIMEEON MOON** A real-time science fiction adventure game of mind-blowing magnitude — staged in deep space on a malign, sentient spacecraft. *Written in fast machine code with 3-dimensional high-res graphics and sound.* **Choose from a myriad of fates. Survive the laser barrage set up by a guantlet of robots bent towards your destruction.**

Written by Kevin Herrboldt & Tim Nelson

**ADVENTURE TRILOGY** A Trilogy of quests *featuring 3-D high-res graphics in machine code.* First comes ritual combat on the **WORLD UNDER THE CIMEEON MOON**, to test your worthiness as a warrior. Once proven, you will be teleported to **DAZMAR'S UNDERWORLD OF DOOM** to search for the Eye of Dazmar. The **FORSAKEN GULCH** is the final arena, where a wicked idol awaits restoration.

**THE NIBBLER & MS. NIBBLER** A fast maze chase game featuring the nibbler man and three bumbling preditors. *Written in machine code and joystick compatible,* this fun packed game is enjoyed by all. **MS. NIBBLER** is similar to THE NIBBLER described above but features a different maze and MS. NIBBLER for the ladies.

**COMBAT GAMEPACK** 3 action packed games featuring lifelike graphics and sound. ***EXTENDED** BASIC **required*** **2-1-0 TANK COMBAT** pits two players against each other in 5 different terrains. **STELLAR BATTLE** lets you pilot a flexwing fighter through deep space fighting dorian squadrons. **GALACTIC** BLOCKADE is a favorite two-player arcade game of speed and skill.

**VEGAS GAMEPACK** The thrills of a Las Vegas casino at home! ***Extended BASIC required.*** **CASINO CRAPS, 21,** ONE **ARMED BANDIT, UP AND DOWN THE RIVER, & KENO.** A bank tracks players winnings from game to game.

## ORDERING

**ALL GAMES ARE $24.95 for 16K Cassette; $29.95 for 32K Disk.**

**Include $3.00 for shipping in the U.S. & Canada, $6.00 for Foreign orders. C.O.D. add $2.00**

**AVAILABLE AT DEALERS EVERYWHERE. IF NOT, ASK WHY!**

**NELSON™ SOFTWARE SYSTEMS**

A Division of Softlaw Corporation

128

VISA MasterCard

9072 Lyndale Avenue So. 612/881-2777

Minneapolis, Minnesota 55420 U. S. A.

*Listing continued*

```
><P> PRINTS HARD COPY.":PRINT:PRINT"PRESS ENTER TO CONTINUE.";
570 Z$=INKEY$:IF Z$="" THEN 570 ELSE IF ASC(Z$)<>13 THEN 570
600 '*****************************
610 'MAIN MENU
620 CLS:PRINT@340,"MAIN MENU":PRINT:PRINT"1.  CREATE NEW FORM":P
RINT"2.  EDIT OLD FORM.":PRINT"3.  RECALL CURRENT FORM FOR EDIT.
":PRINT"4.  END PROGRAM.":PRINT:PRINT"PLEASE PRESS NUMBER FOR CH
OICE.";
630 Z$=INKEY$:IF Z$="" THEN 630
640 IF INSTR("1234",Z$)=0 THEN L=911:M=1:GOSUB 10120:GOTO 630
650 CLS:ON VAL(Z$)GOTO 1020,2020,3020,4020
1000 '***************************
1010 'CURSOR MOVEMENT
1020 PRINT@1840,"PRESS <CRTL> AND:  <X> FOR MENU, <D> TO SAVE TO
 DISK, <P> TO PRINT";:PRINT@0,"";
1030 B$=INKEY$:IF B$="" THEN 1030
1040 IF ASC(B$)=28 OR ASC(B$)=8 OR ASC(B$)=9 THEN PRINT B$;
1050 IF ASC(B$)=31 AND ROW(Z)=21 THEN L=POS(Z):PRINT@L,"";ELSE I
F ASC(B$)=31 THEN PRINT CHR$(255);
1060 IF ASC(B$)=30 AND ROW(Z)=0 THEN L=1680+POS(Z):PRINT@L,"";EL
SE IF ASC(B$)=30 THEN PRINT CHR$(254);
1070 IF ASC(B$)=29 OR ASC(B$)>31 AND ASC(B$)<128 THEN PRINT B$;
1080 IF ROW(Z)>21 THEN PRINT@1680,"";:GOTO 1030
1090 IF ASC(B$)=4 THEN GOSUB 10220:GOTO 1020
1100 IF ASC(B$)=24 THEN W=1:GOSUB 10320:GOTO 620
1110 IF ASC(B$)=16 THEN GOSUB 10420:GOTO 1020
1120 GOTO 1030
2000 '*****************************
2010 'LOAD FORM FROM DISK
2020 PRINT@320,"ENTER NAME OF FILE TO RECALL:  ";:LINEINPUT F$
2030 ON ERROR GOTO 10270
2040 CLS:PRINT CHR$(02):OPEN"I",1,F$:PRINT@1840,"FILE NAME IS: "
;F$
2050 FOR L=0 TO 1680 STEP 80:GOSUB 10520:LINE INPUT#1,A$:PRINT A
$;:NEXT
2060 CLOSE:PRINT CHR$(01):GOTO 1020
3000 '***************************
3010 'RECALL FROM RAM
3020 W=2:GOSUB 10320:GOTO 1020
4000 '****************************
4010 'END PROGRAM
4020 PRINT CHR$(31)+CHR$(26):M=3:GOSUB 10120:PRINT CHR$(30):END
10000 '****************************
10100 'SUBROUTINES
10110 'FLASHING MESSAGES
10120 E$(1)="ONLY 1 TO 4 PLEASE":E$(2)="FILE NOT FOUND":E$(3)="B
YE FOR NOW":E$(4)="TURN PRINTER ON":E$(5)="IMPROPER FILE NAME."
10130 PRINT CHR$(02):FOR X=1 TO 2:PRINT@1760,"*** "+E$(M)+" ***"
;:FOR Y=1 TO 800:NEXT
10140 PRINT CHR$(25):PRINT@1760,SPACE$(80);:FOR Y=1 TO 500:NEXT:
NEXT X:PRINT CHR$(01);:PRINT@L,"";:RETURN
10200 '***************************
10210 'SAVE TO DISK
10220 ON ERROR GOTO 10270
10230 PRINT@ 1760,"ENTER FILE NAME:  ";:LINEINPUT F$:PRINT@1760,
"SAVING FILE    ";
10240 OPEN "O",1,F$
10250 FOR L=0 TO 1680 STEP 80:GOSUB 10520:PRINT#1,A$:NEXT
10260 CLOSE:PRINT@1760,SPACE$(80);:RETURN
10270 IF ERR=53 THEN M=2:GOSUB 10120:RESUME 620
10280 IF ERR=52 THEN PRINT:PRINT:PRINT"FILES NOT OPENED WHEN BAS
IC ENTERED.  YOU MUST START PROGRAM AGAIN.":PRINT"AT TRSDOS READ
Y, ENTER 'BASIC FORMSMKR -F:1'":FOR X=1 TO 1000:NEXT:SYSTEM
10290 M=5:GOSUB 10120:RESUME 620
10300 '***************************
10310 'SCREEN TO RAM
10320 V1=0:U2(1)=94:U2(3)=VARPTR(S(0)):IF W=1 THEN U2(7)=-1 ELSE
 U2(7)=0
10330 DEFUSR2=VARPTR(U2(0)):V1=USR2(0):RETURN
10400 '******************************
10410 'EXECUTE SCREEN PRINT
10420 M=4:GOSUB 10120
10430 V3=0:DEFUSR3=VARPTR(U3(0)):V3=USR3(0):C$=INKEY$:RETURN
10500 '******************************
10510 'READ SCREEN TO A$ TO SAVE OR LOAD
10520 V1=0:U2(1)=11:U2(7)=CVI(CHR$(L-INT(L/80)*80)+CHR$(INT(L/80
))):U2(5)=20480:A$=SPACE$(80):U2(3)=CVI(CHR$(PEEK(VARPTR(A$)+1))
+CHR$(PEEK(VARPTR(A$)+2))):DEFUSR2=VARPTR(U2(0)):V1=USR2(0):RETU
RN
12000 '**********************
12010 'MACHINE LANGUAGE USR ROUTINES
12020 DATA -13023,8797,26623,17441,8830,26625,-15583,8955,26627,
14910,1330,104,-12255,8773,10757,17697,8779,10759,-15583,8959,23
259,26430,-8910,-13990
12030 DATA 33,1784,-15082,-32450,-45,20486,3342,4296,4559,80,-16
103,-4848,201
```

**The Key Box**

**Model I or III**
**16K RAM**
**Cassette or Disk Basic**

*Your routines can become efficient.*

# Screen Strings

*Don Davidson*
*2413 Main Road*
*Silver Creek, NY 14136*

String variables play an important role in almost every program. Learn how your micro handles the strings and you can make your routines more efficient.

### String Variables

When a string variable is put into a Basic program, the computer's type, name, length and address are stored in memory in the variable list table just above the program. Each string record occupies six bytes in the table (Fig. 1). The decimal values in the figure are for A$, five bytes long, at decimal memory address 17143 (247 + 256*66). The values for any variable can be found using the VARPTR command.

Enter the short demonstration in Program Listing 1. Line 30 finds the location of the record for A$ in the variable list table. Line 40 gets the length of the string, 50 obtains the address of the string and 70 converts this address to a decimal number.

Run the program. Note the addresses of the string and its variable pointer. The string is located about fourteen bytes above the beginning of the program. The pointer is about 250 bytes above that, depending on your spacing.

Edit line 10 to read: 10 A$ = "ABCDE" and run it again. The string length changes to five, as you might expect. The string address stays the same, while the pointer address increases by two (we used two more bytes in the program).

Enter as a direct statement (no line number): POKE VP,3: PRINT A$.

The result is ABC. We changed the length of the string by changing the number stored in the address found by VARPTR. Next, if the LSB (least significant byte) of the string address is less than 254, try another direct statement: POKE VP + 1, LSB + 2: PRINT A$.

This adds two to the address in the table telling where the string is stored. The result should be CDE. So, the computer can be told the length and address of any string by POKEing the values into the variable list table.

One more point. Change line 10 again: 10 A$ = CHR$(65).

This time A$ = A, but its address is the last byte in your computer's memory. Any time a CHR$ or STRING$ is used in a string, or when two or more items are combined with a + to form a string, the computer assembles it in reserved string space in high memory.

Try: 10 FOR I = 65 TO 69: A$ = A$ + CHR$(I): NEXT.

The location of A$ will be 20 bytes below the end of computer memory. What is in those last twenty bytes besides our 5-byte string? We can PEEK them to find out. But wait a minute! A$ already points to the beginning of the 20 bytes. It is simpler to use a direct statement again: POKE VP,20:PRINT A$.

The result is ABCDEEABCD DABCCABBAA.

Reading from right to left, the computer interpreted CHR$(I), put it in memory, then added it to the existing A$, which it also put in memory. It repeated this for five loops, each time using the next lower space in the reserved string area specified by the CLEAR nnn statement. In this instance, we did not reserve any, so the computer set aside 50 bytes for us.

### Defining a Screen String

The video screen is memory mapped in addresses 15360–16383: a string can be created in screen memory. Make the following changes in the program and run it.

```
10 A$ = ""
35 POKE VP,64: POKE VP + 1,0: POKE
   VP + 2,60
80 PRINT A$
```

Line 10 creates A$ in the variable list table. Line 35 makes it 64 bytes long, beginning at the screen's upper left (0 + 256 * 60 = 15360 decimal). Now anything on the first line of the video screen is A$ and is printed. To show this more vividly enter: FOR I = 1 TO 15:PRINT A$;:FOR

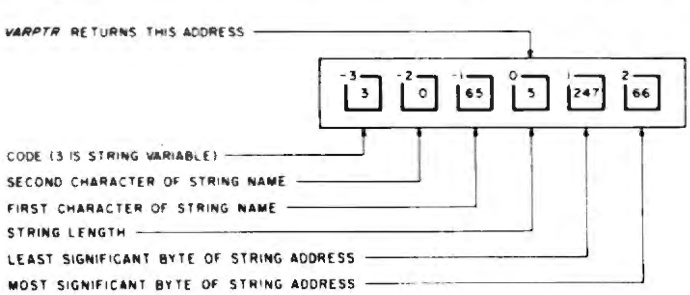



*Figure 1*

J = 1 TO 200:NEXT:NEXT

As the screen scrolls, whatever moves up to the first line becomes A$.

Such a string can be made up to 255 bytes. It can be put anywhere on the screen as long as the whole string is within the limits of video memory.

### Screen Strings for a Text Editor

To show the method's potential, consider a Basic text editor. Text editors or word processing programs, handy for written communication, are very expensive. The basis of their operation is represented in this very simplified routine:

```
10 REM
20 I$ = INKEY$: IF I$ = "" THEN 20
30 REM ' TESTS FOR CONTROL
   CHARACTERS HERE
40 TEXT$ = TEXT$ + I$
50 PRINT I$;: GOTO 10
```

The program works well until it unexpectedly stops. If you were not watching the screen when it happens, you may lose a few words.

Why? Line 40 works like the loop in the demonstration program using 20 bytes of memory to create a 5 byte string. A 60-character line uses 1890 bytes ($n^2/2 + 1.5 \times n$), leaving more than 1800 garbage bytes. When the reserved string space is full, the computer has to stop, sort out extra debris and rearrange current strings to make room for more before resuming its task.

By establishing a screen string for the video line where the input will be, we can delete line 40 and get TEXT$ when the line is finished. With this method we create only 60 extra bytes for a 60 character line—one for I$ each time a character is entered. You may wonder about all the times the program loops through line 20 when there is no input. The interpreter reestablishes the address of I$ each time, even though it does not change. Since its length is zero, no string storage is used.

The reduction from 1800 to 60 bytes is a big improvement, but we can go further. A character picked up by INKEY$ becomes I$ and is stored in the next free space of the string storage area. The computer keeps track of the next byte in addresses 16598 and 16599. With our routine, each time another character is entered the address stored in 16598 is decremented by one.

```
    D E M O N S T R A T I O N   P R O G R A M

5  CLS
10 A$="ABC"
20 T=24: PRINT "A$= " A$;
30 VP= VARPTR(A$): PRINT TAB(T)"ADDR. OF VAR. PTR. =" VP
40 PRINT "LENGTH OF A$=IS" PEEK(VP);
50 LSB=PEEK(VP+1): MSB=PEEK(VP+2)
60 PRINT TAB(T)"ADDR. OF STRING: LSB=" LSB ", MSB=" MSB
70 DEC= LSB + 256*MSB:PRINT TAB(T)"DECIMAL ADDR.OF STRING =" DEC
90 LIST 10-70


95 '
```

*Program Listing 1*

```
S A V E  &  R E S T O R E  T H E  C O N T E N T S  O F  A  S C R E E N

    Save The Screen

97 'S = MOST SIGNIFICANT BYTE OF SCREEN ADDR.
    A$= SCREEN STRING INCLUDES 3 SCREEN LINES + 63 BYTES OF 4TH LINE
    B$(0-3)= SAVES ALL BUT LAST BYTE OF EACH QUARTER OF THE SCREEN
    P(0-3)= LAST BYTE OF EACH QUARTER OF THE SCREEN

100 S=60: A$="": VP=VARPTR(A$): POKE VP,255: POKE VP+1,0: FOR I=0

    TO 3:
    POKE VP+2,S: B$(I)=A$: S=S+1: P(I)=PEEK(S*256-1): NEXT: RETURN

    Restore The Screen

107 'S = LAST BYTE OF EACH QUARTER OF THE SCREEN

110 CLS: S=15359: FOR I=0 TO 3: PRINT B$(I);: S=S+256: POKE S,P(I):
    IF I<3 PRINT: NEXT ELSE NEXT: RETURN


195 '
```

*Program Listing 2*

I$ is used to compare to control codes and take proper action, or to print it on the screen. The current I$ then becomes garbage. If only we could use the same location to store I$ for the whole line.

After the string location is established and the program is ready for a line of input, get the contents of the least significant byte of the string storage address with M = PEEK(16598). Then change line 10 in our simplified routine to 10 POKE 16598,M.

This will cause the same address to be used each time. Except for the rare instance where the first letter of a line is the last available byte of string storage, there is no pause during entry as there is no change in total string storage until the line is finished. This routine is also faster than the original, since no string concatenation takes place during input.

Another problem with Basic word processors crops up when they reach their specified line length in the middle of a word. The slight delay while shifting a word to the next line sometimes causes the loss of one or two letters. This is resolved by writing a short machine-language subroutine into the input routine so characters typed beyond the line length get a short beep from the amplifier. Input continues uninterrupted until a space is typed. The last word is shifted to the next line.

To make or revise your own word processor, use these hints on screen strings.

First, establish the screen string. POKE the desired line length plus twenty into the VARPTR to allow for overrun at the end of the line. Print the line number if desired. Then PEEK locations 16416 and 16417 for the current cursor location. Transfer these to the address of the screen string in the variable list table (in VARPTR + 1 and VARPTR + 2). The screen string will start at the cursor location. An input routine similar to the one in Program Listing 3 (lines 220–260) is used.

Be sure to have your program check the cursor location for each line. If it is greater than 16319, PRINT @ 1023, " " to scroll the screen. Then redetermine the cursor location. This prevents the screen from scrolling during input. Scrolling would move your input out of the screen string location.

To further reduce time-consuming string manipulation, exchange pointers instead of strings when moving lines.

By now, your mind is probably running away with ideas for including this method in some of your own programs. Here are a few short applications you may find useful.

**Saving a Screen Display**

Sometimes it is convenient to save the contents of a screen display for recall later in a program. In Listing 2, line 100 is a one line subroutine to save the screen. Line 110 is another one-liner to restore it.

**Subroutine for String Input**

You have a full well-formatted screen. You need an input from the keyboard. But when you press enter at the end of the input, it erases the next line on the screen. Or someone uses a comma or colon in their input; the computer only takes part of the input and prints ?Extra Ignored on the next line. The unemotional computer can ignore this, but it ruined your nice display. Listing 3 lists the essentials of a subroutine to prevent this problem. Now, in programs for things like data management requiring a lot of input, the program will not hang in the middle of a line.

**Saving Disk Directories in an Array**

Program Listing 4 gives a routine for putting disk directories into a string array for NEWDOS or other systems allowing DOS commands from Basic. It is handy for making an index of your programs.

**A Block of Screen Strings**

Program Listing 5 is a short routine for creating a block of screen strings. Its use can save a portion of the screen, compare current screen graphics with the originals, or output selected lines or line segments to a printer. ■


SPECTACULAR GAMES
For TRS 80
COLOR COMPUTER
$21.95
Each
Arcade Quality
Only 16K Required
Complete Machine Language
Color Graphics
Fast Pace
SPECTRAL ASSOCIATES
SPACE INVADERS
GHOST GOBBLER
Continuous Action
Highest Resolution
No Basic Needed
High Quality
Super Sound
Free Catalog
Satisfaction Guaranteed
141 Harvard
Tacoma, WA.
98466
SPECTRAL ASSOCIATES
119
Local
Information Call
(206)565-8483
If not available at your local dealer ask why not and
Call Toll Free 800-426-1830
Except WA., AK., HI., (8:00-4:00 Mon.-Fri.)
RAINBOW
CERTIFICATION
SEAL

```
        S U B R O U T I N E   F O R   S T R I N G   I N P U T

        ( Variables L & S to be supplied by the calling program )

 197 'L = MAX. LENGTH OF INPUT STRING
      S = SCREEN LOC. ( 0 TO 1023 )
      SC= SCREEN ADDRESS OF S
      SL= LEAST SIGNIFICANT BYTE OF SC
      SM= MOST SIGNIFICANT BYTE OF SC
      K = COUNT OF STRING LENGTH
 199 'M = ADDRESS STORING LSB OF NEXT AVAILABLE SCREEN STOR. ADDR.
      MC= CONTENTS OF M
      A$= SCREEN STRING
      B$=INPUT STRING RETURNED TO THE PROGRAM
 200 K=0: A$="": M=16598: V=VARPTR(A$): SC=S+15360: SM=INT(SC/256):
     SL=SC-SM*256: POKE V,L: POKE V+1,SL: POKEV+2,SM
 210 FOR I=S TO S+L-1: PRINT @ I, CHR$(14);: NEXT: PRINT@S,;: MC=PEEK(M)
 220 POKE M,MC
 230 I$=INKEY$: IFI$="" OR I$=CHR$(10) OR I$=CHR$(26) OR I$=CHR$(27)
     THEN 230 ELSE IF I$=CHR$(13) THEN B$=LEFT$(A$,K): RETURN
 240 IF I$=CHR$(8) OR I$=CHR$(24) THEN K=K-1 ELSE K=K+1
 250 IF K<L PRINT I$;: GOTO 220
 260 IF K=L PRINT CHR$(15) I$;: GOTO 220 ELSE 220
 295 '
```

*Program Listing 3*

```
     L O A D   D I R E C T O R I E S   I N T O   A   S T R I N G   A R R A Y
 297 'B$ ARRAY SAVES ENTRIES
      A$= SCREEN STRING
      E$= DISK I.D.
      I = ENTRY COUNTER
      SL= LEAST SIGN. BYTE OF SCREEN ADDR.
      SM= MOST SIGN. BYTE OF SCREEN ADDR.
      K = GETS LENGTH OF ENTRY
 300 CLEAR 4800: CLS: DIM B$(300): A$="": V=VARPTR(A$): POKE V,13: I=-1
 310 PRINT"(IF NO MORE DISKS TO BE LISTED, ENTER 'NONE'.)":
     INPUT"ENTER I.D. NO. OF DISK WHICH YOU HAVE INSERTED";E$
 320 IF E$="NONE" THEN 370 ELSE SL=108: SM=60: CMD "DIR"
 330 I=I+1: SL=SL+20: IF (SL+4)/64=INT((SL+4)/64) THEN SL=SL+4
 340 IF SL>255 THEN SL=SL-256: SM=SM+1
 350 POKE V+1,SL: POKE V+2,SM: K=INSTR(A$," "): IF K=1 THEN I=I-1: GOTO 310
 360 B$(I)=LEFT$(A$,K-1)+STRING$(15-K,32)+E$: GOTO 330
 370 REM   * B$ ARRAY NOW CONTAINS THE DIRECTORY ENTRIES *
 380 REM  * PUT SORT ROUTINE HERE *
 395 '
```

*Program Listing 4*

```
    C R E A T E   A   B L O C K   O F   S C R E E N   S T R I N G S

 397 'S = SCREEN LOCATION OF 1ST STRING (0 TO 1023 )
      L = LENGTH OF STRINGS
      N = NUMBER OF STRINGS

 400 DIM A$(N-1): SC=S+15296: SM=INT(SC/256): SL=SC-SM*256
 410 FOR I= 0 TO N-1: A$(I)="": V=VARPTR(A$(I)):
     SL=SL+64: IFSL>255 THEN SL=SL-256: SM=SM+1
 420 POKE V,L: POKE V+1,SL: POKE V+2,SM: NEXT: RETURN

 595 ' = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
       End of Routines for article
       Following are Tests to check out SCREEN STRING Routines


 700 ' = = = = TEST FOR SAVE - RESTORE SCREEN ROUTINE = = = =

 710 CLEAR1200: CLS: S=15423
   '  Fill The Screen
 720 FOR I= 65 TO 80: PRINT STRING$(63,I);: POKE S,I: S=S+64:
     IF I<80 PRINT: NEXT ELSE NEXT
 725 GOSUB 100: S=459: FOR I=1 TO 300: NEXT
 730 FOR I=0 TO 2: PRINT@ I*64+S,STRING$(40,32);: NEXT
 740 PRINT@ 529,"PRESS ENTER TO CLEAR SCREEN";:INPUTE
   ' Restore The Screen
 750 CLS: PRINT@ 529,"PRESS ENTER TO RESTORE SCREEN";:INPUTE
 760 GOSUB 110
 770 FOR I=1TO 400: NEXT: PRINT@384,;CHR$(200);:PRINT@384,;: END
 795 '= = = = = = TEST FOR STRING INPUT = = = = = = =
 800 CLEAR 500: CLS: P=512
 805 PRINT@P,CHR$(31);
 810 INPUT"INPUT SCREEN LOCATION (512 TO 1000)";S
 820 INPUT"MAX. LENGTH OF INPUT STRING";L
 822 IF S+L+7 >1023 PRINT"RUNS OFF BOTTOM OF SCREEN.": GOTO 810
 825 PRINT@P,CHR$(31);
 830 PRINT@S,"INPUT:  ";: S=S+7: GOSUB 200
 840 PRINT CHR$(15);: N=N+1: PRINT@ N*64,"B$="CHR$(34)B$;CHR$(34);: GOTO805
 850 END
 875 REM  TO TEST DISK DIRECTORY ROUTINE, ADD LINE 390  FOR J=0 TO I: ? B$(J
 ),: NEXT: END
 895 '= = = = T E S T   F O R   B L O C K   O F   S T R I N G S = = = =

 900 CLEAR1000:CLS:INPUT"LOCATION OF 1ST STRING (0 TO 1023)";S
 910 INPUT"LENGTH OF STRINGS";L
 920 INPUT"NUMBER OF STRINGS";N
 925 DIMB$(N):CLS:FORI=65TO78:FORJ=0TO63STEP4:PRINTUSING"##! ";J;CHR$(I);:NE
XT:NEXT
 927 INPUT"PRESS ENTER TO PRINT BLOCK OF STRINGS";Z
 930 GOSUB400:FORI=0TON-1:B$(I)=A$(I):NEXT
 940 CLS:FORI=0TON-1:PRINTB$(I):NEXT
```

*Program Listing 5*

*How they work for Fortran and Basic.*

# Boolean Logic Operators

*John Crew*
*1106 Karin Drive*
*Normal, IL 61761*

The Level II Basic reference manual mentions only briefly that numeric variables (integer, single precision and double precision) can be used like Logical Fortran variables and how If. . .Then statements work. I shall deal with the more understandable Fortran Logical variables before proceeding to their Level II Basic equivalent. References to Fortran apply specifically to WATFIV-S, although they are true for most versions.

A Logical variable can receive two different values (representing true and false) because of some logical operation. Figure 1 gives examples of Logical variable assignment statements in Fortran and their Level II Basic equivalents.

An assigned Logical variable can replace a logical expression (see Fig. 2). Program Listings 1 and 2 demonstrate Logical variables. In both sample programs the variable name Finish replaces a logical expression after the keyword If. If the variable Finish contains a value representing false, the program will skip the rest of the If...Then statement.

The sample Basic program is like a Fortran program, but with minor differences. Fortran programs must define any variables in the first non-comment program lines. Logical variables must have their own definition statements. Read and Print statements usually have an accompanying Format list. The peculiar default rule assigns a type to each undefined variable so that it is best to define all variables.

A Logical variable separates the evaluation of some logical expression from the If...Then statement or statements where it would normally appear. You might wish to set a flag for later, or to test the same condition in several places in your program. Evaluate the common condition in a subroutine and store the result in a Logical variable, saving typing and memory space.

Level II Basic returns a −1 for a true logical expression and 0 for a false one. In an If...Then statement, when the computer reaches the end of the logical ex-

**The Key Box**

**Basic Level II**
**Models I or III**

```
C   THIS PROGRAM READS STUDENT ID NUMBERS AND TEST SCORES
C   (WHICH RANGE 0 TO 100) AND PRINTS AN AVERAGE SCORE FOR
C   EACH STUDENT. A NEGATIVE ID NUMBER IS USED TO INDICATE
C   THE END OF THE DATA
C
      REAL INDAVG
      INTEGER IDNUM, SCORES(5), I, SUM
      LOGICAL FINISH
    5 READ 10, IDNUM, SCORES
   10 FORMAT (I6, 5(1X,I3))
      FINISH = (IDNUM .LT. 0.0)
      IF (FINISH) GO TO 40
      SUM = 0
      DO 20 I = 1, 5
        SUM = SUM + SCORES(I)
   20 CONTINUE
      INDAVG = FLOAT(SUM) / 5.0
      PRINT 30, IDNUM, SCORES, INDAVG
   30 FORMAT (' ','STUDENT NUMBER: ',I6,' FIVE TEST SCORES',
     +' ', 5(1X,I3),'   AVERAGE OF THESE TEST SCORES: ',
     + F3.2)
      GO TO 5
   40 STOP
      END
```

*Program Listing 1. Sample Fortran program.*

Legal assignment of a Value to a LOGICAL Variable

| LEVEL II BASIC | | FORTRAN |
|---|---|---|
| | equivalent to | |
| P# = −1 | | P = .TRUE. |
| Q! = 0 | | Q = .FALSE. |
| R% = (NOT P#) OR (P# AND Q!) | | R = (.NOT. P) .OR. (P .AND. Q) |
| | | P, Q, and R are LOGICAL variables |

*Fig. 1. Legal assignment of a value to a Logical variable.*

# BUY DIRECT

## AT DISCOUNT PRICES

It's simple . . . .

# TRS-80® COMPUTERS

**NEW!** Radio Shack TRS-80 Hard Drives Model I & III. . .from **$2087**

| | |
|---|---|
| TRS-80 Color Computer | TRS-80 Model II Computer |
|  BUY DIRECT |  BUY DIRECT |
| PRICED FROM **$289** | PRICED FROM **$2899** |
| TRS-80 Model III Computer | TRS-80 Model 16 Computer |
|  BUY DIRECT |  BUY DIRECT |
| PRICED FROM **$599** | PRICED FROM **$4098** |
| Smith Corona TP-1 Daisy Wheel Printer | **FRANKLIN** ACE 1000 ★ ATARI ★ commodore ★ CALL FOR PRICES |
|  BUY DIRECT | |
| PRICED FROM **$579** | |

ALL TRS-80'S ARE PURE RADIO SHACK EQUIPMENT

# 1-800-841-0860

Write for our Consumer Discount Price List

## Micro Management Systems, Inc.

PARCEL DIVISION · DEPT NO. 4
2803 THOMASVILLE ROAD EAST
CAIRO, GEORGIA 31728
912-377-7120

96

# BUYING DIRECT-MAIL QUESTIONS & ANSWERS

**Why Purchase A Computer By Direct-Mail?**
When buying a name-brand product - price is usually the main reason. The product is the same quality no matter where you buy.

**Do I Really Save Money - If I Need Help?**
When you buy by direct-mail the firm has geared its operation to sell at low prices. The savings will give you additional money to buy the extras you may need. Automakers don't teach you how to drive a car - we don't teach you how to use a computer. We do sell you a name-brand computer at low, low prices. Buy the computer and software you want, then you can learn to do-it-yourself and consult a professional only when additional help is required.

**What About The Warranty On My Equipment?**
If you purchase your computer equipment from a direct-mail firm - make sure the firm is an authorized dealer. Beware of firms selling modified or altered name-brand products as the manufacturer's warranty may no longer apply. Micro Management Systems sells only original factory-pure name-brand products covered by the manufacturer's warranty. Write us for a free copy of the warranty on any product we sell.

**Which Direct-Mail Company Should I Buy From?**
Micro Management Systems - Cairo, Georgia. We sell name-brand products. We offer low, low prices. We have a large inventory to assure prompt delivery. Our obligation to you . . . . .QUALITY PRODUCT - PRICE -HONESTY - RELIABLE.

**Who Is Micro Management Systems?**
Micro Management Systems has been selling computer equipment to the consumer on a direct-mail basis since 1978. Since that time thousands of consumers have purchased from us. We are HONEST and RELIABLE. If you plan to purchase by direct-mail you owe it to yourself to call us. We offer quality name-brand products, low prices, prompt delivery, toll free phones for quotes and orders, references, convenient ordering. All products we sell are factory-pure and backed by reliable name-brand manufacturers.

WE SELL AMERICA - Call us 1-800-841-0860
Write for our CONSUMER DISCOUNT PRICE LIST.

| TRS-80 COMPUTERS | |
|---|---|
| COLOR COMPUTER Priced From. . . . . . $289 | **FRANKLIN** ACE COMPUTER CALL FOR PRICE |
| MODEL III TRS-80 Priced From. . . . . . $599 | ATARI |
| MODEL II TRS-80 Priced From. . . . . $2899 | commodore |
| MODEL 16 TRS-80 Priced From. . . . . $4098 | SCM SMITH-CORONA |

# 1-800-841-0860

## Micro Management Systems, Inc.

PARCEL DIVISION · DEPT NO. 4 A
2803 THOMASVILLE ROAD EAST
CAIRO, GEORGIA 31728
912-377-7120

See List of Advertisers on Page 451

| Level II Basic | Fortran |
|---|---|
| equivalent to | |
| P# = −1 | P = .TRUE. |
| Q! = 0 | Q = .FALSE. |
| R% = (NOT P#) OR (P# AND Q!) | R = (.NOT. P) .OR. (P .AND. Q) |
| | P, Q, and R are LOGICAL variables |

| Level II Basic | Fortran |
|---|---|
| equivalent to | |
| IF A = A THEN... | IF A = A THEN... |
| IF −1 THEN... | IF (.TRUE.) THEN... |
| IF NOT 0 THEN... | IF (.NOT. .FALSE.) THEN... |
| Q% = −1: IF Q% THEN... | Q = .TRUE.<br>IF (Q) THEN... |
| Q% = 87: IF Q% THEN... | Q = .TRUE.<br>IF (Q) THEN... |

*Fig. 2. Tests which are always true.*

```
IF P# THEN...
IF Q! THEN...
IF R% THEN...
IF (R% OR Q!) AND NOT (R% AND Q!) PRINT "R XOR Q IS TRUE."
IF A%(3) THEN...
IF IN$ = "Y" and P! THEN...
P! = 21: Q! = 16: IF P! AND Q!, ...'ALWAYS TRUE
P! = 64: Q! = 128:IF P! AND Q! THEN... 'ALWAYS FALSE
IF A + 9 THEN...'FALSE ONLY IF A = -9
```

*Fig. 3. Additional examples of using Logical variables in Level II Basic.*

```
10 REM THIS PROGRAM READS STUDENT ID NUMBERS AND TEST
20 REM SCORES (WHICH RANGE FROM 0 TO 100) AND PRINTS AN
30 REM AVERAGE SCORE FOR EACH STUDENT. THE END OF THE
40 REM DATA IS INDICATED BY A NEGATIVE ID NUMBER.
50 CLEAR 0: DEFINT A-Z: DIM SCRE(4)
60 READ IDNUM!: FOR I = 0 TO 4: READ SCRE(I): NEXT
70 FINISH = (IDNUM!<0)
80 IF FINISH GOTO 160
90 SUM = 0
100 FOR I = 0 TO 4
110    SUM = SUM + SCRE(I)
120 NEXT
130 INDAVG! = CSNG(SUM) / 5.0
140 PRINT USING "STUDENT NUMBER: ###### FIVE TEST SCORES: "; IDNUM!;:
    FOR I = 0 TO 4: PRINT USING " ###"; SCRE(I);: NEXT: PRINT USING "
    AVERAGE OF THESE TEST SCORES: ###.##" ; INDAVG!
150 GOTO 60
160 END
170 DATA 123456, 97, 78, 84, 76, 89, 654321, 79, 80, 91, 83, 100, -1, 0, 0, 0, 0, 0
```

*Program Listing 2. Level II Basic version of Fortran sample program.*

pression (indicated by a "," a ":", the conventional "Then," or the beginning of another Basic statement), it compares the result of the evaluation to 0. If it is 0 (indicating a false expression) the computer scans for the keyword Else and, on locating it, executes any following statements. If Else is absent in that line the computer moves to the next program line.

All Level II needs is 0 stored in a register upon reaching the end of a logical expression after the keyword If. A number, numeric variable or complex arithmetic statement can replace a logical expression (see Figs. 2 and 3).

These techniques should work in other Microsoft versions of Basic, perhaps not with Pocket Computer Basic. ■

*The author is currently a junior at Illinois State University majoring in Physics.*

✓ See List of Advertisers on Page 451

*The screen stops when full.*

# Scroll Your Listings

*Mark C. Paxton*
*4056 Three Oaks Boulevard*
*Troy, MI 48098*

The assembly program presented here changes the way LIST functions on your nondisk Level II TRS-80: When you enter the List command, the screen fills with source lines of any Basic program resident in memory, and stops once the screen is full—it does not spin madly past you.

To continue viewing the program listing one line at a time, press the up arrow key for each new line desired. If you wish to scroll backward one line at a time, press the down arrow key.

### Using the Program

Key the source code in Program Listing 1, using an editor/assembler, and create a system tape named SCRS. Set the memory size to 32540 and load the system tape. When the program stops loading, control transfers to the ready mode, from the system mode.

At this point you can either CLOAD or key in a Basic program. When you execute the LIST command the screen fills with lines; listing will stop when the screen is full. The machine then waits for your input—however, it recognizes only the up or down arrows. To break out of this mode before listing has completed, repeatedly hit the up arrow key and the Break key at the same time until the break is recognized.

To switch between normal LIST mode and the modified LIST mode, key in the SCRS command, or the SCROFF command, in normal command mode as illustrated below:

```
>
>SCRS
>
>SCROFF
>
```

SCRS (SCRoll Screen) causes the LIST command to scroll one line at a time. SCROFF (SCRoll OFF) returns the LIST command to its normal operating mode.

### Analysis of the Program

The program takes advantage of three DOS exits: 41E2H, 41B2H, and 41DFH.

The DOS exit at 41E2H is called from the System command routine (at 02B2H). The SCRS program uses it to pass control back to the Basic ready mode, once it has completed loading.

The DOS exit at 41B2H is part of the ready routine which starts at 1A19H, 41B2H is called from 1A1CH. At this point, you have already made a keyboard entry which has been compressed to space compression codes wherever possible. Therefore, this DOS exit is used to slip out and check the input buffer for the entry of a new command: SCRS or SCROFF. If you do not enter either of these commands, control returns to Basic and no harm is done. If you enter either SCRS or SCROFF, then you execute the appropriate routine in the program to activate or deactivate the modified LIST mode.

The DOS exit at 41DFH is part of the LIST/LLIST command routine logic. LIST starts at 2B2EH, and LLIST starts at 2B29H. The two commands use the same code. However, between 2B29H and 2B2EH, the output device type indicator (located at 409CH) gets set to 01, which indicates the printer. (The default code is zero, for video. A minus 1 means the cassette.) The call to 41DFH is made from 2B44H; it is called once before each line is displayed.

If you have a line printer on your system, put a check in the SCRS program to see if the output device type indicator (at 409CH) contains an 01H. If it does, you can return immediately to Basic to avoid problems. It is not necessary that you do this however; the only adverse effect is that the screen clears every time you enter LLIST.

The program in Listing 1 is not very large considering the functions it performs. This is due largely to the use of numerous ROM routines to perform general tasks. A short study of the listing itself will prove beneficial and may add new ROM routines to your own repertoire. ■

*Mark Paxton, a professional programmer and systems analyst, also teaches Cobol part-time.*

***The Key Box***

**Basic Level II**
**Model I**
**16K RAM**
**Printer Optional**

*Program Listing*

```
41E2              00010           ORG     41E2H    ;SET UP AUTOSTOP OF
SYSTEM CMD
41E2 C3AF7F       00020           JP      AUTO
41B2              00030           ORG     41B2H    ;SET UP USER EXIT TO
 TEST BUFFER
```

*Listing continues*

*Listing continued*

```
41B2 C3C37F               JP      USEXIT
41DF                      ORG     41DFH   ;EXIT 'LIST' CMD
41DF C31C7F               JP      SCRS
                    ;*********************************
                    ;* PROGRAM=======> SCRS          *
                    ;* MAINT-DATE====>  02.12.81     *
                    ;* AUTHOR========>  M.C.PAXTON   *
                    ;* DESCRIPTION===>  SCROLL UP    *
                    ;* OR DOWN OF BASIC LISTINGS     *
                    ;*********************************
7F1C                      ORG     32540 ;SET MEMSIZE TO THIS
7F1C D5             SCRS  PUSH    DE ;LINE DELIMITER
7F1D C5                   PUSH    BC ;NEXT LINE PTR
7F1E E5                   PUSH    HL ;PTR TO LINE #
7F1F 2B                   DEC     HL
7F20 2B                   DEC     HL
7F21 EB                   EX      DE,HL ;DE=PTR TO BEGIN
                                     ;  OF LINE (PTR TO
                                     ;  NEXT LINE)
7F22 2AA440               LD      HL,(40A4H) ;BEG OF BASIC
7F25 B7                   OR      A ;CLR CARRY
7F26 ED52                 SBC     HL,DE
7F28 2003                 JR      NZ,CONT ;NOT FIRST LINE
7F2A CDC901               CALL    01C9H  ;CLS ON 1ST LINE
7F2D ED5B2040       CONT  LD      DE,(4020H)       ;DE=CURSOR P
OSITION
7F31 21003F               LD      HL,3F00H         ;HL=DELIMITE
R LINE ADDRSS
7F34 B7                   OR      A        ;CLEAR CARRY
7F35 ED52                 SBC     HL,DE    ;CHECK TO SEE IF AT
BOTTOM OF SCR
7F37 FA3E7F               JP      M,KBSCAN ;JUMP TO KBSCAN IF
AT BOTTOM
                    ;=====> SCROLL SCREEN FORWARD ROUTINE <=====
7F3A E1             UP    POP     HL       ;UP SCREEN ROUTINE ;
RESET REGS
                                           ;AND GOBACK FROM WHE
NCE WE CAME)
7F3B C1                   POP     BC
7F3C D1                   POP     DE
7F3D C9                   RET
7F3E CD2B00         KBSCAN CALL   2BH ;KEYBOARD SCAN ROUTINE -
 LOOK FOR
                                  ;UP ARROW ENTRY OR DOWN ARRO
W ENTRY.
                                  ;IGNORE EVERYTHING ELSE.....
...
7F41 B7                   OR      A
7F42 28FA                 JR      Z,KBSCAN
7F44 FE5B                 CP      5BH ;UP ARROW
7F46 28F2                 JR      Z,UP
7F48 FE0A                 CP      0AH ;DOWN ARROW
7F4A 2804                 JR      Z,DOWN
7F4C 18F0                 JR      KBSCAN
0002                TSTSCR DEFS   2
                    ;=====> BACKUP DISPLAY ROUTINE <=====
7F50 AF             DOWN  XOR     A
7F51 F5                   PUSH    AF
7F52 21C03F               LD      HL,3FC0H
7F55 224E7F               LD      (TSTSCR),HL ;SET  A=0. A IS
CTR FOR # OF
                                  ; LINES ON SCREEN TO BACKUP
OVER. SET
                                  ; TSTSCR VARIABLE TO LAST LI
NE OF SCREEN.
7F58 B7             AGN2  OR      A        ;CLEAR CARRY
7F59 114000               LD      DE,40H
7F5C 2A4E7F               LD      HL,(TSTSCR)      ;HL=LINE ADD
RESS
7F5F ED52                 SBC     HL,DE    ;CALC NEW SCRN LINE
ADDRSS
7F61 224E7F               LD      (TSTSCR),HL      ;SAVE FOR NE
XT TIME
7F64 CD5A1E               CALL    1E5AH   ;CONVRT TO HEX  (INT
O DE IT GOES)
7F67 7A                   LD      A,D
7F68 B3                   OR      E        ;NZ=LINE # THERE,Z=N
O LINE #
7F69 2005                 JR      NZ,CONT2
7F6B F1                   POP     AF
7F6C 3C                   INC     A        ;INC CTR FOR # LINES
 TO BACKUP
7F6D F5                   PUSH    AF       ;SAVE CTR (IN A)
7F6E 18E8                 JR      AGN2     ;DO IT AGAIN
7F70 21003C         CONT2 LD      HL,3C00H ;LOAD ADDRESS OF SC
R 1ST LINE
7F73 CD5A1E               CALL    1E5AH ;DE=HEX LINE # ON RET
7F76 CD2C1B               CALL    1B2CH ;BC=PTR TO NEXT LINE P
TR
                                  ; ON RETURN
7F79 50                   LD      D,B
7F7A 59                   LD      E,C      ;DE = BC
7F7B 2AA440         AGN   LD      HL,(40A4H) ;HL=BEGIN ADDRSS
OF BASIC TEXT
7F7E E5                   PUSH    HL
7F7F 44             TEST  LD      B,H
7F80 4D                   LD      C,L
7F81 B7                   OR      A
7F82 ED52                 SBC     HL,DE
7F84 E1                   POP     HL
7F85 280C                 JR      Z,DONE
7F87 C5                   PUSH    BC
7F88 DDE1                 POP     IX
7F8A DD6E00               LD      L,(IX)
```

*Listing continues*

*Listing continued*

```
7F8D DD6601                   LD      H,(IX+1)
7F90 C5                       PUSH    BC
7F91 18EC                     JR      TEST
7F93 F1             DONE      POP     AF ;RETRIEVE COUNTER IN A RE
G.
7F94 47                       LD      B,A
7F95 B7                       OR      A
7F96 78                       LD      A,B
7F97 2807                     JR      Z,AROUND
7F99 DDE5                     PUSH    IX        ;PUT ADDRESS OF PREV
 LINE PTR
7F9B D1                       POP     DE        ; INTO THE DE REGIST
ER PAIR
7F9C 3D                       DEC     A
7F9D F5                       PUSH    AF
7F9E 18DB                     JR      AGN
7FA0 4E             AROUND    LD      C,(HL) ;PUT EVERYTHING BACK
IN PLACE
7FA1 23                       INC     HL
7FA2 46                       LD      B,(HL)
7FA3 23                       INC     HL
7FA4 E5                       PUSH    HL
7FA5 C5                       PUSH    BC
7FA6 CDC901                   CALL    01C9H ;CLS
7FA9 C1                       POP     BC
7FAA E1                       POP     HL
7FAB D1                       POP     DE
7FAC D1                       POP     DE
7FAD D1                       POP     DE ;DE=DELIMITER LINE NO. FR
OM STCK
7FAE C9                       RET
7FAF 3EC9           AUTO      LD      A,0C9H
7FB1 32E241                   LD      (41E2H),A
7FB4 21191A                   LD      HL,1A19H
7FB7 E3                       EX      (SP),HL
7FB8 C9                       RET ;GO TO 1A19H <READY>
                    ;*****************************************
************
                    ;*USER EXIT: CHECK INPUT BUFFER FOR SCRS/SCR
OFF ENTRY *
                    ;*****************************************
************
7FB9 53             CMDLIT    DEFM    'SCRSSCROFF'
7FC3 E5             USEXIT    PUSH    HL ;SAVE BUFFER  POINTER
7FC4 08                       EX      AF,AF'  ;SAVE  REGISTERS
7FC5 D9                       EXX             ;SAVE  REGISTERS
7FC6 E1                       POP     HL      ;RETRIEVE PTR TO BUF
FER
7FC7 D7                       RST     10H     ;LOCATE 1ST BYTE OF
COMMAND ENTRY
7FC8 2B                       DEC     HL      ;POINT BEFORE COMMAN
D ENTRY
7FC9 E5                       PUSH    HL      ;SAVE POINTER
7FCA 11B87F                   LD      DE,CMDLIT-1 ;POINT TO LITERA
LS  -  1
7FCD 0604                     LD      B,4     ;LOAD 1ST COMMAND LE
NGTH
7FCF CDE27F                   CALL    CMDTST  ;IS IT <SCRS>?
7FD2 2817                     JR      Z,SCRON ;GO-IF YES
7FD4 E1                       POP     HL
7FD5 0606                     LD      B,6     ;LOAD 2ND COMMAND LE
NGTH
7FD7 11BC7F                   LD      DE,CMDLIT+3 ;POINT TO NEXT L
ITERAL
7FDA CDE27F                   CALL    CMDTST  ;IS  IT <SCROFF>?
7FDD 2813                     JR      Z,SCROFF;GO-IF YES
7FDF 08             GOBACK    EX      AF,AF'  ;REPLACE THE ORIG RE
GS
7FE0 D9                       EXX
7FE1 C9                       RET
7FE2 23             CMDTST    INC     HL
7FE3 13                       INC     DE
7FE4 1A                       LD      A,(DE)  ;GET BYTE FROM LITER
AL
7FE5 BE                       CP      (HL)    ;DOES IT MATCH THE C
ORRESPONDING
                                              ;BYTE THAT WAS ENTER
ED IN THE
                                              ;BUFFER AREA?
7FE6 2002                     JR      NZ,NOPE ;IF NOT,THEN GET OUT

7FE8 10F8                     DJNZ    CMDTST  ;MATCH-TEST NEXT BYT
E
7FEA C9             NOPE      RET     ;Z=FOUND, NZ=NOT FOUND
                    ;*******************
                    ;* SCRON ROUTINE   *
                    ;*******************
7FEB 3EC3           SCRON     LD      A,0C3H
7FED 32DF41                   LD      (41DFH),A ;ENABLE THE JUMP
7FF0 1805                     JR      READY   ;GOBACK TO <READY>
                    ;******************
                    ;* SCROFF ROUTINE *
                    ;******************
7FF2 3EC9           SCROFF    LD      A,0C9H
7FF4 32DF41                   LD      (41DFH),A ;DISABLE THE JUMP.
 RESTORES
                                              ;THE LIST COMMAND TO ITS ORG
INAL MODE
                                              ;OF OPERATION.
7FF7 21191A         READY     LD      HL,1A19H  ;<READY> RTN ADDRE
SS
7FFA E3                       EX      (SP),HL ;STICK IT ON THE STA
CK
7FFB 18E2                     JR      GOBACK  ;RETURN TO <READY> R
TN
1A19                          END     1A19H
00000 TOTAL ERRORS
```

**120 DAY WARRANTY**

*"NO YOU'RE NOT SEEING THINGS, THAT'S FOUR INTERNAL DRIVES IN A MODEL III, WITH UP TO 4 MEG. DISK STORAGE AVAILABLE NOW, ONLY FROM DISPLAYED VIDEO!!"*

# DISPLAYED VIDEO IS DRIVING DOWN PRICES ON DISKS!

DISPLAYED VIDEO is now offering TRS-80* MODEL III with TANDON disk drives, one of the most reliable disk drive systems on the market, for **INCREDIBLY** low prices.

MODEL III with 48K, dual 40 track double density disk drives, complete systems with TRSDOS and ONE BOX OF DISKETTES

*PLUG IT IN AND GO. . . . . .* **$1,749.00**

MODEL III with 48K, dual 40/40 track double density disk drives, complete systems with NEWDOS 80 and ONE BOX OF DISKETTES

*PLUG IT IN AND GO. . . . . .* **$2,149.00**

MODEL III with 48K, dual 80 track double density disk drives, complete systems with NEWDOS 80 and ONE BOX OF DISKETTES

*PLUG IT IN AND GO. . . . . .* **$2,149.00**

MODEL III with 48K, dual 80/80 track double density disk drives, complete systems with NEWDOS 80 and ONE BOX OF DISKETTES

*PLUG IT IN AND GO. . . . . .* **$2,449.00**

MODEL III with 48K, four 40 track double density internal disk drives, complete systems with TRSDOS and ONE BOX OF DISKETTES *PLUG IT IN AND GO....* **$2,399.00**

MODEL III with 48K, four internal drives of any configuration available to achieve up to 4 meg of disk storage .......... **CALL**

MODEL III Internal Drive Kit: Includes controller board, dual drive mounting bracket, dual power supply, all hardware cables and connectors (gold plated) .................. **$329.00**

MODEL 16 w/1 drive....**$4,450.00**.........w/2 drives.............**$4,850.00**

**IMMEDIATE DELIVERY**
**120 Days Parts and Labor Warranty**
**DEALER INQUIRIES INVITED**
*Free Shipping in the U.S. 48 Contiguous States*

**Visit Our Two RETAIL LOCATIONS:**
886 Ecorse Rd., Ypsilanti, MI 48197
(313) 426-5086/(313) 482-4424
111 Marshall St., Litchfield, MI 49252
(517) 542-3280

DISPLAYED VIDEO

**Radio Shack Authorized Dealership at:**
**111 Marshall Street, Litchfield, Michigan 49252** ***#P142***

To Order: Call (313) 426-5086 or (313) 482-4424 or (517) 542-3280 or write DISPLAYED VIDEO, 886 Ecorse Road, Ypsilanti, Michigan 48197

*TRS 80 is a trademark of the Tandy Corporation ✓62 Prices subject to change without notice

# DISPLAYED VIDEO IS DRIVING DOWN PRICES ON DISKS!

"PRICE" "CUT"

"PRICE" "CUT"

**DV'S COLOR COMPUTER FIRST DRIVE**

**$425.00**

**80 TRACK TEAC DISK DRIVE W/CASE & POWER SUPPLY**

**$275.00**

**40 TRACK TEAC DISK DRIVE W/CASE & POWER SUPPLY**

**$239.00**

**TEAC'S NEW 40 TRACK SLIMLINE W/CASE & POWER SUPPLY**

**$269.00**

**TWO DRIVE CABLE $23.99**

With Gold Plated Connectors

**NEWDOS 80 $115.00 MODEL I or III**

Bare Drives & Cases Available Upon Request

**DV'S 5 MEG. HARD DISK**

**$1949.00**

**ATHANA DISKETTS**

**$26.95**

100 Percent Certified W/Own Hub Rings

# EPSON PRICES

We're not going to make you call for prices.
Here they are in black and white!!

**MX 80 (W/GRAPHTRAX) PLUS $459.00**

**MX80FT (W/GRAPHTRAX) PLUS $545.00**

**MX82FT (W/GRAPHTRAX) PLUS $555.00**

**MX100 (W/GRAPHTRAX) PLUS $709.00**

**PRINTER CABLE $23.99**

**Stop In Our Two Retail Locations:**
886 Ecorse Rd., Ypsilanti, MI 48197
(313) 426-5086/(313) 482-4424
or
111 Marshall Street,
Litchfield, MI 49252
(517) 542-3280

We Carry A Full Line of OKIDATA Printers

DISPLAYED VIDEO

**IMMEDIATE DELIVERY**
**180 Days Parts and Labor Warranty**
**DEALER INQUIRIES INVITED**
*Free Shipping in the U.S.*
*48 Contiguous States*

**Radio Shack Authorized Dealership at:**
**111 Marshall Street, Litchfield, Michigan 49252** ***P142**

To Order: Call (313) 426-5086 or (313) 482-4424 or (517) 542-3280 or write DISPLAYED VIDEO, 886 Ecorse Road, Ypsilanti, Michigan 48197

*TRS-80 is a trademark of the Tandy Corporation

Prices subject to change without notice

# APL Primer—Part I

by Margaret M. Grothman

**What's the difference between monadic and dyadic functions? Stay tuned for a description of system commands and variable names.**

APL is such a concise and efficient computer language that Basic seems clumsy by comparison. It has built-in functions that let you do with a few keystrokes what would take many program lines in another language. Because of its large library of built-in functions, APL lends itself well to mathematical and statistical applications. Functions not built in can be defined by you and called by name. This, plus its ability to operate on arrays, gives APL its strength.

APL80, written by Phelps Gates and published by Ramware (Milford, NH 03055), is a large subset of the standard APL language, and is available in either 16K cassette or 32K disk versions for the Model I. All the examples here work with the 16K version.

Because the large number of functions exceeds the number of symbols on a normal keyboard, standard APL requires a keyboard with special symbols, such as Greek letters. Since the TRS-80 has no such keyboard, shifted letters represent many special APL symbols.

### APL as a Calculator

APL has no print or write statement. To print to the screen, type your message in single quotes.

```
Enter: 'THIS IS A MESSAGE.'
Result: THIS IS A MESSAGE.
```

Another example is:

```
Enter: '10 + 11 + 3'
Result: 10 + 11 + 3
```

If you omit the single quotes, APL behaves like a calculator. You are using the execution mode, which is like the command mode in Basic.

APL does not use the dash sign for subtraction. The dash functions as a prefix to indicate a negative number. The sign for subtraction is also a dash, but it is in a lower position and is produced by pressing the clear key.

```
Enter: 30_ 12
Result: 18
```

```
Enter 30 - 12
Result: SYNTAX ERROR: 30 - 12
```

The second example will not execute because APL cannot find an operator. The negation sign indicates only that 12 is negative.

In APL80, you set the symbol for multiplication by pressing the shift key followed by X. It appears on the screen as a small graphics block followed by the letter X. The asterisk, which is the multiplication operator in Basic, is the power function in APL80.

```
Enter: 10■X 3,
Result: 30
```

```
Enter: 10*3
Result: 1000 (10 cubed)
```

Division requires the percent sign. The slash, used for division in Basic, is used in APL for other functions.

```
Enter: 15%3
Result: 5
```

APL is accurate to six significant digits.

```
Enter: 3.333333■X 10
Result: 33.3333
```

```
Enter: 6.666666■X 10
Result: 66.6667
```

Only six digits are displayed in the answer, regardless of how many you enter. Also, in the second example the sixth digit is rounded up.

### Combined Expressions

APL has no hierarchy of operations. Expressions are executed in order from right to left. This is one of the hardest features of APL to adjust to because it differs from other programming languages, and from mathematical convention. One reason for the right-to-left rule is that it is easy to remember. APL has so many functions that remembering a hierarchy would be difficult.

```
Enter: 3■X2 + 12
Result: 42
```

The result is not the same as the

✓See List of Advertisers on Page 451

Basic or Pascal result, and does not follow accepted mathematical practice, in which multiplication is performed before the addition. Here, 2 is added to 12 first, and the result is multiplied by 3.

You can alter the order of execution by using parentheses. Expressions within parentheses are executed from right to left. The parentheses in the next example force execution of 3 times 2 before adding 12.

Enter: (3▪X 2) + 12
Result: 18

If you evaluate $8+20\times2+12$, mathematical convention requires that multiplication be performed before addition, making the correct answer 60. Presenting this task to a Basic interpreter or a Pascal compiler produces the same result. But if you enter APL line 8 + 20 X▪2 + 12, the answer is 288. The right-to-left rule causes execution in this order:

$2 + 12 = 14$
$20 \times 14 = 280$
$8 + 280 = 288$

The result of each operation becomes data for the next operation. If you want the expression evaluated conventionally, use parentheses to execute $20\times2$ before the two addition operations. The equation 8 + (20▪X 2) + 12 produces the mathematically correct answer.

Here are two more examples:

Expression: $\frac{7\times5}{5\times10}$

Enter: 7▪X5%5▪X10
Result: .7

Expression: $\frac{4(3)}{3/15+7}$

Enter: 4▪X 3 % (3 % 15) + 7
Result: 1.66667

The first example requires no parentheses because the normal right-to-left execution causes no errors. The second example, however, requires parentheses around 3%15 to prevent the 7 from being added to 15 before division takes place. When in doubt, use parentheses; extra ones do no harm.

### Monadic Functions

The terms monadic, dyadic, and niladic describe the number of operands or arguments a function requires. All the functions introduced so far are dyadic: they require one operand before the function symbol and one after. A monadic function has only one argument, following the symbol. Niladic functions have no operands. None of APL80's built-in functions are niladic.

The absolute-value function is an example of a monadic function. Its symbol is produced by typing shift, J. (In APL80, all symbols preceded by a shift are displayed on the screen with a small graphics block.)

Enter▪J – 100 (use the negation sign, not the minus),
Result: 100

A monadic function generally shares its symbol with a dyadic function. For example, the dyadic use of shift, J is the residue function, which returns the remainder when division is executed.

Enter: 3▪J 16
Result: 1 (3 goes into 16 with a remainder, or residue, of 1).

The residue function can be used to extract the fractional part of a positive number. (In APL80, the residue function doesn't work with negative arguments.)

Enter: 1▪J 8.75
Result: .75

Each dyadic arithmetic function has a corresponding monadic function that uses the same symbol. The division symbol % used as a monadic operator produces the reciprocal of a number. (The reciprocal is one divided by the number.)

Enter: %5
Result: .2

The dyadic multiplication function shares its symbol, shift X, with the monadic sign function. This function returns a 1 if the operand is positive, – 1 if negative, and 0 if it is zero.

Enter:▪X5
Result: 1

Enter:▪X – 5
Result: – 1

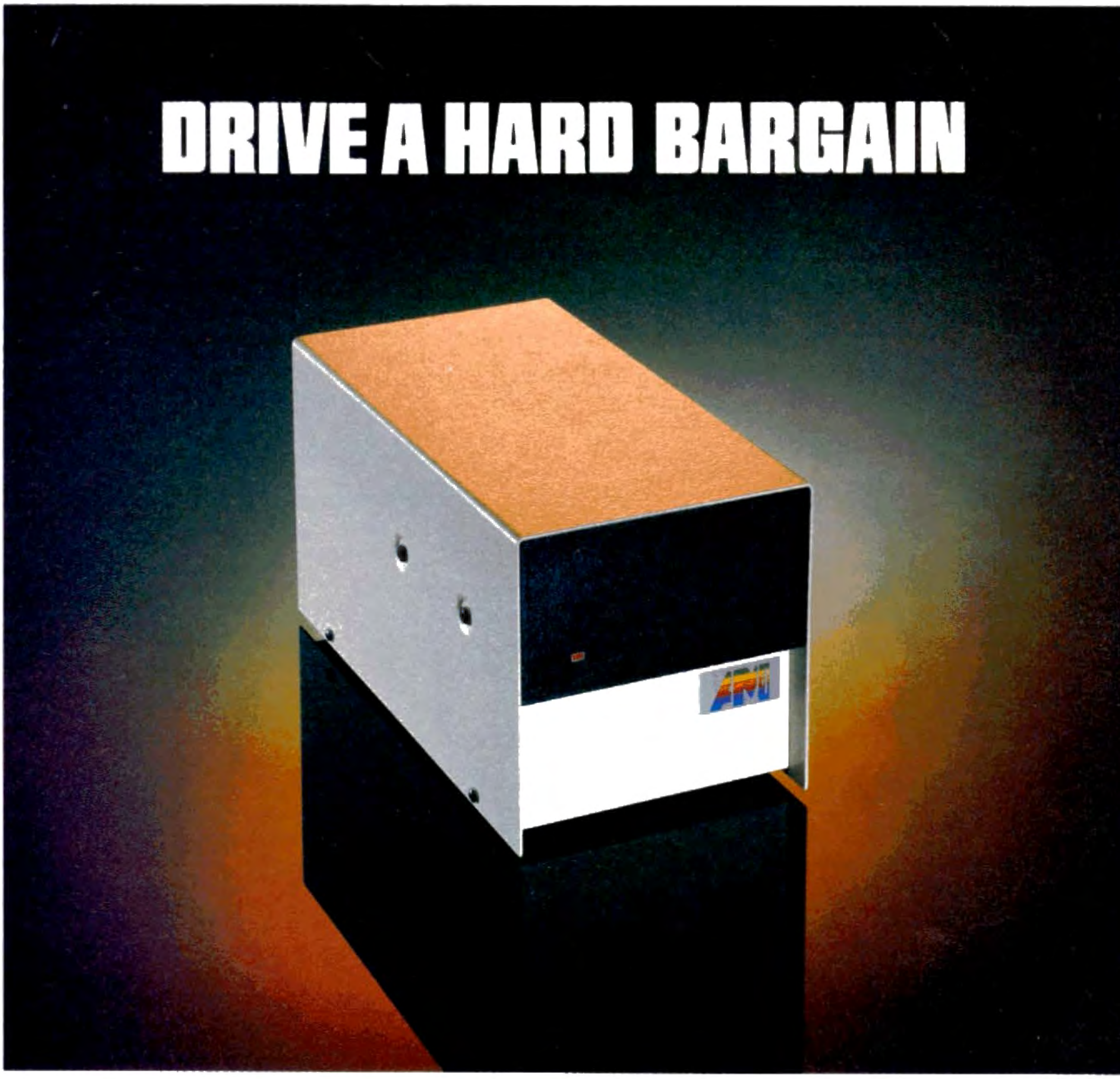
The minus sign (clear key) may be used as a monadic function to change the sign of a number.

Enter: _2
Result: – 2

Another symbol that does double duty as a monadic and a dyadic function is the exclamation point. Monadic! is the factorial function. Remember to put the symbol before the



Circle 38 on Reader Service card.→




**Save more money (and data) with our hard disk drive**
Introducing the new, low-priced Winchester hard disk drive from A.M. Electronics. Designed to put the power of hard disk data storage within reach of more system owners than ever before.

**More megabytes per buck**
Our drive features an unformatted data capacity of 6.7 megabytes, with an average access time of 75 milliseconds. Up to four drives can be daisy-chain connected for even greater capacity.

**Ready to run with TRS-80**
Each drive comes complete with controller board, ST-506 interface, and a DOS Plus 4.0i for complete plug compatibility with your TRS-80 system.

**Quality you can count on**
Manufactured from the finest components, the A.M. Electronics hard disk drive has an expected life rating of five years when used 50% of the time.

Power consumption is unusually low: typically one amp less than other Winchester systems.

The disk and read/write heads are fully enclosed in a module using an integral air recirculation system with an absolute filter. You can depend on stable operation and high reliability in any office environment.

**Think of the possibilities**
Word processing, large mailing lists, database management, financial planning and forecasting . . . demanding applications like these and many more are now possible—and affordable—with the new A.M. Electronics hard disk drive.

For more information, or to find the location of your nearest dealer, please contact: A.M. Electronics, Inc., 3446 Washtenaw Ave., Ann Arbor, MI 48104, 313/973-2075, Telex: 220821 HPC. In Europe, contact A.M. Electronics, Ltd., 10 Barley Mow, Barley Mow Passage, Chiswick, London W4 4PH, England, Tel. (01)994-6477, Telex: 8811418.

A.M. ELECTRONICS INC
THE POWER BEHIND THE DRIVES™

See Us At
COMDEX
Nov. 29-Dec. 2, 1982

™TRS-80 is a trademark of Tandy Corp.

argument, not after, as you would in conventional mathematical notation.

Enter: !4
Result: 24 (4 times 3 times 2 times 1)

The dyadic function that uses ! is called the combinatorial function. It is used to find the number of combinations of a certain size within a larger number. For example, 3!4 returns 4, the number of combinations of 3 that can be formed out of four objects.

The @ sign is also used for a monadic-dyadic function pair. The dyadic form is used to find logarithms to any base. The first argument is the base and the second is the number. To ask for log 16 to the base 4,

Enter: 4@16
Result: 2 (exponent to which 4 must be raised to obtain 16).

When @ is used as a monadic function, the natural log is returned. For example, to find the natural log of .7,

Enter: @.7
Result: −.356675

**Variable Assignments**

You may use long variable names in APL. APL80 will accept names as long as 60 characters. Since using the variable requires typing the entire name, short names are desirable.

Variable names can consist of any combination of letters and numbers, but must start with a letter. Names can contain dashes but not spaces. Assign a variable by typing shift followed by the left arrow.

Enter: NUMBER←15

The value assigned to a variable is displayed when you enter the name. If you type an unassigned name, an error message results.

Enter: NUMBER
Result: 15

Enter: NAME
Result: VALUE ERROR: NAME

Variables are used in expressions in the same way as constants. New variables may be defined in terms of already defined variables.

Enter: VALUE←NUMBER X 3
Enter: VALUE
Result: 45

Once a variable is assigned, it keeps its value until assigned another.

Enter: NUMBER←NUMBER + 1
Enter: NUMBER
Result: 16

The monadic version of the plus sign is used to transfer the value assigned to one variable to another. It is called the identity function. After the transfer the same value will be assigned to both variables.

Enter: V←(15*2) + 3 X 45
Enter: W← + V
Enter: W
Result: 360

Text material may also be assigned to variables. These are called literals. To assign a literal, type the name in single quotes.

Enter: NAME1←'FOZZY'
Enter: NAME2←'BEAR'

Arithmetic operations may not be performed on literal variables.

Enter: NAME3←NAME1 + NAME2
Result: DOMAIN ERROR: NAME3←NAME1 + NAME2

✓ See List of Advertisers on Page 451

There is a way, however, to combine literals.

```
Enter: NAME3←NAME1,NAME2
Enter: NAME3
Result: FOZZYBEAR
```

Or better yet,

```
Enter: NAME3←NAME1,' ',NAME2
Enter: NAME3
Result: FOZZY BEAR
```

The comma is used to chain variables together. This is the catenation function. It may be used for combining any variables, not just literals.

In the following example, variables are used to convert 10 degrees Celsius to Fahrenheit degrees. The first line assigns the value 10 to the variable name CELSIUS. The second line applies the conversion formula, $F = 32 + 9/5 \times C$, and assigns the result to the variable FAHRENHEIT. The last line retrieves the value of FAHRENHEIT computed in the second line.

```
Enter: CELSIUS←10
Enter: FAHRENHEIT←32 + (9%5)■X CELSIUS
Enter: FAHRENHEIT
Result: 50
```

This operation could be done in two lines by using the quad symbol in the second line. The quad symbol is usually represented by a little box called a window. In APL80, it is obtained by typing shift, Q. Replace the second and third lines in the above example with ■Q←32 + (9%5)■ X CELSIUS.

Quad used in this way causes the value assigned to the variable to be displayed right away, making it unnecessary to enter the variable name separately. You will be seeing the quad function later in another use.

Although this example illustrates the assignment and use of variables, it is an awkward way of doing the job. To apply the formula to a different value of CELSIUS would require retyping the lines. There is a better way.

**An APL80 Program**

A more efficient way of converting Celsius temperatures to Fahrenheit is to define a function that can be called by name. Until now, we have done everything in execution mode. To create your own functions in APL, you must be in the definition mode. The system command )DEF takes you to that mode.

To avoid typing long names when the function is used, I've shortened FAHRENHEIT and CELSIUS to FAHREN and CEL. You do not have to type the line numbers; they are done automatically in the definition mode. This function consists of only two lines: the title line and line 1. When line number 2 is presented, press break to exit from the definition mode.

```
)DEF FAHREN
1: 32 + (9%5)■ X CEL
2: break
```

To call the FAHREN function, assign a value to CEL, then type and enter the function name.

```
Enter: CEL←5
Enter: FAHREN
Result: 41
```

This is a niladic function, one with no arguments. Later you will write this function as a monadic function, in which the value of CEL need not be entered separately first.

Try using FAHREN once more. This time assign seven values to CEL, each separated by a space.

```
Enter: CEL← -15 -10 -5 0 5 10 15
Enter: FAHREN
Result: 5 14 23 32 41 50 59
```

You have assigned a series of values to CEL. The variable CEL is now a vector, similar to a one-dimensional array in Basic. No dimension statement is required; dimensioning is done automatically in APL. You need only write an assignment line with the desired values following the left arrow, separated by spaces. When FAHREN is called, the defined operations are performed on each member of the vector.

You have written an APL program. APL programming is just writing user-defined functions.

**Another APL80 Program**

A defined function need not be restricted to one formula as in FAHREN. The function CONE defined below computes the base area and the volume of a cone when its height and the radius of its base are assigned to HT and RAD. Again, short variable names are used to avoid the tedium of typing long names. To use CONE, assign values to RAD and HT, then type CONE.

```
)DEF CONE
1: 'THE AREA OF THE BASE IS'
2: 3.1416■ X RAD * 2
3: 'THE VOLUME OF THE CONE IS'
4: (3.1416 % 3)■ X (RAD*2)■ X HT
5: break
```

```
Enter: RAD←1 2 3
Enter: HT←1 2 3
Enter: CONE
Result: THE AREA OF THE BASE IS 3.1416
12.5664 28.2744
THE VOLUME OF THE CONE IS 1.0472
8.3776 28.2744
```

Like FAHREN, CONE requires that its variables be defined in advance. Also like FAHREN, CONE can be rewritten to avoid the separate assignment statements. Since CONE has two arguments, it will be rewritten in Part II of this series as a dyadic function.

It is possible to assign a single value to one variable and a vector to the other. In this way you can determine the effect of holding one variable constant while varying another.

```
Enter: RAD←2
Enter: HT←4 5 6
Enter: CONE
Result: THE AREA OF THE BASE IS 12.5664
THE VOLUME OF THE CONE IS 16.7552
20.944     25.1328
```

**Other System Commands**

The definitions of several APL80 commands mention the word "workspace." In APL, workspace refers to the area of memory in which assigned variables and defined functions are stored. Work in progress can be stored and retrieved by saving and loading the workspace.

All system commands are preceded by a right parenthesis.

- )CLEAR erases all the assigned variables and defined functions from memory. In APL jargon, you have cleared the workspace.
- )FNS lists the names of all the functions defined in the workspace.
- )VARS lists all variable names that have been assigned.
- )SAVE saves the workspace to tape.
- )LOAD loads a saved workspace from tape.
- )CHECK is used to verify a saved workspace.
- )ERASE removes a function or variable from the workspace. The function or variable name follows the word ERASE.
- )EDIT allows you to make changes or insertions in a named function. The edit function will be described next month.
- )OFF clears APL80 from memory and returns you to Level II Basic. ■

*Margaret Grothman can be reached at 5117 Denton Place, Madison, WI 53711.*

✓ *See List of Advertisers on Page 451*

*Speed up your Basic displays.*

# POKE Graphics

**The Key Box**

**Basic Level II**
**Model I or III**
**16K RAM**

*James S. Schaefer*
*33 Jackson Road*
*Berlin, NJ 08009*

POKE Graphics add high speed displays to a Basic program.

The first two memory locations of each line store the starting address of the next line. To find the start address of the next line use the formula $n1 + n2*256$. The next two memory locations store the present line number. To find the present line number use the formula $n3 + n4*256$. Control, text characters and space compression codes combine to store the rest of the lines. A zero ends each line and tells the computer to go to the next line or end the program.

To create POKE graphic string(s) first make up dummy string(s) to change into graphic strings(s). Begin the program with these strings because the computer must "see" the graphic string(s) before they are printed.

The POKE Graphics program starts at line 300, leaving enough room for any graphic display you put in your program. For example, start a 15 character dummy string in line 10 by writing 10 G1$ = "123456789012345" To change this into a graphic string find the memory locations and POKE the new graphic codes into the locations now storing 123456789012345.

You can use the POKE Graphics program when you have keyed in line 10 as above.

Line 310 PEEKs location 293 to see if the program is in a Model I or Model III. A 73 stored in this location indicates Model III; any other indicates Model I. You are then informed of this in line 312. Line 310 also sets AS (address start) equal to the memory address of the beginning of the Basic program by using the pointer at addresses 16548 and 16549. This makes the program compatible with both the Model I, the Model III and Disk Basic. Line 314 asks what

```
300 CLS:PRINTCHR$(23):PRINT@20,"POKE GRAPHICS";:FORAT=1TO1000:NE
XT:CLS:GOTO350
310 AM=PEEK(293):IFAM=73THENAS=17385:AM$="Model III"ELSEAS=17129
:AM$="Model I"
311 '

James Schaefer
33 Jackson Road
Berlin, New Jersey 08009
609 767-2751


312 PRINTCHR$(23):PRINT@0,"I see you are using a "+AM$;:FORAT=1T
O1000:NEXT:CLS
314 INPUT"What line No. is the string you want to change in";AL
316 A1=PEEK(AS):A2=PEEK(AS+1):A3=PEEK(AS+2):A4=PEEK(AS+3)
318 A9=A3+A4*256
320 IFA9=ALTHENCLS:PRINT"I have found line No. ";AL:GOTO324ELSEC
LS:PRINT"I am looking at line No. ";A9
322 IFA9<ALTHENAS=A1+A2*256:GOTO316ELSEIFA9>ALTHENPRINT"I can no
t find that line. Please check your input.":FORZZ=1TO1000:NEXT:C
LS:GOTO310
324 A0=AS:PRINT"It starts in memory location No. ";A0:PRINT"----
----------------------------------------------------------------"
328 PRINTPEEK(A0);:A0=A0+1:AJ=AJ+1:IFAJ<4THENGOTO328ELSEIFPEEK(A
0)<>0THENGOTO328
330 PRINT:PRINT"------------------------------------------------
-----------------":PRINT"This is the computer code for the line."
:PRINT"The first two numbers (n1+n2*256) = start address of next
 line."
332 PRINT"The next two numbers (n3+n4*256) = present line number
.":PRINT"The remaining numbers are the ASCII & Internal Codes fo
r the":PRINT"program line.":INPUT"Press 'ENTER' to continue.";ZZ
:CLS
334 PRINT"Press the up arrow to go to next higher memory locatio
n.":PRINT"Press the down arrow to go to next lower memory locati
on.":PRINT"Press 'C' to change the present memory location.":PRI
NT"If 'C' is pressed you type in the new Code No. & press 'ENTER
'"
336 PRINT"As you change each memory location you will be shown w
hat":PRINT"is in the next location.  Be careful not to change so
mething":PRINT"outside of the string parenthesis. Press 'BREAK'
to escape."
337 A0=A0-AJ
338 PRINT"Memory location",A0;" = ";PEEK(A0),"ASCII = ";
340 IFPEEK(A0)>=32ANDPEEK(A0)<=191THENPRINTCHR$(PEEK(A0))ELSEPRI
NT"Computer code"
342 GOSUB390:GOTO338
350 PRINT"For introduction press 'I' any other key to start.":GO
SUB390:IFAKEY$<>"I"THENGOTO310ELSECLS
352 PRINT"Using poke graphics can add fast graphics to a BASIC p
rogram.
First you make up dummy strings. Example >> GR$=";CHR$(34);"0123
456789";CHR$(34);"
Next you poke a graphics character code into each memory"
354 PRINT"location. Then when the string is printed in the progr
am
then graphics will appear on the screen. In this example if
we poke graphics character '191' into each location between
the parenthesis when GR$ is printed we will see ";
356 PRINTSTRING$(10,191);"
printed. >>Press any key for program";:GOSUB390:CLS:GOTO310
365 GOTO365
390 AKEY$=INKEY$:IFAKEY$=""THEN390
392 IFASC(AKEY$)=91THENA0=A0+1
394 IFASC(AKEY$)=10THENA0=A0-1
396 IFASC(AKEY$)=67ORASC(AKEY$)=99THEN398ELSERETURN
398 INPUT"New code";AN:POKEA0,AN:RETURN
```

*Program Listing*

| | | |
|---|---|---|
| Memory location | ##### = 71 | ASCII = G |
| Memory location | ##### = 49 | ASCII = 1 |
| Memory location | ##### = 36 | ASCII = $ |
| Memory location | ##### = 213 | ASCII = Internal Code |
| Memory location | ##### = 34 | ASCII = " |
| Memory location | ##### = 49 | ASCII = 1 |

*Table 1*

# COOSOL COMPUTER PRODUCTS

## EPSON PRINTERS

- EPSON MX70 ... $288
- EPSON MX80 F/T GRAFTRAX PLUS ... CALL
- EPSON MX100 GRAFTRAX PLUS ... CALL

**OTHER PRINTERS**

- NEC PC-8023A-C ... CALL
- NEC 3550 ... CALL
- NEC 3510, 3515, 3530 ... $1695
- NEC 7710, 7715, 7730 ... $2275
- NEC 7720, 7725 ... $2645
- C ITOH 8510AP CALL • 8510ACD ... CALL
- C ITOH 1541SP CALL • 1541SRE ... CALL
- C ITOH F10-40PU $1395 • F10-40RU ... $1395
- C ITOH F10-55PU $1795 • F10-55RU ... $1795
- OKIDATA 82A CALL • 83A ... CALL
- OKIDATA 84A ... CALL • 82A TRACTOR ... $55
- TALLY MT 1602 MT 1605 ... $1495
- TALLY MT 1802 MT 1805 ... $1795
- DIABLO 630 and 620 SERIES ... CALL
- QUME SPRINT 5 45/RO,KSR & 55/RO,KSR ... CALL
- ANADEX ADX-9500-P, ADX-9501-P ... $1295
- COMREX CR-1-S, CR-1-C, CR-1-Q ... CALL
- SMITH-CORONA TP-1 ... CALL

## HAYES SMART MODEM & CHRONOGRAPH



- HAYES SMARTMODEM (300 BAUD) ... $229
- HAYES SMART MODEM (300 & 1200 BAUD) ... $595
  - RS-232C COMPATIBLE
  - FULL and HALF-DUPLEX OPERATION WITH BOTH
  - TOUCH TONE and PULSE DIALOG
  - AUTO-ANSWER/DIAL/REPEAT
  - COMPLETELY PROGRAMMABLE
  - BUILT-IN AUTO MONITOR
  - STATUS AT A GLANCE
  - DIRECT-CONNECT DESIGN
- HAYES CHRONOGRAPH ... $209
- NOVATION AUTO-CAT AUTO ANS ... $219
- SIGNALMAN MARK 1 (300 BAUD) ... $89
- U.D.S. 103 OALP (300 BAUD) ... $175
- U.D.S. 103 JLP (300 BAUD) ... $199
- U.D.S. 202 SLP (1200 BAUD) ... $249

**TERMINALS**

- TELEVIDEO 910 $595, • 912 ... $705
- TELEVIDEO 920 $750, • 950 ... $959
- ADDS Viewpoint A1, A2, A3 ... $535

## SANYO COMPUTER



- SANYO MBC-1000 ... $1599
- SANYO MBC-2000 ... $2699
- SANYO MBC-3000 ... $4999
- SANYO ACCESSORIES & SOFTWARE ... CALL

**OTHER COMPUTERS**

- ALTOS Computer Systems ... CALL
- ADDS Computer Systems ... CALL
- NEC PC-8001A, PC-8012A, PC-8031A, and JB-1201 (NEC SYSTEM) ... $2149
- NEC Other CPU's and Software ... CALL
- DYNABYTE 5605-6-1 ... $5999
- DYNABYTE OTHER SYSTEMS ... CALL
- XEROX 820 SYSTEM w/5 1/4" DRIVES ... $2499
- XEROX 820 SYSTEM w/8" DRIVES ... $2995
- SOFTWARE FOR THE ABOVE ... CALL

COOSOL SPECIAL DISCOUNTS

APPLE SOFTWARE ... CALL
IBM SOFTWARE ... CALL

✓105

PHONE 7 DAYS A WEEK — Calif. (714) 545-2216 — (800) 854-8498

**COOSOL, INC. P.O. BOX 743, ANAHEIM, CALIFORNIA 92805-0743**

# TRS-80™ "CAN YOU BUY DIRECT?"

Bu... s buying from
the... ser this is not
po... to the source
the... r can come to
"a... SELL LOWER.
It (...

ON...
fro...
the...
BU...

**EXTRA! EXTRA!**

$30.00
**CHRISTMAS BONUS**

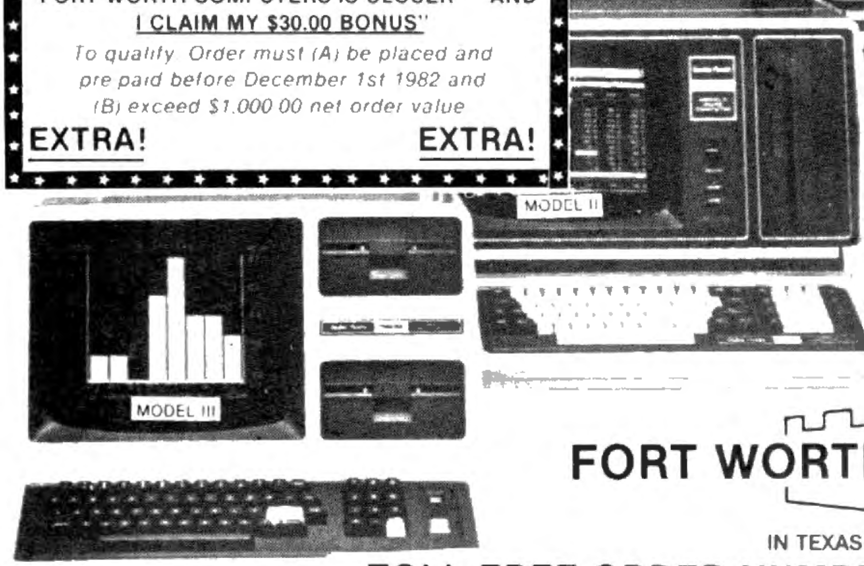
*The reader of this magazine will receive a bonus of $30.00 OFF our already-low-low prices "IF THEY CORRECTLY QUOTE" to us.*

"FORT WORTH COMPUTERS IS CLOSER — AND I CLAIM MY $30.00 BONUS"

*To qualify. Order must (A) be placed and pre paid before December 1st 1982 and (B) exceed $1,000.00 net order value*

**EXTRA! EXTRA!**







SAVE SALES TAX* PLUS DISCOUNT

*TEXAS RESIDENTS ADD ONLY 4%

"WE ARE CLOSER"

**FORT WORTH COMPUTERS**

✓214

IN TEXAS CALL 817-573-4111

**TOLL FREE ORDER NUMBER: 1-800-433-S-A-V-E**

TM: TRADE MARK OF THE TANDY CORPORATION

377 PLAZA GRANBURY • FORT WORTH • TEXAS 76048

line and AL (associate line) holds the line number.

A1 holds the least significant byte and A2 holds the most significant byte for the memory location of the next line. (Remember n1 + n2*256.) A3 holds the least significant byte and A4 holds the most significant byte for the present line number. (Remember n3 + n4*256.)

Line 318 makes A9 equal to the present line number (A3 + A$*256). If A9 equals AL then you have the line number you need and A0 = AS. A0 is the memory location for the line you need. Line 328 prints the ASCII and control codes to check for the dummy string. AJ records the number of memory locations the line uses so that after the code is printed you can return to the beginning of the line in memory. If the line found does not equal the line you need A9 is set to equal the start of the next line in memory. This continues until the line is found or there are no lines left.

You can now step through memory locations searching for the 123456789012345 you want to change. The up arrow steps up in memory; the down arrow steps down in memory. Press the up arrow until you see the contents of Table 1.

Now you can change the 1 stored here into a graphics code. Press C and answer the question New code ? with a graphics control code between 128 and 191. After you change this memory location you will see the new code. If you change the 1 into graphic code 191 the screen will show:

```
Memory location ##### = 191
                ASCII = ■
```

If you wanted this, press the up arrow to move to the next higher memory location. If not, press C to change the code stored in the memory location. Remember that if you change one outside of the quotation marks you could destroy your program. Be careful! ■

# Easy Picture Editor

**by Ron Ginger**

The TRS-80 Color Computer provides some great graphics commands that make graphics programs easy for the novice programmer—easy, but tedious. All graphics require x and y coordinates and some require several arguments. Keeping the arguments straight and getting the right coordinates takes careful work and much patience. Drawing the picture on graph paper helps keep things organized, but it's hard to see exactly what the screen will look like until you type in the entire program.

**Simple commands for art or game graphics: lines, boxes, and circles for the Color Computer.**

I use a GIGI color graphics terminal connected to a VAX 11/780 at work and have access to some powerful graphics editor software that lets me generate picture files by simple menu-driven keypad commands. I discovered that the same capability was possible for the Color Computer.

The program lets you create a drawing with simple keyboard commands and then save a tape file, in Basic program format, that can be loaded and run to draw your picture. You can use this program to create a display or as the basis of a game program.

Operation is simple. The text screen displays all possible commands and arguments. A blinking cursor appears in the first display column which you move up and down with the arrow keys. When the cursor is in front of the item you want to draw, press enter. If the command requires more arguments, the cursor will jump to the appropriate place and await your entry.

Once you have selected an object to draw, the graphics screen displays a single dot. Each arrow key moves this marker one pixel in the corresponding direction. With shift and an arrow, the marker moves 10 pixels. When the marker is where you want it, press enter. For lines, place the marker at the beginning of the line, press enter, move the marker to the end of the line, and press enter again. A line will be drawn between the selected points. You can continue to select and enter points as needed.

Boxes are entered in a similar manner. You move the marker to one box corner, press enter, move it to an opposite corner, and press enter again. For circles, first select the center, then a point of the circumference. Pressing an incorrect key returns the text screen for another command input.

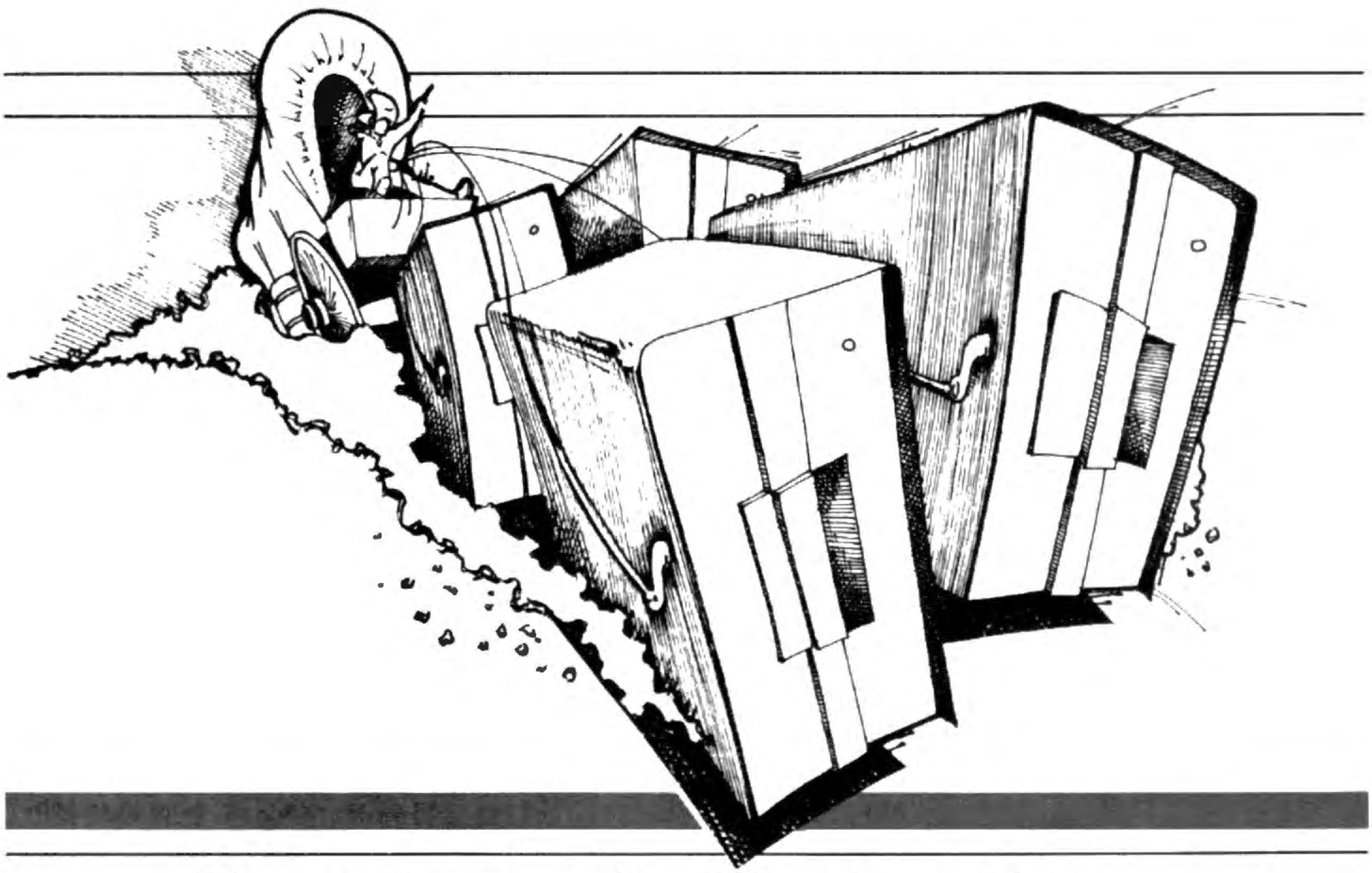
*Program Listing*

```
1 REM ************************
2 REM *   PICTURE EDITOR      *
3 REM *  BY RON GINGER       *
4 REM * VER 1   JAN 9 1982   *
5 REM ************************
10 CLEAR 1000:PMODE 3,1:PCLS:DIM L$(50)
15 D$=CHR$(94)+CHR$(9)+CHR$(10)+CHR$(8)+CHR$(95)+CHR$(93)+CHR$(9
1)+CHR$(21)
20 CS$(0)="GRN YEL BLU RED":CS$(1)="BUF CYN MAG ORA"
25 CLS:BL$=STRING$(32," "):FC=4:CM=1:CS=0:BC=1
30 C$(1)=" GREEN  ":C$(2)=" YELLOW ":C$(3)=" BLUE   ":C$(4)=" RE
D      "
35 C$(5)=" BUFF   ":C$(6)=" CYAN   ":C$(7)=" MAGENTA":C$(8)=" OR
ANGE"
40 GOTO 2000
45 PRINT@CP,CHR$(143);:C$=INKEY$:IF C$="" THEN PRINT@CP,CHR$(175
);:GOTO 45
50 RETURN
55 C=INSTR(1,"12345678",C$)
60 IF C=0 THEN RETURN
65 IF CS=1 AND C<4 THEN C=C+4
70 IF CS=0 AND C>4 THEN C=C-4
75 RETURN
100 REM **********COMMAND INPUT*****
105 C$=INKEY$:IF C$="" THEN 105
110 SCREEN 0,1
115 CP=CM*32:GOSUB 45
120 IF ASC(C$)=94 THEN CM=CM-1
125 IF ASC(C$)=10 THEN CM=CM+1
130 IF CM<1 THEN CM=13
```

*Listing continues*

**The Key Box**

**Color Computer**
**16K RAM**
**Extended Color Basic**



# MACHINE-LANGUAGE DISK I/O & OTHER MYSTERIES, TO HARNESS THE POWER OF DISK DRIVES.

## The best and most complete book on TRS-80 Model I and Model III disk I/O available.

"Machine-Language Disk I/O & other mysteries" by Michael Wagner. Volume 5 of the popular ". . . and other mysteries" series published by IJG.

More than 190 pages of information and discoveries on the subject of disk I/O for the beginner and expert alike, using Z-80 Assembly Language to directly control the TRS-80 Model I and Model III disk drives and interrupt systems. With this book anyone can become a disk I/O "expert", and make more efficient use of disk space, or to write his or her own "full blown" disk operating system!

## For the beginner and old pro'.

Machine-Language Disk I/O & Other Mysteries is packed with source code and flow chart illustrations for every function and operation described. And the source code for two complete disk I/O driver routines, one for the Model I and one for the Model III, are also included for "instant" access to your floppy drive systems.

Plus, the source code for a small disk operation system (S/OS) is included as a bonus. This S/OS contains all of the routines described in the book and shows how all of the disk I/O and interrupt functions should be incorporated into a working system.

## On machine language disk I/O.

Machine-Language Disk I/O & Other Mysteries explains what the floppy drive system is all about, what Tracks and Sector are, the Western Digital Floppy Disk Controller 1771 and 1793 chips, read/write access, reading a selected drives' status, how to test and make sure a disk is in a drive or if a drive is on the system (on-line), the Head Restore command, the Head Step and Seek commands, the "Force-Interrupt of function" command, the Read Address command, the Read Track command, the Write Track (FORMAT) command, the Read Sector command, the Write Sector command, Post Non-Maskable-Interrupt Processing for the model III, and much more.


## On TRSDOS file I/O

and most TRSDOS-like operating systems, Machine-Language Disk I/O & Other Mysteries explains; what a disk file really is, what records are and how they're stored on the disk, creating a file, performing Direct Record I/O, performing Single Byte I/O, closing & killing files; error processing and TRSDOS error codes with their meanings, and much more.

Plus many other handy programs, routines, and revelations are included. Such as a TRSDOS error message displayer, a disk formatter program, a program to calculate a password for a given encode, a object file load-format displayer, a file oriented full screen file editor, and a FULL BLOWN smart terminal program.

Machine-Language Disk I/O & Other Mysteries is a must for all programmers at just $29.95. Available at computer stores, selected B. Dalton Booksellers and independent book dealers. If your dealer is out of stock, order direct. Include $4.00 for shipping and handling. Foreign residents add $11.00 plus purchase price, in U.S. funds.

IJG
1953 West
11th Street
Upland, CA
91786 (714)
946-5805

When you are ready to save your masterpiece, simply select the save line and enter the file name you want to give the program. You are asked for a beginning line number for the Basic program. If the image is simply a drawing, 10 is a good place to start. If you plan to add some statements to the program to create a game, select a line number that puts the graphics in the correct location for your program.

**How It Works**

The program begins by initializing several needed variables and then, in lines 2000–2075, setting up the screen with the command list. It then jumps to the command-input routine at line 100. Lines 120 and 125 test input for the up and down-arrow keys (ASCII 94 and ASCII 10) and adjust CM to indicate the command selected. If you press enter, line 150 branches to an ON CM GOTO to dispatch to the section to handle the command.

Lines 200–295 handle the input of x and y coordinates. I tried to use joysticks for this, but found them somewhat clumsy and inaccurate. Note the use of INSTR in line 215 to select one of eight possible entries. This technique is faster and neater than the usual section of If...Then lines sometimes used.

The string D$ is set up in line 15, using CHR$ for each possible value of arrow and shift arrow keys. In 215 the entered variable, C$, is tested with INSTR and, if any arrow was pressed, returns a value between one and eight. If any other character is pressed, C will be set to zero and the On GOTO will fall through to the next line.

The program data is stored in an array L$, with LP pointing to the next element to add to the array. Note PIC-ED dimensions L$ to 50. If you intend to create a very long program, you might need to change the DIM and the Clear for string space. The commands are added to L$ without a line number. The tape-writing routine (lines 900–1015) adds a beginning line with PMODE 3,1 PCLS and Screen command and develops a line number based on the input value for starting line. A final line is added with a GOTO to maintain the display in the finished program.

I've found the program both fun to work with and useful as an easy way to generate graphics for other programs. It brings out the artist in everyone. ■

*Ron Ginger (17 Potter Road, Framingham, MA 01701) owns a Color Computer and a PDP-11. His interests are computing and boat building.*

*Listing continued*

```
135 IF CM>13 THEN CM=1
140 IF ASC(C$)=13 THEN 150
145 GOTO 115
150 ON CM GOTO 300,340,340,600,710,700,400,500,500,800,1100,1200
,900
200 REM **********COORDINATE INPUT**
205 PC=PPOINT(X,Y)
210 C$=INKEY$:IF C$="" THEN 210
215 C=INSTR(1,D$,C$)
220 PSET(X,Y,PC)
225 ON C GOTO 235,240,245,250,255,260,265,270
230 C=ASC(C$):RETURN
235 Y=Y-1:GOTO 275
240 X=X+1:GOTO 275
245 Y=Y+1:GOTO 275
250 X=X-1:GOTO 275
255 Y=Y-10:GOTO 275
260 X=X+10:GOTO 275
265 Y=Y+10:GOTO 275
270 X=X-10:GOTO 275
275 IF X>255 THEN X=255
280 IF X<0 THEN X=0
285 IF Y>191 THEN Y=191
290 IF Y<0 THEN Y=0
295 PC=PPOINT(X,Y):PSET(X,Y,(PC AND 7)+1):GOTO 210
300 REM***********COLOR CHANGES*****
305 CP=43:GOSUB 45
310 IF C$="1" THEN CS=1:GOTO 325
315 IF C$="0" THEN CS=0:GOTO 325
320 SOUND 10,5:GOTO 305
325 PRINT @43,CS;CS$(CS);
330 L$(LP)="SCREEN 1,"+C$:LP=LP+1
335 GOTO 110
340 REM B/F COLOR
345 CP=82:GOSUB 45:GOSUB 55
350 IF C=0 THEN SOUND 10,5:GOTO 345
355 PRINT@82,C;C$(C);:BC=C
360 CP=114:GOSUB 45:GOSUB 55
365 IF C=0 THEN SOUND 10,5:GOTO 360
370 PRINT @114,C;C$(C);:FC=C
375 COLOR FC,BC
380 L$(LP)="COLOR"+STR$(FC)+","+STR$(BC):LP=LP+1
385 GOTO 110
400 REM **********CIRCLES**********
405 CP=242:GOSUB 45:GOSUB 55
410 IF C=0 THEN 100
415 CC=C:PRINT@242,CC;C$(CC);
420 SCREEN 1,CS
425 GOSUB 200:IF C<>13 THEN 110
430 XO=X:YO=Y
435 GOSUB 200:IF C<>13 THEN 110
440 R=INT(SQR((XO-X)^2+(YO-Y)^2))
445 IF AR=1 THEN 525
450 CIRCLE(XO,YO),R,CC
455 L$(LP)="CIRCLE("+STR$(XO)+","+STR$(YO)+"),"+STR$(R)+","+STR$
(CC)
460 LP=LP+1
465 IF AR=1 THEN 525
470 GOTO 100
500 REM **********ARC,ELLIPSES******
505 CP=271:GOSUB 540:ST$=N$:ST=N
510 CP=281:GOSUB 540:SP$=N$:SP=N
515 CP=304:GOSUB 540:HW$=N$:HW=N
520 AR=1:GOTO 420
525 L$(LP)="CIRCLE("+STR$(XO)+","+STR$(YO)+"),"+STR$(R)+","+STR$
(CC)+","+HW$+","+ST$+","+SP$
530 CIRCLE(XO,YO),R,CC,HW,ST,SP
535 AR=0:LP=LP+1:GOTO 100
540 REM **********NUMBER INPUT******
545 PRINT @CP,"    ";:N$=""
550 GOSUB 45
555 IF ASC(C$)=13 THEN 575
560 IF ASC(C$)<46 OR ASC(C$)>57 THEN SOUND 10,5:GOTO 550
565 N$=N$+C$:IF LEN(N$)=4 THEN 575
570 PRINT @CP,C$;:CP=CP+1:GOTO 550
575 N=VAL(N$):RETURN
600 REM *********LINES************
605 SCREEN 1,CS
610 GOSUB 200:IF C<>13 THEN 110
615 XO=X:YO=Y:PSET(X,Y,FC)
620 GOSUB 200:IF C<>13 THEN 110
625 LINE(XO,YO)-(X,Y),PSET
630 GOSUB 660
635 L$(LP)=L$(LP)+"PSET":LP=LP+1
640 GOSUB 200:IF C<>13 THEN 110
645 LINE -(X,Y),PSET
```

*Listing continues*

*Listing continued*

```
650 GOSUB 665
655 L$(LP)=L$(LP)+"PSET":LP=LP+1:GOTO 640
660 L$(LP)="LINE("+STR$(XO)+","+STR$(YO)+")-("+STR$(X)+","+STR$(
Y)+"),":RETURN
665 L$(LP)="LINE-("+STR$(X)+","+STR$(Y)+"),":RETURN
700 REM **********BOXES*************
705 F=1
710 SCREEN 1,CS
715 GOSUB 200:IF C<>13 THEN 115
720 XO=X:YO=Y:PSET(X,Y,FC)
725 GOSUB 200:IF C<>13 THEN 115
730 IF F THEN LINE(XO,YO)-(X,Y),PSET,BF
735 IF F=0 THENLINE(XO,YO)-(X,Y),PSET,B
740 L$(LP)="LINE("+STR$(XO)+","+STR$(YO)+")-("+STR$(X)+","+STR$(
Y)+"),PSET,B"
745 IF F THEN L$(LP)=L$(LP)+"F"
750 LP=LP+1
755 F=0:GOTO 100
800 REM ***********PAINT************
805 CP=333:GOSUB 45:GOSUB 55
810 IF C=0 THEN 110
815 PP=C:PRINT @332,PP;C$(PP);
820 CP=350:GOSUB 45:GOSUB 55
825 IF C=0 THEN 110
830 BB=C:PRINT @350,BB;
835 SCREEN 1,CS
840 GOSUB 200:IF C<>13 THEN 110
845 PAINT(X,Y),PP,BB
850 L$(LP)="PAINT("+STR$(XO)+","+STR$(YO)+"),"+STR$(PP)+","+STR$
(BB)
855 LP=LP+1
860 GOTO 100
900 REM ***********SAVE TO TAPE*****
905 CP=432:GOSUB 1240
910 CP=443
915 GOSUB 45
920 IF ASC(C$)=13 THEN 945
925 PRINT @CP,C$;:LN$=LN$+C$
930 IF LEN(LN$)=4 THEN LN=VAL(LN$):GOTO 945
935 CP=CP+1:GOTO 915
940 PRINT@450,"LOAD TAPE-PRESS RECORD";
945 C$=INKEY$:IF C$="" THEN 945
950 PRINT @500,"...WORKING";
955 OPEN "O",-1,N$
960 X$=STR$(LN)+" PMODE 3,1:PCLS:SCREEN 1,1":PRINT #-1,X$
965 LN=LN+10
970 FOR I=0 TO LP
975 X$=STR$(LN+I*10)+" "+L$(I)
980 PRINT #-1,X$
985 NEXT I
990 LN=LN+(I+1)*10
995 X$=STR$(LN)+" GOTO"+STR$(LN)
1000 PRINT #-1,X$
1005 CLOSE
1010 PRINT@448,BL$;:PRINT@478,BL$;
1015 GOTO 110
1100 REM *********LOOK AT GRAPHICS***
1105 SCREEN 1,CS
1110 GOTO 100
1200 REM *********SKIP A FILE********
1205 CP=400:GOSUB 1240
1210 PRINT @450,"LOAD A TAPE-PRESS PLAY";
1215 C$=INKEY$:IF C$="" THEN 1215
1220 PRINT @500,"...WORKING";
1225 SKIPF N$
1230 PRINT @450,BL$;:PRINT @478,BL$;
1235 GOTO110
1240 REM *********FILENAME INPUT*****
1245 N$=""
1250 GOSUB 45
1255 IF ASC(C$)=13 THEN RETURN
1260 PRINT @CP,C$;
1265 N$=N$+C$:IF LEN(N$)=8 THEN RETURN
1270 CP=CP+1:GOTO 1250
2000 REM *********SET UP TEXT SCREEN**
2005 CLS:PRINT@8,"PICTURE EDITOR";
2010 PRINT@33,"COLOR SET";CS;CS$(CS);
2015 PRINT@65,"BACKGROUND COLOR  1 GREEN";
2020 PRINT@97,"FOREGROUND COLOR  4 RED";
2025 PRINT@129,"LINES";
2030 PRINT@161,"OPEN BOX";
2035 PRINT@193,"FILLED BOX";
2040 PRINT@225,"CIRCLE     COLOR";
2045 PRINT@257,"ARC     START      STOP";
2050 PRINT@289,"ELLIPSE   H/W";
2055 PRINT@321,"PAINT COLOR            BORDER";
2060 PRINT@353,"LOOK AT DRAWING";
2065 PRINT@385,"SKIP FILE NAME";
2070 PRINT@417,"SAVE FILE NAME           LN";
2075 GOTO 110
```

# 80 micro BOOKSHELF

**HOBBY COMPUTERS ARE HERE!** If you want to come up to speed on how computers work—hardware and software—this is an excellent book. It starts with fundamentals and explains the circuits and the basics of programming, along with a couple of TVT construction projects, ASCII, Baudot, etc. This book has the highest recommendations as a teaching aid. $2.97.* BK7322

**THE NEW HOBBY COMPUTERS**—This book takes up from where Hobby Computers Are Here! leaves off, with chapters on Large-Scale Integration, how to choose a microprocessor chip, an introduction to programming, low-cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! BK7340 Only $2.97.*

**BOTH VOLUMES FOR $4.95!**
**ORDER HC002**

**AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. BK1130 $12.50.*

**VOL. I—NEW 3rd Edition!** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume 1 is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume 1 builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. BK1030 $15.95

## NEW NEW NEW

**INTRODUCTION TO WORD PROCESSING** by Hal Glatzer. This book explains in plain language what a word processor can do, how to use one, how it improves productivity—especially in businesses that handle lots of words—and how to buy one wisely. No technical knowledge required, for all first-time users and those considering purchasing a word processor. BK1238 $12.95

**A USER GUIDE TO THE UNIX SYSTEM** by Jean Yates and Rebecca Thomas. Here at last is a clearly written book that allows you to use the Unix operating system easily, and at a fraction of the time it previously took. If you're using, evaluating or simply curious about this system, this is your book. BK1242 $15.99

**WORDSTAR MADE EASY** by Walter A. Ettlin. Now WordStar is as simple to learn as it is easy to use. This book teaches WordStar in 14 easy lessons, saving hours of hard work. It comes with a convenient pull-out Command Card. BK1239 $11.95



**KILOBAUD KLASSROOM**—by George Young and Peter Stark. Learning electronics theory without practice isn't easy. And it's no fun to build an electronics project that you can't use. *Kilobaud Klassroom*, the popular series first published in *Kilobaud Microcomputing*, combines theory with practice. This is a *practical* course in digital electronics. It starts out with very simple electronics projects, and by the end of the course you'll construct your own working microcomputer!

Authors Young and Stark are experienced teachers, and their approach is simple and direct. Whether you're learning at home or in the classroom, this book provides you with a solid background in electronics—and you'll own a computer that you built yourself! BK7386 $14.95

**TOOLS & TECHNIQUES FOR ELECTRONICS**—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects, as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.* BK7348

**• UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved. $10.95*

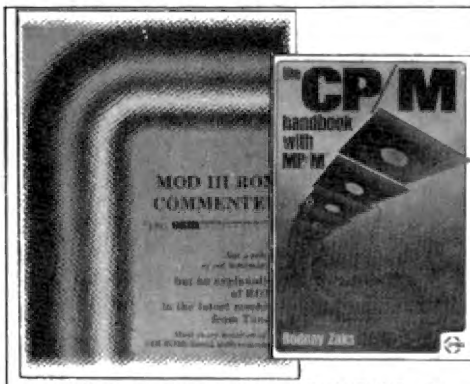


**• SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING**—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

**DON'T (or How to Care for Your Computer)**—by Rodnay Zaks. In plain language, with numerous illustrations, this book tells all the do's and don't's of the care, preservation and correct operation of the small computer system. Specific chapters cover each piece of hardware and software, as well as safety and security precautions and help for problem situations. Have your computer work right the first time and keep it working. No technical background required. For all computer users. BK1237 $11.95

**YOUR FIRST COMPUTER**—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) BK1191 $8.95*

**MICROPROCESSOR INTERFACING TECHNIQUES**—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. BK1037 $17.95.*




## Special Interests

**DOS RANDOM ACCESS & BASIC FILE HANDLING**—By H.J. Muller. This book was written for the nonprogrammer. It is ideal for the businessman or professional who needs to solve and write special programs for in-house business problems, or the hobbyist who wants to go beyond the cassette recorder and into disk storage and file manipulation. It is written as a self-instruction tutorial and will provide anyone with some Level II experience with the ability to write special programs for inventories, mailing list, work scheduling, record keeping, research project data manipulation, etc. 150 pages. BK1236 $29.50

**TRS-80 DISK AND OTHER MYSTERIES**—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to use," information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. BK1181 $22.50.*

**MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*. Contains more than 6500 lines of comments for the disassembled Level II ROMs and six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. BK1186 $29.50.*

**MOD III ROM COMMENTED**—Soft-Sector marketing, 1981. This book is not an instruction course on machine language, but rather an information source that you can use time and time again for writing your own program or patching old Mod I machine language programs. It contains an explanation of ROMs in the latest machine from Tandy, with most every location of the 14K ROMs listed, with comments. BK1235 $22.50

**THE CUSTOM TRS-80 AND OTHER MYSTERIES**—by Dennis Kitsz. More than 300 pages of TRS-80 customizing information. With this book you'll be able to explore your computer like never before. Want to turn an 8 track into a mass storage unit? Individual reverse characters? Replace the BASIC ROMs? Make Music? High speed, reverse video, Level I and Level II? Fix it if it breaks down? All this and much, much more. Even if you have never used a soldering iron or read a circuit diagram, this book will teach you how! This is the definitive guide to customizing your 80! BK1218 $29.95.*

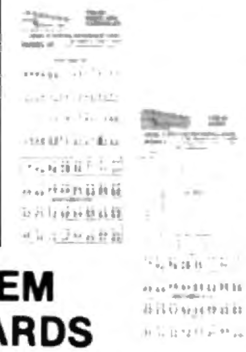
**BASIC FASTER AND BETTER AND OTHER MYSTERIES**—by Lewis Rosenfelder. You don't have to learn assembly language to make your programs run fast. With the dozens of programming tricks and techniques in this book you can sort at high speed, swap screens in the twinkling of an eye, write INKEY routines that people think are in assembly language and add your own commands to BASIC. Find out how to write elegant code that makes your BASIC really hum, and explore the power of USR calls. BK1221 $29.95.*

**THE CP/M HANDBOOK (with MP/M)**—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. BK1187 $14.95.*

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address. Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Micro* Book Department•Peterborough NH 03458. Be sure to include check or detailed credit card information.

**For Toll Free Ordering Call 1-800-258-5473**

# 80 micro BOOK SHELF

## NANOS SYSTEM REFERENCE CARDS

### For Models I, II, III and color computer

At last! No more flipping through the pages of the Basic manual! No more working through the maze of machine language instructions! These cards completely summarize the Basic and Assembler manuals! FEATURES INCLUDE: memory map, eyeball graphics, math instructions, basic commands, store instructions, basic functions, load instructions, basic statements, move instructions, special keys, exchange instructions, print using examples, shift instructions, basic special characters, compare instructions, basic and assembler messages and codes, branch instructions, basic facts, data alteration instructions, reserved words, I/O instructions, ROM routines, complete character chart with graphics and space-compression codes, hex-dec chart, control code cross-reference, assembler instructions commands and operators, screen line layout, editor commands & subcommands, condition code easy access

Designed as a fold-up, accordion-style card, fits in your pocket. Panels organized for optimum speed for reference.

| | |
|---|---|
| **Model I:** Basic only FC1001 | $2.95 |
| **Model I:** BASIC and Assembler FC1002 | $4.95 |
| **Model II:** BASIC and Assembler FC1005 | $5.95 |
| **Model III:** BASIC only FC1004 | $3.95 |
| **Model III:** BASIC and Assembler FC1003 | $5.95 |
| **Color:** BASIC and Extended FC1006 | $4.95 |
| **Apple II:** BASIC only FC1007 | $3.95 |
| **Apple II:** BASIC & 6502 FC1008 | $4.95 |

## Z80 6089

**INSIDE LEVEL II**—For machine language programmers. This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. BK1183 $15.95.*

**• Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z-80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z-80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

**TRS-80 ASSEMBLY LANGUAGE**—by Hubert S. Howe, Jr. This book incorporates into a single volume all the pertinent facts and information you need to know to program and enjoy the TRS-80. Included are clear presentations of all introductory concepts, completely tested practical programs and subroutines, details of ROM and RAM and disk operating systems, plus comprehensive tables, charts and appendices. Suitable for the first time user or more experienced users. BK1217 $9.95.*

**PROGRAMMING THE Z-80**—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. BK1122 $15.95.*

**6809 MICROCOMPUTER PROGRAMMING AND INTERFACING**—by Andrew C. Staugaard, Jr. Getting involved with Tandy's new Color Computer? If so, this new book from the Blacksburg Group will allow you to exploit the awesome power of the machine's 6809 microprocessor. Detailed information on processor architecture, addressing modes, register operation, data movement, arithmetic logic operations, I/O and interfacing is provided, as well as a review section at the end of each chapter. Four appendices are included covering the 6809 instruction set, specification sheets of the 6809 family of processors, other 6800 series equipment and the 6809/6821 Peripheral Interface Adapter. This book is a must for the serious Color Computer owner. BK1215 $13.95.*

***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments, and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*


# WAYNE GREEN BOOKS

**SELECTRIC INTERFACE**—by George Young. You need the quality print that a daisy wheel printer provides but the thought of buying one makes your wallet wilt. SELECTRIC™ INTERFACE, a step-by-step guide to interfacing an IBM Selectric I/O Writer to your microcomputer, will give you that quality at a fraction of the price. George Young, co-author of *Kilobaud Microcomputing* magazine's popular "Kilobaud Klassroom" Series, offers a low cost alternative to buying a daisy wheel printer. SELECTRIC INTERFACE includes: step-by-step instructions, tips on purchasing a used Selectric, information on various Selectric models, including the 2740, 2980, and Dura 1041, driver software for Z80, 8080, and 6502 chips, tips on interfacing techniques. With SELECTRIC INTERFACE and some background in electronics, you can have a high quality, low-cost letter quality printer. Petals not included. BK7388 (125 pages) $12.97.

**TEXTEDIT—A Complete Word Processing System in Kit Form**
*by Irwin Rappaport*
TEXTEDIT is an inexpensive word processor that can be adapted to suit your differing needs—from form letters to lengthy texts. Written in TRS-80 Disk BASIC, the system consists of several modules, permitting the loading and use of only those portions needed. A disk is also available which provides the direct loading of the modules; however, the book is required for documentation. For Model I and III with TRSDOS CONVERT, one disk drive (2 disk drives or copy utility need ed to transfer to system disk). Runs under TRSDOS 2.2/2.3. May not function under other systems. BK7387 (book, 90 pages) $9.97 DS7387 (disk) $19.97.

**THE NEW WEATHER SATELLITE HANDBOOK**—by Dr. Ralph E. Taggart WB8DQT. Here is the completely updated and revised edition of the best selling *Weather Satellite Handbook*—containing all the information on the most sophisticated and effective spacecraft now in orbit. Dr. Taggart has written this book to serve both the experienced amateur satellite enthusiast and the newcomer. This book is an introduction to satellite watching, that tells you how to construct a complete and highly effective ground station. Not just ideas, but solid hardware designs and all the instructions necessary to operate the equipment are included. An entire chapter is devoted to microcomputers and the Weather Satellite Station, and for the thousands of experimenters who are operating stations. *The New Weather Satellite Handbook* details all the procedures necessary to follow the current spacecraft. Weather Satellite contains Operation Satellite Systems, Antenna Systems, Weather Satellite Receivers, A Cathode Ray Tube (CRT) Monitor for Satellite Picture Display, A Direct-Printing Fascimile System for Weather Satellite Display, How to Find the Satellite, Test Equipment, Microcomputers and the Weather Satellite Station, Station Operations. **BK7383 $8.95.***

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address. Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Micro* Book Department•Peterborough NH 03458. Be sure to include check or detailed credit card information.

## For Toll Free Ordering Call 1-800-258-5473

# 80 micro BOOK SHELF

## Basic & Pascal


**THE BASIC HANDBOOK**—SECOND EDITION—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. BK1174 $19.95.*

**LEARNING TRS-80 BASIC**—by David A. Lien. Dr. Lien, who is the author of THE BASIC HANDBOOK and the original Radio Shack LEVEL I USER'S MANUAL, has compiled a tutorial which includes portions of the original USER'S MANUAL, and most of LEARNING LEVEL II along with extensive additions. It will completely cover the TRS-80 Models I, II, III, and 16 (sorry, not the color or pocket computers). It is, of course, written in the easy learning style which readers of Dr. Lien's books have come to enjoy. BK1175 $19.95.

**BASIC BASIC (2ND EDITION)**—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. BK 1026 $11.50.*

**INTRODUCTION TO TRS-80 LEVEL II BASIC AND COMPUTER PROGRAMMING**—by Michael P. Zabinski. Written by an experienced educator, this is the book for those beginners who want to learn about computers without having to become an expert. It has practical programs, useful line-by-line comments, excellent flowcharts accompanied by line numbers and over 200 exercises which help the reader assess progress, reinforce comprehension, and provide valuable practical experience. BK1219 $10.95.*

**50 BASIC EXERCISES**—by J. P. Lamoitier. This book is structured around the idea that the best way to learn a language is through actual practice. It contains 50 completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. Program subjects include mathematics, business, games, and operations research, and are presented in varying levels of difficulty. This format enables anyone to learn BASIC rapidly, checking their progress at each step. BK1192 $12.95.*

### NEW REVISED EDITION

**PROGRAMMING IN PASCAL**—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language and therefore suitable for an introductory course. BK1140 $12.95.*

**ADVANCED BASIC**—Applications including strings and files, coordinate geometry, area, sequences and series, simulation and graphing and games. BK1000 $11.50.

**INTRODUCTION TO PASCAL**—By Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. BK1189 $15.95.

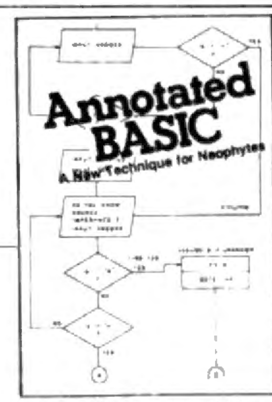
## Moneymaking

**HOW TO MAKE MONEY WITH COMPUTERS**—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. BK1003 $15.00.*

**HOW TO SELL ANYTHING TO ANYBODY**—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. BK7306 $2.25.*

**THE INCREDIBLE SECRET MONEY MACHINE**—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. BK1178 $5.95.*

## NEW NEW NEW




**ANNOTATED BASIC—A NEW TECHNIQUE FOR NEOPHYTES.**—BASIC programming was supposed to be simple—a beginner's programming language which was so near to English that is could be easily understood. But in recent years BASIC has become much more powerful and therefore much more difficult to read and understand. BASIC simply isn't basic anymore.

*Annotated BASIC* explains the complexities of modern BASIC. It includes complete TRS-80* Level II BASIC programs that you can use. Each program is annotated to explain in step-by-step fashion the workings of the program. Programs are flowcharted to assist you in following the operational sequence. And—each chapter includes a description of the new concepts which have been introduced.

*Annotated BASIC* deals with the hows and whys of TRS-80 BASIC programming. *How* is a program put together? *Why* is it written that way? By observing the programs and following the annotation you can develop new techniques to use in your own programs—or modify commercial programs for your specific use.
Annotated BASIC Volume 1 BK7384 $10.95
Annotated BASIC Volume 2 BK7385 $10.95
Order Both Volumes and Save! BK738402 $18.95

## Games

**CAPTAIN 80 BOOK OF BASIC ADVENTURES**—by Robert Liddil. This book contains 18 of the most popular Adventure programs available today; all in one easy to read book with listings ready to be keyed in to your computer. This unique book also contains an ADVENTURE GENERATOR program, not available from any other source. This generator program will actually write another BASIC ADVENTURE PROGRAM! Although specifically written for the TRS-80 Model I & III these programs are adaptable to other computers using Microsoft BASIC. BK 1240 $19.95.

**40 COMPUTER GAMES FROM *KILOBAUD MICROCOMPUTING***—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. BK7381 $7.95.*

**BASIC COMPUTER GAMES**—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. BK1074 $7.50.*

**MORE BASIC COMPUTER GAMES**—Edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. BK1182 $7.50.*

**WHAT TO DO AFTER YOU HIT RETURN**—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions and many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. BK1071 $16.50.*

## Business

**BUSINESS SYSTEM BUYER'S GUIDE**—By Adam Osborne with Steven Cook. When you enter the market place of small business computers, you face a bewildering array of products, prices, features and fables. This guide cuts through the jargon and unravels the task of buying the right computer system. This book provides solid information on how to determine your needs, how to choose software and hardware for all business applications, what to expect from vendors, what to avoid, and what questions you must ask. It also provides a wealth of detailed information on products, manufacturers, retailers and the whole microcomputer market. BK1229 $9.95

**SO YOU ARE THINKING ABOUT A SMALL BUSINESS COMPUTER**—by Richard G. Canning and Nancy C. Leeper. For a well-organized manual on the process of selecting the right computer system for your small business, this text can't be excelled. Designed to introduce the novice in data and word processing to the real benefits of computerization, the book is filled with money- and time-saving tips, photos of equipment, lists of suppliers, prices, explanations of computer terminology, and helpful references to additional sources of information. Everyone contemplating a first computer installation should have this book. BK1222 $14.00*

**SOME COMMON BASIC PROGRAMS**—Published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. BK1053 $14.99 paperback.

**THEORY Z—How American Business Can Meet the Japanese Challenge**—by William Ouchi. Why are the Japanese catching up and surpassing American industrial productivity? What allows Japanese industrialists to offer guaranteed lifetime employment to their workforce? This book will help you understand the Theory Z managerial philosophy and its implications for the American corporate future. Examples are given of the American industrial giants already operating under Z-style management, and the impact of this style on the quality of their executives and workers is explored. A must for the alert businessman, large or small. BK1226 $12.95*

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address. Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Micro* Book Department•Peterborough NH 03458. Be sure to include check or detailed credit card information.

## For Toll Free Ordering Call 1-800-258-5473

# School micros: boon or bane?

## *Panel to block conspicuous computing in schools*

By Eric Maloney
80 Micro Managing Editor

"Scam," "religion," and "fad" is how members of the Committee on Basic Skills Education describe the increased use of microcomputers in the classroom. The committee charges microcomputer manufacturers are making misleading claims to hustle their machines. It also says educators are throwing away money that could be used for better teaching tools and touting computers to cover up their failure to effectively teach basic skills.

"There is some sort of urgent requirement that every child must become computer-literate, which in the parlance of manufacturers means that he must learn how to interface with a computer and program a computer," said cofounder Eric Burtis. "This is absolutely untrue. The future of the U.S. is not based on everybody becoming computer-literate but on becoming literate in basic skills."

The committee was formed in April by Burtis and three other San Francisco-area residents. Burtis is the president of Centurion Industries, a manufacturer of dedicated microprocessor-based learning machines.

Manufacturers are pushing schools to "get a computer first and then get the kid to learn the other stuff," Burtis said. He called it "tremendous overkill," and said schools often end up using their computers for drill and practice.

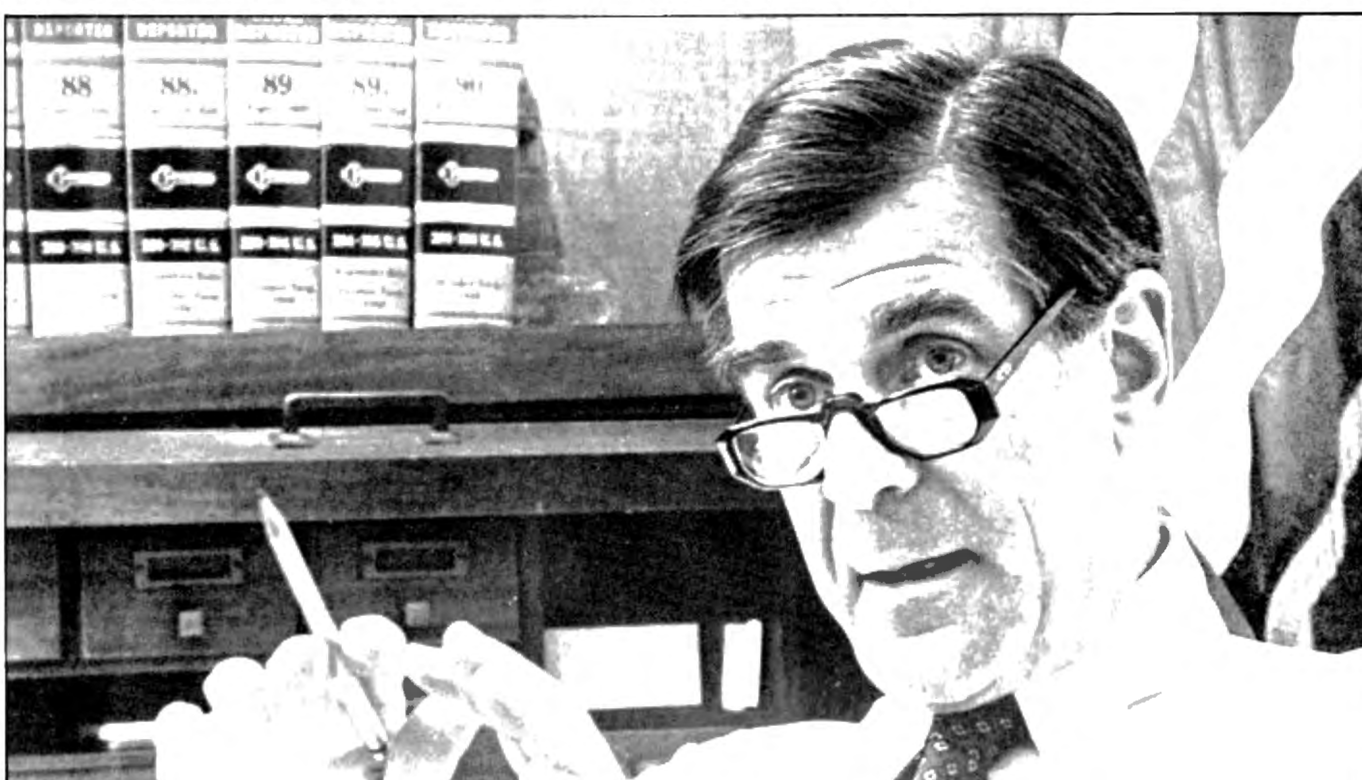
"Drill and practice can be done a lot more effectively and at lower cost in other ways," he said.

A. Daniel Peck, a professor of education at San Francisco State University, agreed, saying a $200 dedicated microcomputer is often far more efficient. He said schools are being pressured by manufacturers into buying micros, and then don't know what to do with them. He likens the micro boom to the large-scale purchase of language labs in the early 60s, which he said are now "languishing."

Burtis added: "American educators have a history of seeking a bandwagon as a means of disguising the real problems. Manufacturers say, 'If you want to do the job for the American public, you must use computers.' And the teachers say, 'If you can't teach kids how to read and write, maybe this will work.' "

But Burtis also placed part of the blame on parents: "Parents are distressed about their kids not being able to learn, so they say, 'Well, if these computers are the answer, then buy some.' The teachers are being told,

continued



**Congressman Stark: Public financing of Apple's merchandising plan for the 80s?**

**EDUCATION** *continued*

'You'd better get computers.' "

Burtis feels computer literacy is not the requirement for future success manufacturers and some educators claim it is.

"It's a lie to say to kids, 'If you become computer literate you'll be more successful in the world of tomorrow,' " he said, "because what they learn on a micro will have no bearing on what they will need in the future."

Burtis said manufacturers are implying computer literacy comes before traditional skills. But, he said, effective computer use "is dependent on the individual being academically proficient. If you don't have academic proficiency, the computer isn't worth a damn."

Burtis and Peck are opposed to the Technology Education Act, the so-called Apple bill encouraging manufacturers to donate microcomputers to elementary and secondary schools. The bill, introduced by Fortney H. Stark, Jr. (D.-Calif.), was prompted by an offer from Apple to donate 103,000 computer centers across the country.

"It's just another means of asking the public to fund a merchandising plan for a manufacturer," Burtis said. "I think it's disgraceful. Even Apple blandly admitted that the purpose of the bill is to get Apples in front of kids so they'll grow up with them."

Peck added that the Stark bill doesn't consider additional expenses the school will face to use the equipment, such as teaching aids, hardware, and software.

Is there a solution to the problems the committee sees? Burtis thinks the key is public awareness.

"People should demand of computer manufacturers, 'Hey, design them to make our lives easier,' rather than be told 'Here's a computer; figure out something to do with it,' " he said. ■

## Tandy says hard to find **good numbers**

When it comes to computers in education, good numbers are hard to find, said Radio Shack Education Director William D. Gattis.

Gattis, on CompuServe's TRS-80 owners' special interest group, defended Tandy's refusal to release information on its computer sales to schools.

"The fact is," he said, "we simply do not have exact figures representing total TRS-80 computers delivered to schools through our 4,300 company-owned Radio Shack stores in the USA, nor the 2,000 independent Radio Shack dealers. So while we do monitor school purchase orders processed through our National Bid Department in Fort Worth, this does not represent all our sales to schools."

"Our own experience," he continued, "leads me to question the claims by other manufacturers regarding their percentage share of the education marketplace. With their products being sold to schools primarily through independent dealers, they are even further removed from accurate and objective sources of information than we are."

Gattis said: "The figures released by commercial research firms are particularly frustrating since all have different numbers and almost none compare to our own current estimates. Of those I've seen recently, only the year-old Report on Microcomputers in Education published by Creative Strategies International seems to us a credible document."

"The only sources of statistics we've encountered that we feel we trust," he noted, "are the surveys conducted by the various state departments of education. These agencies have the attention of public school districts throughout a state in a way that no commercial firm has achieved. Whereas a commercial firm might do well to see a 5 percent to 6 percent response to a survey mailing, most of the state surveys report information from over 50 percent of the school districts in the state, and a couple are approaching 100 percent."

Gattis reported:

- In Pennsylvania, the contracting unit for microcomputer purchases for local agencies during 1981 reported $1.52 million in purchases of five microcomputer brands—$1 million of it for TRS-80s.
- In Florida, the state Education Department's Management Information Services Center reported late in 1981 that more Florida schools, 401, had TRS-80s in them than any other micro.
- In Louisiana, the state Education Department reported in February 1982 that more than 60 percent of the schools used TRS-80s.
- In Oklahoma, a survey conducted by the state Department of Education in December 1981 reported 70 percent of the micros in that state's schools were TRS-80s.
- In Kentucky, a survey conducted by KET-TV, the public education tv station operated by Kentucky State University, showed, with 78 percent of the districts reporting, that more schools used TRS-80s than the next two brands of computer combined.
- In Idaho, a survey reported 56 percent of the micros in that state's schools were TRS-80s.
- In Montana, 37 percent of microcomputers in the schools were Tandy's, compared with 27 percent for the second-place company.

"We are not saying that the results from these states can be extrapolated to the whole country," Gattis observed. "On the other hand, we did not simply choose to share information only on those states' surveys which appear favorable to us. So far, every state survey we've seen corroborates Radio Shack's position as a leader in the burgeoning educational use of microcomputers."

"Tomorrow, next week, or next month, we may get results from all other states that show us in a much weaker position," he added, "but I doubt it. Just as much as I would encourage anyone to doubt the so-called statistics from any source that's based on projection or assumption instead of good, objective accounting." ■

# M-ZAL™

## Release Two Is Here!

**Our Users asked for it, and we made the best TRS-80 disk assembler even better! Now M-ZAL has:**

- Recursive MACROS with full symbolic parameter substitution
- Conditional assembly with unlimited nesting
- Text Editor warm-start/recovery option
- Support for all popular source file formats
- Load Module inspect/superzap using LINKER

**and the features that made it famous:**

- True multi-pass assembly
- Full Screen Option Menus
- ENTRY/EXTERN symbols, relocation, and object module linking
- 8 character labels with sorted symbol table
- Built-in symbol table cross-reference
- Nested *INCLUDES (source file chaining/nesting)
- Lower case support (Model 3 only)
- Over 175 pages of documentation including Z-80 Technical Manual
- Full Screen Text Editing and much, much more!

*M-ZAL provides a modular, structured development environment that makes programming your TRS-80 a truly enjoyable experience. Order yours today. (Requires 32k, 2 disk system)*

| | | |
|---|---|---|
| #1050-10 | Mod 1 | $149.00 |
| #1250-10 | Mod 3 | $149.00 |

### CAU Development Software— The Professionals Choice:

Full Screen Text Editor for BASIC: we were the first to give the TRS-80 this indispensable ability! A must for every BASIC programmer:

| | | |
|---|---|---|
| #1210-20 | Mod 1 & 3 | $29.95 |

T-ZAL: custom designed for the Model 3, this tape based assembler has many of M-ZAL's most desired features.

| | | |
|---|---|---|
| #1250-20 | Mod 3 only | $49.95 |

XBUG: for machine language analysis and debugging—this tiny (2.5k) but powerful program lets you do it all!

| | | |
|---|---|---|
| #1020-10 | Mod 1 | $19.95 |
| #1220-10 | Mod 3 | $19.95 |

FILEXFER: transfer any disk file between two TRS-80s. Use direct cable link or RS modems. Mod 1 and Mod 3 can be mixed. RS-232 and disk reqd.

| | | |
|---|---|---|
| #1040-10 | Mod 1 and 3 | $49.95 |

*"We've been using M-ZAL heavily for over two months now, and it's been worth its weight in gold."*
—Chuck Tesler
PROSOFT

*"...well written software that is long overdue."*
—Bruce Douglass
80 Microcomputing
BASIC Editor Review

*"During the first 3 days of use, M-ZAL saved more money in programmers time than we had paid for the product."*
—William Denman
Author of Asylum
MED SYSTEMS

*"The editor has at least doubled my productive capacity...Having spent a good deal of money on so called 'utilities' that don't work, it's a delight to find one that not only lives up to its claims but surpasses the advertising."*
—BASIC Editor user

| Our choice for the best TRS-80 DOS: | | |
|---|---|---|
| DOSPLUS 3.4 | Mod 1 | $149.00 |
| DOSPLUS 3.4 | Mod 3 | $149.00 |

Call or write for more information.
All products require level II.
Mail orders specify catalog #.
Send check, money order, or MC/VISA numbers and expiration date to:

**Computer Applications Unlimited**
**P.O. Box 214, Dept. ABM Rye, New York 10580**
**(914) 937 6286**

N.Y./State residents add applicable sales tax.
Outside USA and Canada add $10.00 shipping.
Dealer Inquiries Invited.

MasterCard VISA

COMPUTER APPLICATIONS UNLIMITED™
a div. of CAU, Inc.

120

TRS-80 is a TM of Tandy Corp. M-ZAL is a TM of CAU, Inc. Z-80 is a TM of Zilog, Inc.

See List of Advertisers on Page 451

# Health risks in V-games

*Overreaction to games leads to heart disease*

Concerned with the moral and intellectual effect of Pac-Man and its ilk on children, an army of parents across the nation is skirmishing with video game purveyors over regulation of arcade-style machines. The real danger, though, might not be to children, but to adults, who risk health problems by playing the games.

That's the indication of research conducted by Dr. Robert S. Eliot at the University of Nebraska Medical Center in Omaha.

Those indications became a deadly reality in Calumet, IL, where the stress from playing "Berserk" short-circuited an 18-year-old man's heart.

Eliot has tested more than 1,000 patients during the past five years to learn how they respond to stress. One conclusion he's reached is one-third of the nation's population are "hot reactors" to video games. Their bodies overreact to them, making them particularly susceptible to heart disease.

To identify hot reactors, Evans's lab has more than $300,000 in computerized equipment: machines to monitor blood pressure, heart rate, stroke volume, adrenal gland secretions, and resistance the heart must overcome to move blood through the circulatory system. In an article appearing in the May-June issue of *American Health*, author Stephen Kiesling described a demonstration he witnessed at the lab:

"One of Eliot's patients, a middle-aged administrator, has a base-line blood pressure of 134/89 when he sits down to play the game....

"But within seconds, though he doesn't feel any different, his body responds with signs similar to those of an athlete in action. His systolic pressure jumps to 207, while his diastolic pressure falls to 66. Meanwhile his pulse rate increases from 80 to 109 beats per minute, and his cardiac output doubles from 4.4 liters of blood per minute to 8.4 liters....

"So far the patient's body has reacted dramatically but in ways nature intended. Stress has triggered an outpouring of catecholamines, substances produced by the adrenal glands to stoke up the body to fight or flee.... To achieve the same elevated heart rate and blood pressure on a treadmill—or track— would take eight or nine minutes rather than the same number of seconds....

Photo by Omaha World-Herald

**Eliot: Drag racing with the brakes on.**

"Activity uses up the catecholamines as they are excreted into the blood stream, damping the physiological increase. But instead of springing into action, the patient continues to play the video game. His physiological responses soon take an ominous turn....

"The patient still feels much the same, but his body is no longer behaving like an athlete's; instead, he has symptoms like those that can lead to a stroke. Stress is squeezing his blood vessels like a clamp on a garden hose.

"His peripheral resistance triples, cutting his heart output back to 4.4 liters per minute. His blood pressure goes from 207/66 to 183/125.

"Eliot stops the game. 'What you are doing,' he tells the patient, 'is like drag racing a car with the brakes on.' "

Work with lab animals, Kiesling explained, has shown excess catecholamines can rupture small muscle fibers in the heart within five minutes. And autopsies of young men who suffered stress-related heart attacks showed their heart muscles had ruptured in seconds from stress-induced outpouring of catecholamines.

Kiesling wrote there are other stress producers in Eliot's lab—interviews, arithmetic tests, or holding a hand in ice water—that can create much the same physiological response. "The difference between those tests and video games," the author noted, "is that people don't go out of their way to engage in them, let alone spend countless hours and quarters for the experience."

According to Kiesling, Eliot said for hot reactors, video games "give all the excitement of competition without the physical component." He added Eliot fears "we may be heading toward a generation of hypertensives in this country where young people play video games instead of sporting activities which condition the body."

In the Calumet incident, the video game victim subjected an already weakened heart to the stress of playing "Berserk." Lake County, IN, Coroner Dr. Albert Willardo told *80 Micro* Peter Bukowski had "some pathology in his heart muscle which probably was due to a viral infection that was not detected."

He explained: "All of us have had virus infections where we've felt pretty lousy for a long time and then we get well and we assume there's nothing left, there's been no damage, but in fact many times the heart and other organs have suffered damage.

Because Bukowski's heart was in a weakened condition, Willardo said, "anything could have precipitated the cardiac arrhythmia which ultimately caused his death. It could have happened while he was playing tennis or just walking down the street."

Despite his findings, Eliot has no ill-feelings toward video games. He told the *Omaha World-Herald*: "Video games can be a form of relaxation for many people; a means of taking your mind off your troubles. But we use video games as a diagnostic tool because they offer one way of testing how an individual deals with pressure."

If a person is a hot reactor, he added, he or she should "seek appropriate help and advice that will allow them to handle game and real-life pressures. It doesn't mean they have to give up playing. If you react poorly to adding up a column of figures, you don't stop doing your income tax." ■

## MSS MANAGEMENT SYSTEMS SOFTWARE

### Business Programs

| | TRS-80 Model I/III | Model II |
|---|---|---|
| Proforma Cash-Budget | $125 | $175 |
| Proforma Income and Balance Sheet | 125 | 175 |
| Lease-Purchase Decision | 100 | 150 |
| Business Analysis Package (12 individual programs) | 100 | 150 |
| Procurement | 100 | 150 |

### Educational Programs

| | | |
|---|---|---|
| College Enrollment Projection | 100 | 150 |
| Gradebook | 50 | 100 |

All programs on disk with extensive documentation. Send for a brochure which fully describes the programs.

MANAGEMENT SYSTEMS SOFTWARE, INC.
2912 Brandemere Drive
Tallahassee, Florida 32312
Phone (904) 386-6561

457

---

— *Professional* —

## REAL ESTATE SOFTWARE

**for APPLE, TRS-80 & CPM SYSTEMS**

- **PROPERTY MANAGEMENT SYSTEM: $450**
  - Tenant History — Operating Stmt.
  - Late Rent Report — Building Reports
  - Vacancy Report — Utilities Report
  - Income Report — Tax Expense Report
  - Auto Late Charge — Prints Checks
  - Returned Checks — Prints Receipts
- **PROPERTY LISTINGS/COMPARABLES: $325**
  - — SCREEN BY —▶ Max/Min Price
  - 22 Items/Listing — Max Price/Income
  - 1000 Listing/Disk — Max Price/Sq Foot
  - Listing Memo Field — Min Cashflow
- **REAL ESTATE ANALYSIS MODULES: $50/Module**
  - Home Purchase — Tax Deferred Exchange
  - Income Prop Analysis — APR Loan Analysis
  - Property Sales — Loan Amortization
  - Construction Cost/Profit — Depreciation/ACRS Analysis
  - Loan Sales/Purchase — Loan Wrap Analysis
- **WORD PROCESSOR — WORD STAR: $295**

VISA

Realty Software Company™

At Computer Stores Everywhere or Order COD Direct
Cal Residents add 6½% Sales Tax
(213) 372-9419

Suite E, 1116-8th Street, Manhattan Beach, CA 90266

---

# TRS-80
## COMPUTER DISCOUNTS

- Factory Direct
- Best Prices Anywhere
- No Out-of-State Taxes
- 100% Radio Shack Warranty
- Free Price List

**SCOTT TASSO**
Owner
Radio Shack Associate Store #7053
**175 E North Delsea Drive**
**Vineland, N.J. 08360**
**800-257-0426**
**NJ 609-691-7100**

510

---

## TRS-80 MODEL I T.M.*
## GOLDPLUG - 80

Eliminate disk re-boots and data loss due to poor contact problems at card edge connectors. The GOLD PLUG - 80 solders to the board card edge. Use your existing cables.

CPU/keyboard to expansion interface . . . . . $18.95
Expansion interface to disk, printer, RS232, screen printer (specify) . . . . . . . . . . .$9.95 ea
Full set, six connectors. . . $54.95

VISA master charge

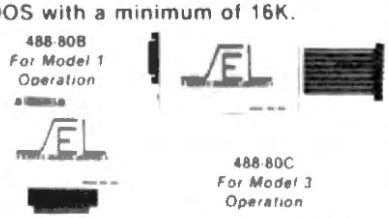
EAP COMPANY
P.O. Box 14, Keller, TX 76248
(817) 498-4242
*TRS-80 is a trademark of Tandy Corp.

216

---

## IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1 or 3, Level 2 or DOS with a minimum of 16K.




Model 488-80B or 488-80C Price $375
*+ shipping, insurance & tax*
***WHEN ORDERING SPECIFY DISK OR TAPE***

**SCIENTIFIC ENGINEERING LABORATORIES**
11 Neil Drive • Old Bethpage, NY 11804
Telephone: (516) 694-3370

203

**Trademark of Tandy Corp*
*There is no affiliation between Scientific Engineering Laboratories and Tandy Corp or Radio Shack*

---

## Maxell Floppy Disks

The Mini-Disks with maximum quality.


Dealer inquires invited. C.O.D.'s accepted. Call FREE (800) 235-4137

**PACIFIC EXCHANGES**
100 Foothill Blvd. San Luis
San Luis Obispo, CA 93401
In Cal. call (800) 592-5935 or (805)543-1037

207

---

## 96K MODEL 1

```
BIGMEM* replaces the 16K keyboard
RAM with 64K chips.  A custom mem-
ory controller mounts inside the
keyboard case to give you
 • 48K keyboard memory on power-up
 • plus access to 32K E/I memory.
 • 2.7K RAM above ROM is available.
 • Optionally overlay ROM with RAM.
 • BIGMEM works with TRS-80** DOS or
 • remaps kybd/video for 64K CP/M***

..$180 post paid with check or money
order.  COD add $3.  NYS residents
add sales tax.  BIGMEM incl. 64K RAM
chips, A&T controller board, internal
cable, wire, solder and instructions.

        MICROHATCH
        PO Box 501
        DeWitt, NY 13214
     (315)446-8031 after 6PM

Trademarks of *Microhatch, **Tandy
Corp. and ***Digital Research.
```

419

---

## ™TRS80 color

From the January 1981 issue of the CSRA Computer Club newsletter

> There was some amusement at the November meeting when the Radio Shack representatives stated that the software in the ROM cartridges could not be copied. This month's 68 Micro Journal reported they had disassembled the programs on ROM by covering some of the connector pins with tape. They promise details next month. Never tell a hobbyist something can't be done. This magazine seems to be the only source so far of technical informations on the TRS 80 color computer." Devoted to SS 50 6800 and 6809 machines up to now. 68 Micro Journal plans to include the TRS-80 6809 unit in future issues

To get the MOST from your 6809 CPU. This is the BEST SOURCE! The ONLY Magazine for the 6809 Computer. Months Ahead of All Others!

**68 MICRO JOURNAL**
5900 Cassandra Smith Rd.
Hixson, TN 37343

USA
1 Yr $24.50 2 Yr $42.50 3 Yr $64.50
*Foreign Surface Add $12 Yr to USA Price
Foreign Air Mail Add $35 Yr to USA Price
*Canada & Mexico Add $5.50 Yr to USA Price

213

---

## "80" DEMI-CASE

Shown 80 MDX DEMI-CASE

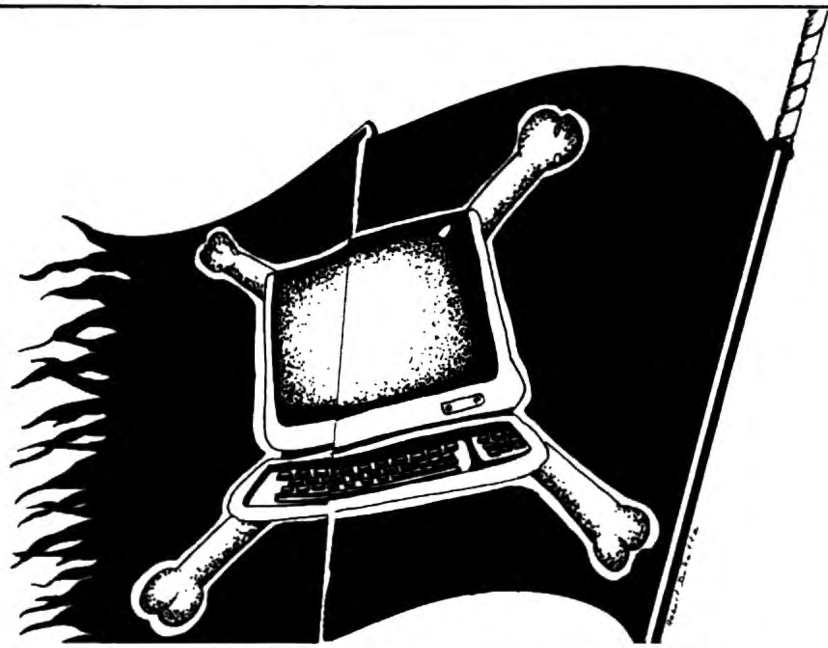
174

**Encloses LNW & MDX I & II**
**P.C. Boards, plus**
**Cabling, A/C Cords, etc.**
**Silver-gray Fiberglass**

**$32.50 plus $2.50 shipping**
**N.Y.S. residents add 7% tax**

**SYRACUSE R&D CENTER**
Box 125, Dewitt, N.Y. 13214
Specializing in Electronic Packaging

See List of Advertisers on Page 451



# Wordstar makers win lawsuit against pirates

*MicroPro International and Digital Research settle out of court with Sunnyvale softbucs*

By Kerry Leichtman
80 Micro Staff

Claiming the first major victory in the war against software thieves, the makers of Wordstar and the CP/M operating system won a $250,000 judgement against Data Equipment Inc. of Sunnyvale, CA, and its chairman, Daniel O'Rourke.

MicroPro International and Digital Research Inc. filed their lawsuit in November 1981, (see *80 Micro,* March 1982, page 378). In September, San Francisco Federal District Judge William W. Schwarzer entered a stipulated final judgement against Data Equipment.

The suit claimed Data Equipment produced and distributed unauthorized copies of MicroPro's Wordstar, Datastar, Wordmaster, and Mail Merge programs, and Digital's CP/M operating system. Data Equipment acted as an OEM distributing complete hardware and software systems to businesses. MicroPro and Digital Research discovered the misuse of their programs when Data Equipment customers called them for update information.

According to the terms of the settlement, Data Equipment will pay $250,000 to the plaintiffs, and will reimburse their legal expenses. In addition, Chairman O'Rourke is personally obligated to pay the plaintiffs $30,000.

While it is disputable whether this was the "first ever microcomputer software copyright infringement suit" settled in favor of a program's originators, experts said this and several other similar cases are fine tuning the 1976 Copyright Act into good protection for microcomputer product originators.

Seymour Rubinstein, president of MicroPro, also hopes the rest of the industry will be encouraged by the court's favorable decision, "Our success... should be taken as a sign that software manufacturers will act aggessively to protect their own products against all unauthorized duplication."

Daniel T. Brooks, of Computer Law Advisors, Washington, DC, thinks the present copyright laws are plenty thorough. Quoting from the bill that eventually became copyright law, Brooks told *80 Micro* during a telephone interview, "Under the bill, it makes no difference what the form, manner or medium of fixation may be; whether it is in words, numbers, notes, or any other graphic or symbolic indenture; whether embodied in a physical object, in written, printed, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device known or later developed."

To Brooks the law is very clear in definition and intent. The other school of thought likens Read-Only Memory to a permanent machine part. Its code, its proponents contend, speaks directly to the machine, not to human beings, and therefore is not copyright protected.

There is legal precedent in that argument. The U.S. Supreme Court ruled (*White v. Smith*) at the beginning of this century that piano rolls for player pianos were not subject to copyrights, because their code was not designed to be communicated to human beings. Brooks explained, "At that point in time the Supreme Court was saying the copyright laws applied only to those things which the copyright laws say they apply to and we don't see where this applies, therefore no protection."

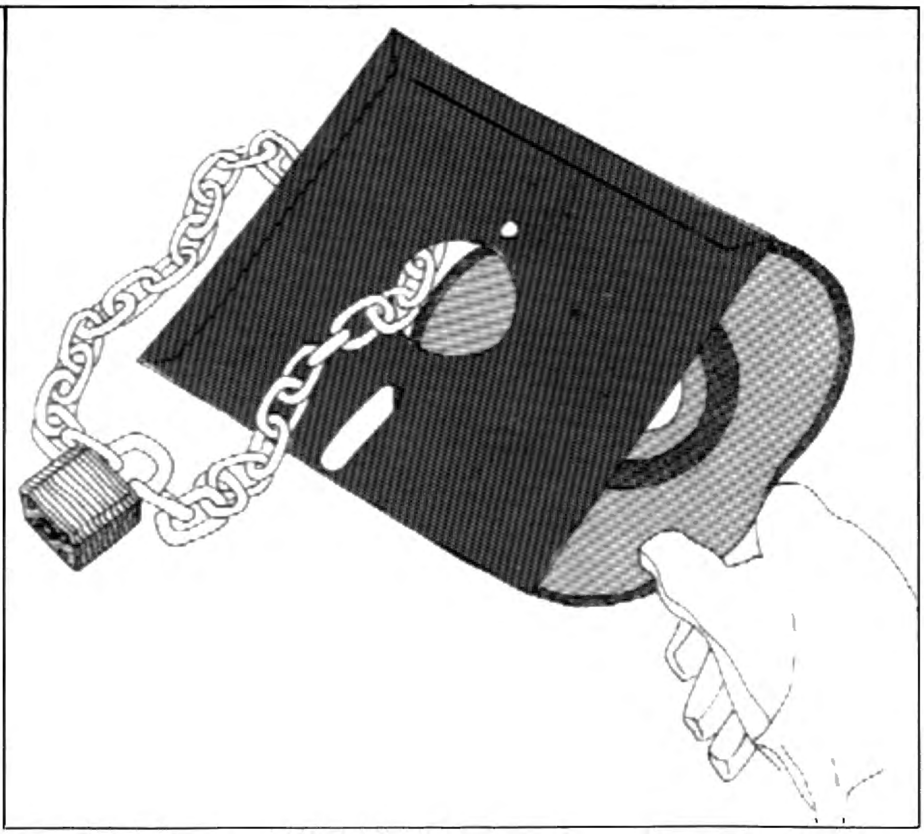
"In 1976," Brooks continued, "Congress expressly overruled that result. There's a whole paragraph in the legislative history that basically says this particular section is designed to overrule that notion. It says, 'This broad language is intended to avoid the artificial and largely unjustifiable distinction derived from such cases such as White-Smith, under which statutory copyright ability has been made to depend on the form or medium in which the form is fixed.'

"Whether it be the MicroPro case or *Tandy v. PMC,* the decisions are coming out in the right places now. The program, in ones and zeros, or in the volts and amps form is simply a representation of a copyrighted work and with the aid of a machine or device, may be communicated to human beings."

When a judge sits before two people where one is suing the other for a copyright infringement on a computer program he has to fully understand the issues before him so that he can make the right decision. Judges, like a lot of society, are not educated in the bits and

*continued*

**LAWSUIT** *continued*



bytes of computerdom. Brooks said the confusion, which led to some rulings inconsistent with the intent of the law, is understandable: "They do not have a working familiarity with what computers are about. Many of them look at it as an alien object. The consequence is that there is some room for confusion until responsible lawyers have framed the arguments in intelligible form providing the information the judiciary needs to make the right decision."

Will the framing take much time? Brooks doesn't think so: "I hope within a few years this particular point will go away. That isn't to say that the fertile mind of man won't come up with some new ones. It's only that this particular one will have disappeared. ■

# Networking Poet Laureate

## *E-poet says micros will shatter Gutenberg literature*

By John P. Mello Jr.
80 Micro News Editor

When the wired generation holds its convention to nominate candidates for Poet Laureate of the Network Nation, expect to see the name of David Hughes on the ballot.

Hughes, 55, believes some day microcomputers will shatter traditional forms of writing based on Gutenberg technology. He's paving the way for the new literature by championing the cause of vanity publishing on The Source, a computer-based information service owned by Readers Digest.

"I said to Readers Digest," the former West Point teacher told *80 Micro*, "I don't think you know what business you're in. You think you're selling information. I think you're selling electronic experiences."

He pointed out to Source officials: "Isn't it interesting that use of chat and mail—lateral contact between users—is far greater than the use of some of the dead data bases?"

Chat and mail were okay, but Hughes maintained subscribers needed more sophisticated lateral contact. How The Source met that need, the community planner and educator said, is "almost a revolution in electronic literature."

"What we did was create magazines on The Source," Hughes explained. Authors of the magazines receive royalties based on how often their work is accessed by subscribers. Over a three month period, Hughes' magazine earned him $826 in royalties.

But there are costs to vanity magazine publishers, Hughes noted. They pay for time—time needed to load their material into the system and time to store it there. Storage fees are on a sliding scale based on blocks of 2048 characters, about 380 words. For the first 10 blocks, a vanity publisher is charged 50 cents a block a month; for blocks 11 to 99, 20 cents a block; 100 to 999, 15 cents; 1,000 to 9,999, 10 cents.

"The magazine can definitely pay for itself if you put together an electronic bestseller," the Joy, CO, native said.

He added: "Imagine the kid that sits down, writes a piece, and can publish it nationally. He can begin to get his reputation. If he's got something to say, people will access him. If he doesn't, he's paying for it. It's self-correcting because he's not going to store forever garbage nobody reads."

Hughes' magazine, SourceTrek, includes 13 pieces ranging from a speech delivered to the Colorado Springs Rotary Club to ghost stories of Colorado. He uses his e-mag to experiment in writing beyond the Gutenberg dimension.

"A lot of word processing assumes the name of the game is to sit and use the screen to reproduce Gutenberg en

continued

# Give Your TRS-80* a Tremendous Boost with RACET COMPUTES Software

RACET COMPUTES Utility Software makes the TRS faster, more efficient, and easier to use. Our programming aids improve your productivity. Our reputation is for products that are professional in design and work **as advertised!!!**

**★ NEW SPEED-UP KIT FOR TRSDOS ★**
**V2.0a & 2.0b MODEL 11 & 16 • $99.95**

This software 'fix' to TRSDOS 2.0 is a MUST to users impatient with the speed of TRSDOS 2.0. Important features include: Boot speed improved by a factor of 2 1/2 times. Time and Date need only be entered once at power on. Data reads up to five times faster than in standard configuration. The system runs faster then when verify detect feature has been turned off but still retains the reliability of the verify detect function.

Special Note for SCRIPSIT users — Scripsit turns off the verify detect feature. The Speed-Up Kit turns it back on at no loss in disk I/O time. The Speed-Up Kit should dramatically reduce "blown" directories and garbled data.

**★ SUPERZAP for CP/M-80 • $99.95 ★**

You asked for it!!! And now it's here!!! Runs on any CP/M version 2.x system running on a Z80 CPU and 8-inch diskettes!!! Recover Blown Files! Access, modify, copy, zero, and print any sector on disk or in memeory — in Hex or ASCII. Full screen editor. File edit mode.

**★ SUPERZAP for TRSDOS II for Models II and 16 • $99.95 ★**

Recover Blown Files!! Now you can directly access, modify, copy, zero, or print any sector on your diskette or Radio Shack Hard Drive. Also included in this package is a SCAN utility to facilitate a disk search for a specified string. Documentation includes a description of the TRSDOS II directory structure.

**★★ Model II Fast Backup Utility ★★ $75**

5 TO 10 times faster backups!!! Full disk backup (including verify) in 55 **seconds!!!** on two drive system — 2:15 on single drive system. In business, time is money, and one BACKUP is worth 1000 tears!!

## RACET COMPUTES Utilities & Programmer Aids

**Generalized Subroutine Facilities (GSF) Mod I $25, Mod III $30, Mod II $50** The STANDARD against which all other sorts are compared! And then compare price!! Multi-key multi-variable and multi-key character string sorts. Includes zero and move arrays and sample programs.

**Basic Link Facility 'BLINK' Mod I and III $30 Mod II $50**

Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program. The chained program may either replace the existing program or can be merged by statement number. The statement number where execution is to start can be specified!

**Infinite BASIC Mod I $50, Mod III $60** Extends Level II BASIC with complete Matrix Functions and 50 more string functions. Includes RACET in-memory sorts!! Select only those functions you want to optimize memory.

**Infinite BUSINESS (Requires Infinite BASIC) Mod I and III $30** Complete printer pagination controls, hash codes, binary array search, and packed decimal arithmetic (127-digit accuracy).

**COMPROC Command Processor Mod I and Mod III $30** Auto your disk to perform any sequence of instructions that you can give from the keyboard!! Ideal for setting up dedicated applications.

**Disk Sort/Merge (DSM) Mod I $75, Mod III $90, Mod II $150** All machine language stand-alone Random file sort package. Establish sort specification in simple BASIC command file. Sorts multiple diskette files. Only operator action is to switch diskettes when instructed. Super fast sort times - improved disk I/O makes this the fastest Disk Sort/Merge available.

**KFS-80 Keyed File System Mod I and III $100, Mod II $175** Machine language BASIC ISAM utility provides keyed and sequential access to multiple files. Simple interface to BASIC. Binary tree index system provides rapid access to records.

**Utility Package Mod II $150** Recover blown diskettes. Includes Superzap, bulk copies, and other utilities for repair of blown diskettes. Complete documentation on diskette structure and guidance for repair. Other utilities include Dynamic DEBUG (with single-step, trace, subroutine calling, program looping and more), XCR, DISKID, Directory Catalog System, XPURGE.

✓468

**Development Package Mod II $125** Includes RACET Superzap, Apparat Disassembler and Model II interface to Microsoft Editor Assembler + and documentation for Mod I and uploading service. Assemble directly into memory, MACRO facility, save all or portions of source to disk, extended editor commands.

**Basic Cross Reference Utility Mod II $50** SEEK and FIND functions for variable, line numbers, strings, keywords. 'All' options available for line number and variables. Load from BASIC - call with control 'R'. Output to screen or printer.

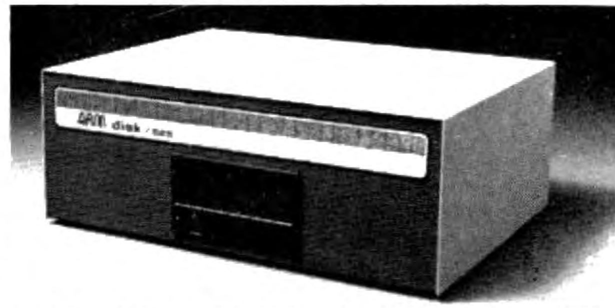
## FIELD PROVEN

### 5 to 140 MEGABYTES On Line for the TRS-80* Model II

Winchester and Cartridge Disk Drives available for immediate delivery!!

**5 MEGABYTE Winchester Disk Drive — CALL for PRICING**

LOWEST PRICES AVAILABLE ANYWHERE!!



**14 + Megabyte (formatted) ARM Winchester Disk Drive $3595**

includes ECC error detection and correction **FAST.** Service Contract Available ($30/month/drive) Multiplexor Available

**20 Megabyte CII Honeywell Bull Cartridge Drive $7995**

10 Megabytes fixed, 10 removable for the professional installation requiring Removable Media BACKUP. 60 and 120 Megabyte add-on Winchester drives available.

**10 Megabyte Cameo - Western Dynex Cartridge Drive $5995**

5 Megabytes fixed, 5 removable for the smaller professional installation. Removable Media BACKUP. Multiplexor available for up to four users!!

### HARD/SOFT DISK SYSTEM (HSDS) SOFTWARE

HSDS is available for the **Radio Shack (2.0(A) compatible)**, Cameo, Cynthia Bull, ARM, Corvus, OCS, Data Peripherals, and Santa Clara Systems hard disk systems and others!!!

Compatible with your Version 2.0(a) TRSDOS files. Compatible with your existing BASIC programs and files. Compatible with most machine language programs that use the standard calling sequence. You can access **both** your floppy disk drives and HSDS files **interchangeably!!** Complete utilities include HZAP (Hard Disk SUPERZAP), Directory Catalog System, Parameterized FORMAT, HPURGE (Bulk Copy/Purge Utility) and others.

The Hard/Soft Disk System (HSDS) Software has more than One Year's FIELD Experience. The latest HSDS version adds several enhancements including maintenance of system files on the hard drive, files as large as the disk, the ability to segment the disk as logical drives, and definable directory size. Uses 3.2K memory. Mixed floppy and hard drive operation is supported.

**HSDS for Radio Shack $500**
**HSDS for Other Drives $400**

RACET COMPUTES LTD.
Integrity in Software

1330 N. Glassell, Suite M, Orange, CA 92667 (714) 997-4950

**CALL FOR COMPETITIVE PRICING ON HARD DRIVE SUBSYSTEMS — BUY WHERE YOU CAN GET SOFTWARE SUPPORT!!**

CHECK, VISA, M/C, C.O.D. ON CERTAIN HARDWARE
TELEPHONE ORDERS ACCEPTED (714) 997-4950

*TRS-80 IS A REGISTERED TRADEMARK OF TANDY CORPORATION
CP/M IS A TRADEMARK OF DIGITAL RESEARCH

✓41 CIRCLE READER RESPONSE FOR FREE CATALOG

✓See List of Advertisers on Page 451

## HUGHES*continued*

route to a piece of paper," he contended, "but I think you can do things on a screen you can't do on a piece of paper."

Hughes maintained that until now, writing has operated on a fixed data base—the 26 letters of the English language—and required our eyes to be dynamic. He reasoned, "If you take the dynamic things of games—the graphics—and apply them to letters, I've got a sneaky suspicion we will create a new form of extension."

Hughes playfully emphasizes this point in the opening of his magazine. He prints on the screen a quote from *The Rubaiyat of Omar Khayyam*:

The Moving Finger writes;
and, having writ,
Moves on: nor all your Piety
nor Wit
Shall lure it back
to cancel half a Line,

Then he reverses the cursor and erases half a line.

On another occasion, Hughes was inspired by an Orchestra 80 rendition of a Scarlatti pastoral he downloaded from CompuServe. He wrote for his magazine an article titled "Sunday on a CRT." "When I reached the point in text," he said, "where in the print world you would describe the music, suddenly I realized, 'Why should I do this?' I went back to the file, picked a fragment of code, embedded it in the text and went on."

During the last two years, Hughes has pushed beyond the limits of the alphabet in what he calls "word. dance." He described the technique as "creative electronic literature beyond the edges of Gutenberg English."

"One of my first tasks," he explained, "was to eliminate punctuation on the screen, on the ground there's a brain equivalent and ways to make emphasis occur on the screen that are superior."

"This convention called punctuation," he continued, "is nothing more than ASCII English—in other words, code. When I speak, I don't say comma or period, but on paper, I have to have a comma to pause. On the screen, I don't have to have a comma. I can actually pause."

Hughes and some of his students practice their Electronic English with the help of a micro.

Hughes isn't the only artful experimenter on The Source. People have composed electronic duets in verse. They have assumed fantasy roles, like the Pink Elephant, and conducted improvisational "theater" on the message system. "That's damn exciting," the electronic bard said, "because I think it will draw people into the computing world who call computers dehumanizing. I say they are the most humane thing ever invented."

He added, "It is my absolute conviction that until the people with words begin to understand computing in a cultural context, not as a clerical skill, not as a computer skill; that they see telecommunications as a device that can do things other technologies cannot do; then we're not going to drag enough of the human race into this."

One way to get people to understand the cultural ramifications of computers is through education. But much of that has become part of the problem, according to Hughes. He declared: "This myth that everyone has got to learn to program is one of the biggest impedi-

continued

**HUGHES***continued*

ments to why people aren't getting into computers."

"Computer literacy has come to mean programming classes," he continued. "Even when Radio Shack offers teachers programming classes, they find very few English teachers. And after the few English teachers that show up learn to program, they ask, 'Now what?'"

"A woman can learn how to drive a car without learning how to build an internal combustion engine," he observed. "I say to my students, 'I will teach you how to write on this thing. Then it might become useful for you to learn how to program.' Just like a woman might learn how a carburetor works so she won't get ripped off by a mechanic."

Hughes has pioneered a college credit course for Colorado Technical College called Electronic English. The entire course is conducted over The Source.

He said his course focuses on screen characteristics and teleconferencing.

"People 'recite' in class," he noted. "I grade them on screen readability. You're taught in speech class about how to stand when you talk, eye contact, and having your voice loud enough to be heard. In electronic speech, if you want to be an effective writer, you have to be acutely conscious of what a person is reading, how they're reading it, and the rate they're reading it.

"If you can't handle a keyboard, don't know what a smart terminal program is, don't know what 300 baud and ASCII code is, don't visualize what a guy with an Apple with 40 characters uppercase only is seeing while you're writing in 64 characters upper- and lowercase, you don't know how to speak to another person."

Guest teachers in Hughes' classes don't give lectures. They give electures, plugging their lessons into The Source where students access them.

Hughes explained his students put in three hours of class work a week but don't meet at regular hours. However, his course differs from correspondence courses because a student is connected to his classmates and the instructor.

Two or three times during the 11-week course, he continued, students must "bring their brains to the classroom." During those sessions, students upload 2,000 word compositions and the class discusses them on the system in real time.

Hughes also teaches his classes about software aids to writing like Aspen Software's Grammatik, Proof-Edit, and Proofreader. And he leads discussions on what freedom of information means in the electronic era.

"Why don't we create programs that allow the individual to screen, by a filter, words offensive to him," he opined. "System operators should never try to censor what a speaker says because in the microcomputer age, readers can be their own censors based on their own value system."

Hughes' students attend his classes from all over the United States and the world. They've included a former lieutenant governor of Alaska, a hay and grain dealer in Iowa, a travel agent in Australia, and a professional writer in London.

The electronic educator believes breaking down geographical barriers through microcomputer networks can preserve cultural diversity.

He cited the case of Anne Charles, a 21-year-old Cree Indian, who travelled 350 miles from L'Orange to Regina, Canada, to ask Hughes if she could take his course. Hughes, who was in Regina to address the International Association of Business Communicators, told the association's conferees:

"Anne Charles wants to be a journalist. She does not have to move to Montreal and become French-Canadian to do that. She could learn journalism while staying in L'Orange, then turn around and export the culture of her people for pay."

Later, Hughes explained to Charles how she could use an Epson printer to preserve the Cree language: "For her uncles, who speak Cree and write in a cunieform, I did a quick little program that spit out of the Epson the language of her people in cunieform form. I said her brothers and sisters could learn how to put the speech of her people on an Epson and communicate it over network lines."

By using microcomputer technology in this fashion, Hughes contends society "can really start to solve the cultural and economic problems of the world."

"We already provide highways for people's bodies to get them to big centers," he observed. "Why don't we provide highways for their minds?" ■

**"We already provide highways for people's bodies. Why don't we provide highways for their minds?"**

## COLOR COMPUTER

**PROGRAMMABLE**

# SOUND MODULE

**AT LAST — the COLOR COMPUTER's audio capabilities even SURPASS its great graphics! If you can program in *BASIC* — plug in the PSM cartridge and start to create life-like explosions, engines, animals, wind, water and other sounds limited only by your imagination. Complex sound effects are continuously maintained independently of your program. The PSM is a power-packed hardware/software combination and 100% compatible with ALL models of the Color Computer.**

**PSM + complete instructions ... $98.95**

Available from:

**MAPLE LEAF SYSTEMS**
**P.O.BOX 2190, STATION 'C', Downsview**
**Ontario, CANADA M2N 2S9** ✓92

---

# MOD 2

**MERGIT-GLII**

LINKS MODEL 2 ONE DISK GENERAL LEDGER (PN 26-4501) PROGRAM WITH SCRIPSIT™ 2.0 ALLOWING ULTIMATE FLEXIBILITY IN REPORT FORMATS AUTOMATICALLY............$89.95

**GAMES-V**

5 assorted GAMES for mod2 $19.95

SAVE!! Buy both for..... $99.95

INDEPENDENT SOFTWARE
POB 3126
FEDERAL WAY, WA 98003
(206)941-6022 WA res add 6.5% tax
no COD orders please
™trademark of TANDY CORP ✓494

---

$$$ **D/Punch?** $$$

D/Punch Conversion Kits will help you **SAVE MONEY, SAVE STORAGE SPACE, SAVE TIME** D/Punch Kits are engineered so you can quickly, accurately & safely make alterations necessary to let you use both sides of your H or S sectored SINGLE-SIDED Diskettes Converting 1 box of diskettes will more than pay for each D/Punch Kit. • Hub-Reinforcer Kit accurately positions Hub-Reinf Ring to hub opening • Disk Sleeve lets you replace your lost or damaged sleeves Indicate your system config with order

| | |
|---|---|
| a D/Punch Kit A — (5¼" drives & eq ) for ex [Apple, Atari, CBM 4040, Franklin] | $ 6 95 |
| b D/Punch Kit B — (5¼" drives & eq.) for ex [TRS 80 (Mod I, III), IBM, Osborne, Zenith] | 14 95 |
| c D/Punch Kit C — (8" drives & eq ) for ex [Altos, Zenith, TRS 80 Mod II] | 16 95 |
| d Labels — [100 pcs] | 3 00 |
| e. Write-Protect Stickers — [100 pcs] | 2 65 |
| f Hub-Reinforcer Kit (5¼") | 10 99 |
| g Hub-Reinforcer Kit (8") | 12 99 |
| h Hub-Reinforcer Rings (5¼") — [50 pcs] | 5 85 |
| i Hub-Reinforcer Rings (8") — [50 pcs] | 6 20 |
| j Disk Sleeves (5¼") — [10 pcs] | 2 35 |
| k Disk Sleeves (8") — [10 pcs] | 3 85 |

Add $2 shipping & handling for each $15 order increment AK, HI, & overseas cust write for S&H charges MA res add 5% tax Send check or money order to

**D/Punch Co (12)** ✓506
**P.O. Box 201, Newton Mlds, MA 02161**
**Tel (617) 964-2126 Telex: 4991009 CHTRI**
©1982 D/Punch Corp., Pat Pend

---

## MEMOREX Diskettes

Top-quality Memorex Diskettes from **Tech•Data**, your complete word and data processing supply center. Dealer inquiries invited.

**Call Toll Free**
**1-800-237-8931.**
**In Florida, call**
**813-577-2794.**

**Tech•Data Corporation**
3251 Tech Drive North ✓440
St. Petersburg, FL 33702

---

**W..I..D..E REPORT SUPPORT**
MOD II/16 and DAISY WHEEL II

**Print 15 and 20 Characters/inch**

D L P **Company**
6798 Wetheridge Dr.
Cincinnati, Ohio 45230
513 232-7791 ✓413
VISA OR MASTERCARD ACCEPTED

$75.00

260 characters in just 13" at 20 CPI; 170 in 7". Excellent quality 15 CPI printing; 195 Char. in 13"; 120 in 8". SCRIPSIT™ 2.0 TRSDOS™ 2.0/4.1, VISICALC®, PROFILE™, BASIC, COBOL, etc.

---

**EPSON MX-80** CARTRIDGE RIBBONS — **$7.00**
MINIMUM ORDER 3

**LABELS** — **$2.99 per M**
$\frac{15}{16}$x3½ Pin-fed white
PRESSURE SENSITIVE
Minimum Order 1up 5,000 2up 10,000 3up 15,000 4up 20,000

**PAPER** — **$20.00**
20# white
1250 sheets

**Prices Include Shipping!**
Check must accompany orders
Ohio residents add 6% Sales Tax

**DATACOM**
COMPUTER SALES AND SUPPLIES
P.O. Box 02294 - Cleveland, OH 44102
**(216) 281-8820** ✓253

---

**ABS Suppliers**
3352 Chelsea Circle
Ann Arbor, MI 48104
(313) 971-1404

**B17 TAPE OPERATING SYSTEM** will let you
- Save & Load programs 6X faster than Model-1
- Save & Load Data Arrays over 800X faster
- Backup standard System tapes that load-n-go

**$14.95** Specify 16K, 32K, or 48K

**MICRO-SCOPE** diagnostic for ANY Disk System Tests MEMORY, FDC, DISK READ/WRITE, PRINTER & RS-232. Built-in Disk Drive timer Exceptional value.
**$16.95**

**WORDSMITH** Word-processing program in BASIC A 'Trainer' for more complex W/P programs. Easy-to-use. Specify Tape/Disk version. Tape version requires B17 Tape Operating System (above) Disk version requires 32K RAM-up. Uses MX-80 Printer
**$14.95**

Add $1.50 postage/handling per Model-1 item
Write for our FREE Reference Sheet ✓173

---

# SECURE PROGRAMS

**AND ASCII FILES**
**WITH**
**CODE 4**

**COMES WITH**
**LIST SOURCE**
**AND BRUN UTILITY**

**$19.95**

**SEND CHECK OR MONEY ORDER TO**

**HPB VECTOR CO.**
**130 CENTER ST.**
**EAST STROUDSBURG**
**PENNSYLVANIA 18301** ✓487

***ALLOW TWO WEEKS DELIVERY***

---


**SUPER GRAPHICS SOUND**

| | | |
|---|---|---|
| FOR MODEL I AND III TRS 80 | 16K TAPE | $14.95 post paid |
| | 32K DISK | $19.95 post paid |

✓323

BE SURE TO INCLUDE MODEL AND WHETHER FOR DISK OR TAPE
VISA ACCEPTED BUT REMEMBER TO INCLUDE EXPIRATION DATE

SEND ORDER TO KATAHDIN SOFTWARE
35 W. BROADWAY
LINCOLN, MAINE 04457

KATSOFT

Dealer inquiries welcomed

# PULSETRAIN

## Will the meek inherit the market?

Industry watchers were surprised by the meek pricing moves made by Atari and Commodore in the face of Texas Instruments' $100 rebate plan, according to *Electronic News*.

That weekly speculated soon after TI announced its rebate plan Atari was prepared to give away $150 in software and gift certificates to buyers of its low-end micro and Commodore would be knocking $50 off its VIC.

The official word from the two firms, however, was less dramatic. Atari—faced with potential resistance from retailers for whom software is a high margin item—anounced it would offer six $10 certificates good for software and peripherals with the sale of its 400 computer. And Commodore revealed a $25 to $40 VIC pricecut, depending on purchase volume.

*Electronic News* also reported the low-end-micro price war has forced Commodore to change some of its production and marketing schedules. Production of the VIC, set for 50,000 units a month, has been upped to 70,000 units a month. That, in turn, has reduced Commodore's commitment to its new game micro, the MAX Machine.

Commodore had planned to place equal marketing emphasis on the VIC and MAX Machine, but the firm's vice president for marketing, Kit Spencer, told the newspaper: "We have decided to make VIC our major product for the Christmas season."

**Roach: We've been tarred and feathered.**

*Electronic News* said Tandy's official line on the fireworks in the low-end market is its company-owned stores insulate it from price bombings. Tandy's president and chief executive officer, John Roach, told the industry weekly: "We're getting tarred and feathered with people guessing our response to the TI, Commodore and Atari business. It's difficult to understand. We're a retailer and whatever our strategy is, we will not talk about it until we implement it."

## Computer division boosts Tandy's profits picture

Tandy Corporation experienced a 32 percent jump in profits for its fiscal year ending June 30. Contributing to the increase was a 72 percent hike in shipments from its computer division.

The $224 million in profits for the 1982 fiscal year compares to $169 million earned in FY1981.

The Fort Worth firm estimated computer revenues account for 31 percent of its annual revenues.

Computers are in a strong growth cycle, the company noted, and calendar 1982 will be the first big year for home sales.

Tandy projects sales of its product will grow three to four times in 1982, compared to 1981.

The firm also said early orders for its Model 16 have exceeded expectations.

## Tandy to give National Science Foundation hardware

Tandy Corporation announced an agreement with the National Science Foundation to provide computer equipment to institutions for use in education programs in science, mathematics, and engineering through the Fort Worth firm's TRS-80 Educational Grants Program.

The foundation, an independent agency of the U.S. government, will be awarding the computers as the equipment portion of projects supported by its ongoing development-in-science education program.

Charles Phillips, Radio Shack senior vice president for special markets, told the *T.H.E. Journal* the program is "a timely and important step supporting the need for renewed emphasis on science education in this country."

He added that "this participation with NSF is in accord with the philosophy behind our Tandy TRS-80 Educational Grants Program—that is, rather than simply giving out hardware, we want to support projects that have a meaningful plan for educational applications using that hardware."

# PROGRAMMING TOOLS FOR YOUR TRS-80™ MODEL I AND MODEL III

## INSTANT ASSEMBLER

**The INSTANT ASSEMBLER** is a powerful disk or tape-based assembler and debugger for the TRS-80. Now you can assemble directly to memory and immediately debug your program with the built in single stepping debugger. Quickly switch from assembler to debugger and back again without losing the source code. This feature makes INSTANT ASSEMBLER an excellent learning tool for assembly language programming.

**INSTANT ASSEMBLER** is absolutely unique among tape based assemblers in that it produces relocatable code modules that can be linked with the separate LINKING LOADER, which is supplied in two versions for loading programs into either high or low RAM. This lets you build long programs with small modules. INSTANT ASSEMBLER also features immediate detection of errors as the source code is entered, a compactly coded source format that uses 1/3 as much memory as standard source, and many operational features including single stroke entry of DEFB and DEFW, pinpoint control of listings, alphabetic listing of symbol table, separate commands for listing error lines or the symbol table, block move function, and verification of source tapes.

**INSTANT ASSEMBLER**'s debugger provides single stepping with full register displays, decimal or hex entry of addresses, forward or backward memory displays, disassembly of object code in memory, memory display in ASCII format, and hex-to-decimal or decimal-to-hex conversion. The single stepper will step one instruction at a time or at a fast rate to any defined address.

**INSTANT ASSEMBLER** occupies less than 8400 bytes of memory. In a 16K machine this will leave you enough memory to write assembly language programs of around 2000 bytes. This and its module-linking feature make INSTANT ASSEMBLER ideal for users with only 16K machines. The instruction manual may be purchased separately for $3, which will apply towards the purchase of the INSTANT ASSEMBLER. In addition to disk I/O, the disk version includes a stand-alone version of the debugger.

**Specify Model I or Model III. TAPE INTASM ...................... $29.95 on tape**
**Specify Model I or Model III. DISK INTASM ...................... $35.95 on disk**

## RESTORE DAMAGED TAPES WITH RESQ2

Cassette recordings are subject to several types of damage. Thin spots in the oxide, dirt, voltage fluctuations while recording, or stray magnetic fields can all contribute to lost or added bits. RESQ2 was written to provide a method of restoring tapes that can no longer be loaded for these reasons. It can restore BASIC, SYSTEM, ASSEMBLER, and DATA tapes. RESQ2 compares two copies of the damaged tape to attempt a restoration, though restoration can often be accomplished with only one copy. After the damaged data is corrected in memory, a new tape may be recorded and verified which does not contain the errors. The success rate of RESQ2 will depend on the severity and quantity of errors. RESQ2 comes with a comprehensive user manual and examples of two types of 'crashed' programs to practice on.

**Specify Model I or Model III. RESQ2 ............................ $19.95 on tape**

## RAM SPOOLER AND PRINT FORMATTER

This program is a full feature print formatting package featuring user defineable line and page length (with line feeds inserted between words or after punctuation), screen dump, printer pause control, and baud rate selection. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Works with cassette or disk systems. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. Output may be directed to either the parallel port, serial port, or the video screen. 80 Microcomputing said "I can only give my highest recommendation of Spooler and Mumford Micro Systems."

**Specify Model I or Model III. SPOOLER ........... $16.95 on tape, $21.95 on disk**

## DUPLICATE SYSTEM TAPES WITH CLONE

Make duplicate copies of almost any tape including Basic, SYSTEM, data lists, assembler source, or "custom loaders". The file name, load address, entry point, and every byte (in ASCII format) are displayed on the video screen. Model III version allows changing tape speed so you can load in a tape at 500 baud and write it out at 1500.

**Specify Model I or Model III. CLONE .............. $16.95 on tape, $21.95 on disk**

## MACHINE CODE FAST FOURIER TRANSFORM

Written by Dr. A.H. Gray, Jr., co-author (with J.D. Markel) of the classic text "Linear Prediction of Speech", this complete package includes 3 versions of the machine language FFTASM routine assembled for 16, 32, and 48K machines, a short sample Basic program to access them, a 10K Basic program which includes sophisticated interactive graphing and data manipulation, and a manual of instructions and examples. The machine language subroutines use variables defined by a supporting Basic program to make data entry and retrieval automatic, without PEEKs and POKEs. They perform 20 to 40 times faster than their Basic equivalent (256 points in 12.5 seconds), and handle up to a 1024 point complex FFT. The FFT is useful in analyzing stock market and comodity trends as well as for signal analysis.

**Specify Model I or Model III. FFTASM ........................... $49.95 on tape**
**FFTASM on disk with source code ....................................... $69.95**

## INSIDE LEVEL II
### The Programmers Guide to the TRS-80 ROMS

**INSIDE LEVEL II** is a comprehensive reference guide to the Model I and Model III ROMs which allows the machine language or Basic programmer to easily utilize the sophisticated routines they contain. Concisely explains set-ups, calling sequences, and variable passage for number conversion, arithmetic operations, and mathematical functions, as well as keyboard, tape, and video routines. Part II presents an entirely new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer including tape formats, RAM useage, relocation of Basic programs, USR call expansion, creating SYSTEM tapes of your own programs, interfacing of Basic variables directly with machine code, and special precautions for disk systems. INSIDE LEVEL II was reviewed in the April 1982 issue of 80 Microcomputing which said "The book has no flaws; it is a perfect gem." Byte Magazine said "I recommend this book to serious machine language programmers."

**Includes updates for Model III. INSIDE LEVEL II .......................... $15.95**

## SINGLE STEP THROUGH RAM OR ROM

STEP80 allows you to step through any Basic or machine language program one instruction at a time, and see the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The top 14 lines of the video screen are left unaltered so that the "target program" may perform its display functions unobstructed. STEP80 will follow program flow right into the ROMs, and is an invaluable aid in learning how the ROM routines function. Commands include step (trace), disassemble, run in step mode at variable step rate, display or alter memory or CPU registers, jump to memory location, execute a CALL, set breakpoints in RAM or ROM, write SYSTEM tapes, and relocate to any page in RAM. The display may also be routed to your line printer through the device control block so custom print drivers are automatically supported.

**Specify Model I or Model III. STEP80 ............. $16.95 on tape, $21.95 on disk**

## SMART TERMINAL PROGRAM

This machine language program may be used as a smart terminal with time share systems or for high speed file transfers between two disk-based micros over modems or direct wire. It is menu driven and extremely simple to use. Functions include real-time terminal mode, save RAM buffer on disk, transmit disk file, receive binary files, examine and modify UART parameters, program 8 custom log-on messages, automatic 16-bit checksum verification of accurate transmission and reception, and many more user conveniences. Supports line printers and lowercase characters. With this program you will no longer need to convert machine language programs to ASCII for transmission, and you will know immediately if the transmission was accurate.

**Specify Model I or Model III. TELCOM ........................... $39.95 on disk**

## DISK INDEX VERSION 3.0

Our excellent disk indexing program has now been entirely rewritten in machine language. It will run on either a Model I or Model III and catalog disks for either machine regardless of which one is running it. (Model I owners must have double density to catalog Model III disks.) DISK INDEX will assemble an index of your entire program library by automatically reading program names and free space from each disk directory. The index may then be alphabetized or searched for any disk, program, or extension. Disks or programs may be added or deleted, and the whole index or any part may be sent to the printer. The index itself may also be stored on disk for future access and update. A 48K machine will hold over 2500 programs in each index, and you may build as many indexes as you need. Version 3.0 runs substantially faster than our previous version and works with any operating system written for the Model I or Model III except CP/M.

**Specify Model I or Model III. INDEX 3.0 ......................... $29.95 on disk**

## 4 SPEED OPTIONS FOR YOUR TRS-80

The SK-2 clock modification allows CPU speeds to be switched between normal, an increase of 50%, or a 50% reduction, selectable at any time without interrupting execution or crashing the program. Instructions are also given for a 100% increase to 3.54 MHz. The SK-2 may be configured by the user to change speed with a toggle switch or on software command. It will automatically return to normal speed any time a disk is active, requires no change to the operating system, and has provisions for adding an LED to indicate when the computer is not at normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with socketed IC's and illustrated instructions.

**Model I only. SK-2 ....................................................... $24.95**

## RAMTEST

This machine language program is a very thorough test for several types of RAM errors and will indicate which chip, if any, is faulty. It also includes a separate test for power line glitches.

**Specify Model I or Model III. RAMTEST ........... $12.95 on tape, $17.95 on disk**

## MUMFORD MICRO SYSTEMS

**ORDERING:** Complete satisfaction is guaranteed or a full refund will be made. Include $2.00 for postage and handling. California residents add 6% sales tax. Visa, Mastercharge and COD orders accepted. **SPECIFY MODEL I OR MODEL III.** Dealer inquiries invited.

Box 400-E Summerland, California 93067 (805) 969-4557

✓ 67

Quality software since 1978

# PULSETRAIN

## *Cosmo* discovers Network Nation

The Cosmo girl has discovered the Network Nation and it looks like an 8.5. "*Forget* dating through the personals or friends' fixups!" *Cosmopolitan* magazine exhorted its readers in its September issue. "An electronic brain right in your own living room can introduce you to thousands of men...in seconds!"

*Cosmo* discovered the wonders of networking through author Elizabeth Ferrarini, who stumbled on The Source while looking for a man "who'll be G.I.B., pay for dinner, and treat me like precious porcelain...have a college degree, solid profession, good looks, weight in proportion to height, and make at least $50,000 a year."

Despite her assets—magna cum laude college graduate, homeowner, late-model sports car driver, and good teeth—Ferrarini found "Mr. Terrific was being strangely elusive."

"Just in time to avert terminal loneliness," she wrote, "a friend told me about a company that allows nearly 20,000 computer owners to communicate with each other in a flash. When I learned that over 90 percent of these computer buffs are males, their median age is 35, and they are college-educated professionals with an average yearly income of almost $50,000, I knew the object of my capitalistic fantasies must be among them."

After purchasing a micro and learning how to use The Source, the *Cosmo* author realized the potential of networking: "Using my computer to communicate via the telephone line, I could dash off torrid love letters, place ads on an electronic bulletin board, and even carry on 'conversations' with other subscribers by way of our respective computers' keyboards. Admittedly, all of this sounded to me like science fiction. *Imagine* getting my mail delivered by an electronic brain and communicating with people I'd never seen. A smart *chimpanzee* could be sending me electronic billets-doux."

Ferrarini placed a lonely hearts message on The Source's electronic bulletin board and 24 hours later, she had 16 e-letters, 12 of them with one thing in common: a trip to her home state within the next two months.

Overall, the *Cosmo* author appeared satisfied with the chipsters her ad attracted. "Although I've encountered my share of duds," she wrote, "computer owners are, on the whole, a likeable lot. With my desk-top model and a telephone, I can plug into an endless supply of available males—and keep my checkbook balanced, too."

## Computer life after 40?

Is there computing after 40? The Source says yes. In a statement on a recent survey it completed on its subscribers, the information utility said the average age of its users is 42, three years older than the average age in 1981. The Source also noted 64 percent of its subscribers are over 35, compared to 56 percent last year.

In a related matter, The Source announced it is carrying an electronic version of the *U.S. News Washington Newsletter*. According to a statement from the information utility, each week the letter carries about a dozen stories on investment opportunities, economic trends, and key government activities.

## Apple millionaire raps Woodstock

Although pre-concert publicity touted the US Festival as the Woodstock of the 1980s, the man

# SMART MOVES...


## MI & MIII DOSPLUS 3.4 $110

**TRS-80 MODEL III MICROCOMPUTER**
2 Single Sided 40 TRK Drives ........ $1,895

**MODEL III DISK CONTROLLER KITS**
J & M Systems .......... $ 325
A.M. Electronics ........ $ 325

**RS-232 INTERFACE KITS**
V.R. Data RS-232 Kit ...... $ 90

**RANDOM ACCESS MEMORY**
200 ns N.E.C. RAM ...... $ 2

**DOUBLE DENSITY KITS**
Model I A.M. Multiplier ... $ 99

**CANON DISK DRIVES**
Slimline 40 TK Single Sided Complete .......... $ 235
Slimline 40 TK Double Sided Complete .......... $ 320

**SANYO EXTERNAL DRIVE** .............. $ 350

**ORC-85 DISK VERSION** ............ $ 99

**ORC-90 DISK VERSION** ............ $ 99

**SMITH-CORONA TP-1 PRINTER** ........ $ 675

**BARE DISK DRIVES — Tandon Drives**
Single Sided 40 TRK Drives ......... $205
Single Sided 80 TRK Drives ......... $ 320

40 TK 5 MSEC Stepping, gold contacts.
Complete Case $230
Bare Drive $175

**EPSON PRINTERS** $599
MX 82 F.T

**OKIDATA LINE PRINTERS**
Microline 82A .......... $ 450
Microline 83A .......... $ 695
Microline 84 .......... $1,100

**ASG-3 WINCHESTERS FOR TRS 80 MODEL III**
5m Byte Formatted ...... $1,795
10m Byte Formatted ...... $1,995

These drives plug into the 50 pin I-O expansion port. System includes host adapter, cables, fan-cooled cabinet. An LDOS compatible driver is included with the kit for those who already own LDOS.

**90 Day Warranty**
**One Year Extended Warranty Available**

EPSON Factory Authorized Service Dealer

Call or write for Free Brochure:
U.S. Prices F.O.B. Winter Garden, FL

**ORDERS ONLY:**
**1-800-327-6590**

✓178

TRS is a trademark of Tandy Corp.
LDOS is a trademark of Logical Systems, Inc.

**all Systems Go**
© 1981

638 S. Dillard St.
West Orange Shopping Center
Winter Garden, FL 32787-0915
Voice: (305) 877-2830
Data: (305) 877-2829

✓ See List of Advertisers on Page 451

who picked up the $13 million tab for the event disavowed any similarities between the rock-a-thons.

Millionaire and Apple computer founder Steve Wozniak rapped Woodstock in an article appearing in the *Providence Sunday Journal*. "If I listed the things that were wrong with Woodstock, it would overflow one diskette on my word processor," he said. "Woodstock was really a lot of lies, miserable lies. They were a bunch of hippies spending money that wasn't even theirs. They didn't even have any sanitation."

Sanitation is something they had plenty of at Wozniak's concert. At the 500-acre park carved to Wozniak's specifications by the construction superintendent at Disney World, there were 2,500 latrines, the best commode-to-concert-goer ratio in history.

Apparently, US Festival planners hoped its participants would react differently to their concert than their Woodstock predecessors. While people attending Woodstock professed to be members of a mythical love-peace entity called "The Woodstock Nation," rock impresario Bill Graham told the *Journal* that an US Festival goer—gazing at laser lights and multi-dimensional visual effects—was supposed to "feel like an extra in a Steven Spielberg movie."

"Woodstock was essentially anti-establishment," one festival goer, a 30-year-old insurance man from Santa Barbara, CA, opined. "This festival isn't anti-establishment. Hell, we are the establishment."

## Computer books hot, but some make them shoddy


Computer books have been selling as never before and publishers are turning out dozens of new titles every month. Unfortunately, according to *Omni* magazine, in the scramble to publish as many computer books as possible, publishers have rushed out amateurish guides written by computer freaks who know their subject but don't know how to explain it clearly.

The opportunistic tactics that some publishers employ can also contribute to low standards in computer books, Charles Platt wrote in the pop-science magazine.

He added: "Anyone who writes a short article for one of the computer magazines, for instance, will have his name picked up by at least a couple of book publishers, who'll send form letters encouraging him to write a book for them on any computer-related subject, regardless of his qualifications. Indeed, some book publishers prefer to hire amateurs, because they are less trouble to work with. A would-be writer, excited by the prospect of seeing his work in print, won't complain if the publisher makes changes, and he will do the work for lower pay."

But even books done by reputable publishers and professional authors can be disappointing, Platt said. He cited the second edition of *Personal Computing*, by Daniel R. McGlynn, published by John Wiley and Sons of New York. Although authoritative and thorough, Platt noted, readers without a science background will find the book's explanations difficult to understand.

Another book that looks like a winner but is disappointing, he said, is *Without Me You're Nothing*, by Frank Herbert, author of *Dune*. "Herbert claims he's out to debunk and demystify computers," Platt wrote, "but he has a technofreak's love of jargon and the book is full of sweeping generalizations that betray a lack of firsthand experience. It will intimidate

## ADVANCED BAUDOT RADIOTELETYPE

**FOR THE TRS-80 MODELS I and III**

**Disassembled Handbook - Volume 5**

Knowledge of assembler is NOT required
For both BASIC newcomers and advanced assembly language programmers Model 1 3
- Load handle call QTH rig CQ etc from BASIC to disk /CMD ONE TIME ONLY
Includes BASIC load pgm for newcomers
60 66 75 and 100 speed Baudot RTTY
Type ahead in BOTH receive & transmit
21 prepared messages + 5 input messages
Automatic Morse code ID with your call
Store 25K bytes receive message automatically retransmission if desired
- Edit above similar to Pencil
5 Chapters for new BASIC programmers
5 Chapters for advanced assembly programmers
Requires 48K + PORT zero decoder + TU
- No RS 232C interface is required
Newcomers SHOULD order disk & note Model

**VOLUME 5: - $20 ppd — WITH DISK $49 ppd**
**(US funds) add $8.50 overseas airmail**

German - French - Japanese language editions

**RICHCRAFT ENGINEERING LTD.**
**#1 Wahmeda Industrial Park**
**Chautauqua, New York 14722** ✓303
**COD orders—US only (716) 753-2654**

## FALL SPECIALS

| | |
|---|---|
| 48K 2 Drive Mod III | $1579.00 |
| 125K LNW-80 Mod II | $1499.00 |
| C. Itoh Prowriter 8510 Printer | $ 439.00 |
| Sooper Spooler Printer Interface | $ 299.00 |
| Verbatim Disks, SS, DD, 40tk | $ 23.00 |
| Elephant Disks, SS, DD, 40tk | $ 21.00 |
| Daisy Wheel II Cartridge Ribbon | $ 6.50 |
| Mailing Labels (5,000) | $ 15.00 |
| 20# 9.5" x 11" White Tractor Paper | $ 25.00 |

***Hardware for your computer at a price you can afford!***

**EAST COAST DATA**
**4825 Wycliff Road • Portsmouth, VA 23703**
**804/484-6636 (Voice) • 804/484-9636 (Data)**

***Prices and specifications subject to change.***
***MasterCard and Visa welcomed.***

✓513

## NEW SOFTWARE for TRS 80 Model III and the Color Computer

**■ Church Contribution System**
designed to simplify and facilitate the tedious chore of recording envelopes. Provides a variety of reports. Maintains its own data-files Only **$150**

**■ Data Base Manager**
designed to help organize all your data and provide you with meaningful reports. Add or delete any information. New files can be created and old information transferred Only **$150**

**■ Single Entry Ledger**
designed as an uncomplicated control of finances for home or small business. Add, delete, edit at any time. Compatible with DBM Only **$95**

Write or phone for complete software price list

UNIVERSAL DATA RESEARCH INC.
2457 Wehrle Drive
Amherst, NY 14221
716 631-3011
✓417

## "SCOREIT" TAKES THE GREMLINS OUT OF GRADING

A COMPUTER PROGRAM FOR THE RADIO SHACK (TM) TRS-80 MODEL III WHICH SCORES MULTIPLE CHOICE TESTS AND GENERATES:

* Test Statistics
* Score Roster
* Histogram of Scores
* Item Statistics
* Map of Item Responses

UNIQUE FEATURES

* Built-in error detection
* Special diagnostics to guide test revision
* Interface with answer sheet scanners (optional at additional cost)

Include printer model and type when ordering. License for use of the compiled version of the program may be obtained by remitting $195.00 to:

**J. Eli & Associates**
**635 South Hennessey St.**
**New Orleans, LA 70119**

Sample Output Free on Request
Please send SASE ✓491

## SEND AND RECEIVE MORSE CODE

**•SEND UP TO 40 WPM**
**•COPY OVER 100 WPM**

**Your Model I or III becomes a programmable keyboard keyer that can send morse from the keyboard or from stored messages. Keying is done either by the relay or by the audio output. Received morse is decoded, displayed, stored and printed. The cassette earphone plug connects to the receiver speaker to copy off the air. No hardware except patch cords is required for many setups.**

satisfaction guaranteed

**Cassette with instruction booklet only $19.95 postpaid. Model I or III, LEVEL II, with at least 16k required.**

**ROGO Computer Products**
**4752 DeBeers Drive**
**El Paso, Texas 79924** ✓343

## 2 OR MORE PRINTERS TO CONNECT TO YOUR COMPUTER(S)? VICE VERSA?

- STOP PLUGGING AND UNPLUGGING CABLES to change back and forth. It may damage them ... or you may damage your computer
- UNIVERSAL PrinterPicker® switching unit lets you change printers at the flip of a switch. One PrinterPicker® will control 2 printers and
- MULTIPLE PrinterPicker® UNITS MAY BE INTERCONNECTED to form a system to interswitch any number of printers and/or computers
- LOCATE PrinterPicker® WHEREVER IT'S CONVENIENT FOR YOU within the 3 ft. reach of the adaptor cable
- WORKS WITH ALL STANDARD PARALLEL INTERFACED PRINTERS
- WORKS WITH ALL COMPUTERS with appropriate adaptor cable set (we can provide those listed below)
- ALL CONNECTIONS ARE GOLDPLATED for maximum reliability
- VICE VERSA? You bet! PrinterPicker® will also interswitch multiple intermixed or similar computers to one or more printers

ALL ORDERS add $3.50 for shipping and handling

PrinterPicker® Switch Unit.........$65.00
Adaptor Cable Set*.................$25.00

* Each PrinterPicker® Switching Unit requires an Adaptor Cable Set. Specify type: TRS-80[1] Model I, Model II or Model III or EXTENDER (for connection to another PrinterPicker® Switch Unit) or NON TRS 80[1] (for all other computers)

NOTE: ALL computer systems require a TRS-80[1] Model I or III Printer Cable Set for each printer (No 26-1401) our price $25.00

[1]TRS-80 is a trademark of the Tandy Corporation

✓511

**COMPUWARE Corporation**
1008 Abington Road
Cherry Hill NJ 08034
609 428-2309

New Jersey buyers ADD 5%
Terms: CASH, MC or Visa

## PRESERVE 80 micro

## BINDERS & FILE CASES

Keep your issues of 80 Micro together, handy and protected in handsome and durable library files or binders. Both styles bound in dark green leatherette with the magazine logo stamped in gold.

Files: Each file holds 6 issues, spines visible for easy reference. $5.95 each, 3 for $17.00, 6 for $30.00

Binders: Each binder holds 6 issues and opens flat for easy reading. $7.50 each, 3 for $21.75, 6 for $42.00

(Postage paid in USA. Foreign orders include $2.50 per item)

Please state years: 1980 to 1983

Send check or money order to:
JESSE JONES BOX CORP.
P.O. Box 5120
Philadelphia, PA 19141

Please allow 6-8 weeks for delivery

## LARGE CAPACITY MAILIST for MODEL I/III

***Easily holds 3000 names and addresses on just 2 drives. Delete function to automatically find duplicate names. Fast operation***

***4 machine language sub-routines***

***$129***

**RED WING SOFTWARE**
**8809 S. Penn**
**Oklahoma City, OK 73159**
✓498

## NC SOFTWARE INC

Model III

**COMMPAC** is the name
terminal **COMM**unication **PAC**kage is the game

If you would like to be able to use your modem for business or pleasure in a simple and efficient manner without a lot of fuss and three weeks training out of the manual, then read on ... **NC SOFTWARE** has something for you.

The **COMM**unications part is the smart terminal software to compliment the **HAYES SMARTMODEM** in ease of use and overall efficiency. A thorough and easy to use **MENU** enables verification of all program features without being unclear or confusing as to the intent. A quickscan **MINI-MANUAL** is always available from the screen for further reference or clarification. An **AUTO-DIAL / LOGON** feature allows the kind of use the **SMARTMODEM** was meant for. There is also a 20K **CAPTURE BUFFER** and **DOSPLUS** commands and a **FILE EDITOR** feature. **SPECIAL** characters can be generated from the keyboard as well as full control of the transmitting speed rate. A zero **ELAPSED TIME** clock will keep you informed as will the **VIEW STATUS** feature and the quickscan of a **HELP** file. Printer, RS 232 and **ECHO** toggles plus a **WARM RESTART** will keep you in control of all the capabilities. **COMMPAC** is currently running under **TRSDOS**, **NEWDOS-80** and **DOSPLUS** and together with the **SMARTMODEM** makes an unbeatable combination.

The **PAC**kage comes with a **SMARTMODEM** auto-dial and auto-answer modem. This unit not only offers simple phone hookups (pulse dialing or touch tone programming control via **ASCII** strings), audio monitoring, variable dialing parameters and visual signal **LED** indicators but a name, reputation and quality in the industry that is unquestioned.

INTRODUCTORY PRICE $260.00
Also available EDTASM source listing on same disk

Allow 2 extra weeks for personal checks
$2.50/C.O.D. ← **ADD** → $3.50 Shp. Hndl
MINNESOTA RESIDENTS ADD 5%

612 ... 8562 Voice — 612 553 1557 Modem

**NC SOFTWARE INC.**
**7216 Boone Avenue**
**Minneapolis, Minnesota 55428** ✓539

and baffle the reader who does not already own a computer and doesn't know the language."

Platt identified as a useful and well-researched handbook Wilma E. Bennett's *Checklist/Guide to Selecting a Small Computer*, published by Pilot Books of New York. The book contains no fewer than 332 questions to ask a computer salesman before you buy a computer.

And if you need excuses to avoid buying a computer, there's *Computer Choices: Beware of Conspicuous Computing!*, by H. Dominic Covvey and Neil Harding McAlister. Platt said that book debunks the whole computer fad with case histories of small businesses that went bankrupt after computerizing and private users who bought home computers, couldn't find anything useful to do with them, and abandoned them to gather dust in the attic.

## Teacher shortage giving high-tech advantage to Japan, Germany

A worsening math teacher shortage is allowing Germany and Japan to develop the capability to outdistance the United States in the high-technology era, according to Dr. Stephen Willoughby, president of the National Council of Teachers of Mathematics.

Studies by the NCTM have yielded results similar to findings by independent researchers (see *80 Micro*, October 1982, p. 382). They show the math teacher shortage the most severe of any subject taught in the nation's education system. The shortage of science teachers is next.

Since 1972 there has been a 77 percent decline in the number of secondary-level mathematics teachers prepared in 600 teacher-training programs nationwide. And only 55 percent of graduates prepared to teach math go into teaching.

Willoughby, in a report by United Press International, estimated more than half the new math teachers are not qualified to teach.

The situation can be turned around, however, by some proposals currently before Congress, he said. Those bills would—

- Set up tax credits for firms assigning full-time math-competent workers part-time to teach in public schools;
- Give tax credits to firms hiring math and science teachers for the summer; and
- Direct loans at 7 percent interest to students aiming to teach math or science.

## Japanese penetrate disk-drive market

Increased Japanese penetration into the U.S. disk-drive market was reported by the *Electronic News*.

Alps Electronics Ltd. of Tokyo dislodged Shugart Associates as a supplier of 5.25-inch drives to Apple, the industry weekly reported, while TEAC won a 5.25-inch drive contract with the Digital Equipment Corporation and Sony locked up a $30 million contract to supply 3.5-inch drives to Hewlett-Packard.

It was a low bid that won the Apple contract for Alps. The Tokyo firm bid under $50 a drive to supply the mechanism-only devices to Apple, which will insert the controller-boards in the drives.

Despite Alps' and TEAC's victories in the 5.25-inch market, the Japanese may be planning to corner the disk-drive market elsewhere. Grover Mullin, general manager of OEM market requirements for Control Data Corporation of Minneapolis, told *EN*.

Since the Japanese entered the 5.25-inch market late, he contended, they might concentrate on putting higher density on smaller floppies.

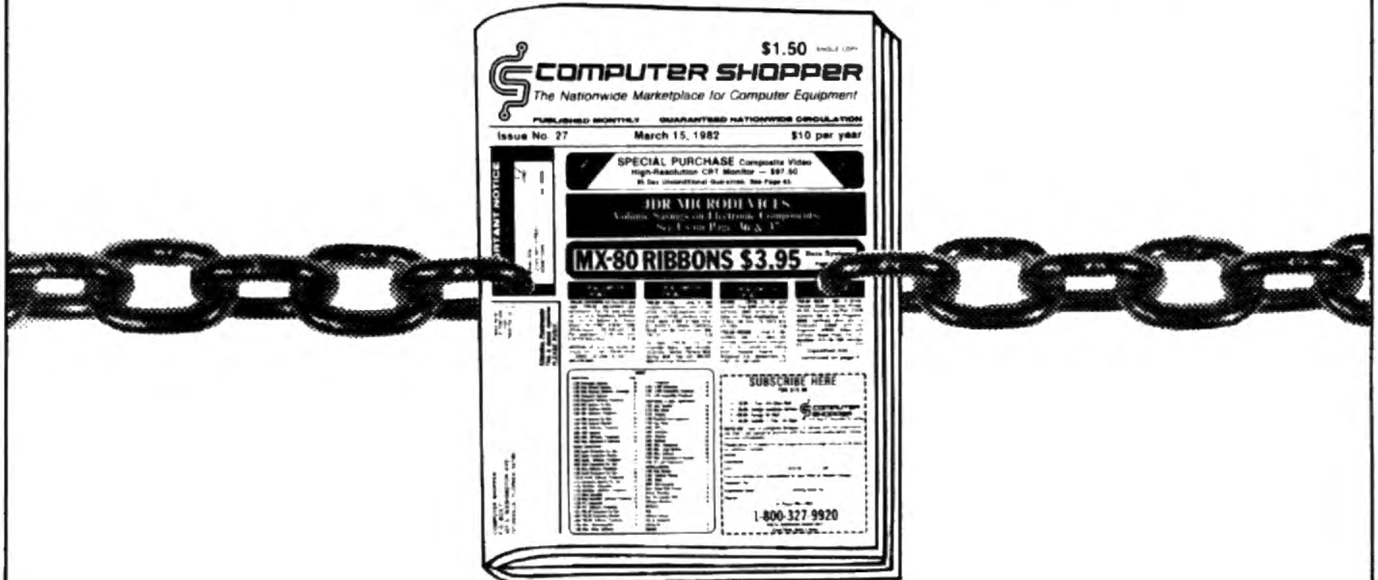
Sony is the only company to deliver to the U.S. market a microfloppy drive—3.5 inches, single-sided, double-density—although Hitachi and Matsushita are ready to bring their 3-inch drives to market. Sony is expected to release a double-sided microdrive next year.

In additon to increased activity in the microfloppy market, 5.25-inch and 8-inch manufacturers are finding a demand for slimline or half-height drives. Some 1500 8-inch half-height drives a month are being shipped by NEC, *EN* reported.

Currently, NEC and Tandon are the only firms producing the 8-inch half-heights, but MPI, Shugart, Qume, and Mitsubishi have announced they will be producing them, lowering the price of the drives to $370 a

# YOU'VE JUST FOUND

# THE MISSING LINK!




Computer Shopper is *your link* to individuals who buy, sell and trade computer equipment and software among themselves nationwide. No other magazine fills this void in the marketplace chain.

Thousands of cost-conscious computer enthusiasts save by shopping in Computer Shopper every month through hundreds of classified ads. And new equipment advertisers offer some of the lowest prices in the nation.

Computer Shopper's unbiased articles make for some unique reading among magazines and there's a "help" column to answer difficult problems you may have with interfacing, etc.

For a *limited time* you can subscribe to Computer Shopper with a 6 month trial for $6 or 12 months for $10.

You risk nothing, because if you're not 100% satisfied after receiving the first issue, you may cancel and receive a full refund. No questions asked!

If you want faster service, call in your subscription and charge on your MasterCard or VISA account.

Our business office is open from 8 a.m. to 5 p.m. Monday through Friday. Call for charge card orders or send your check or money order TODAY while the offer is still good.

---

**6 month trial, $6 or 12 months for $10**

---

**COMPUTER SHOPPER**

P.O. Box F-324 • Titusville, FL 32780
305-269-3211

# PULSETRAIN

drive for volume shipments.

One NEC official told *EN*: "We're seeing a lot of interest in the product. The fact that it has reduced power requirements and can fit in a smaller space makes it attractive to customers."

## Correction

In the October issue of *80 Micro* the number for Bob Rosen's BBS was published incorrectly. The correct number is (212) 441-3755.

## Anti-video game forces paranoid

Computer games aren't creating a generation of computer zombies, a child psychologist and computer expert said in a report appearing in the *Providence Journal*.

Mitchell Robin, a professor of psychology at the New York City Technical College, maintained parents are "paranoid" about computers, because they grew up during a period when the machines were portrayed as Ghengis Khans in waiting—just biding their time until the malevolent machines could take over the world and force humans to watch reruns of *I Love Lucy*.

In the story appearing in the Rhode Island daily, Robin said his two daughters, aged 8 and 5, are more proficient in killing off multi-colored video monsters than their father, something he doesn't mind.

"When they win," he said, "I'm delighted they can see dad as being fallable. They learn that I'm human with limitations and that I'm not always successful in what I'm doing. I'm not superhuman."

Robin contended parents could learn a great deal about themselves and how their children see them competing with them in video game play:

"I find that parents who must always be right and never lose anything are parents whose children distrust them. Parents are humans and we tend to get very easily seduced into being perfect" in the eyes of children.

Robin debunked a common fear among anti-video game elements that the machines' warlike, violent themes will have a bad effect on children: "I've worked in programs with children as young as 8 years old. I see their humanity increasing, not decreasing."

Fink: Driving force behind VIC.

## Commodore's Fink moves to ad agency

Alan W. Fink, the driving force behind the spectacular rise of the VIC-20 microcomputer, has left Commodore and joined Kornhauser & Calene Inc., a New York advertising agency.

The agency, which worked closely with Commodore in marketing the VIC-20, hired Fink to head its new consumer electronics marketing group.

According to a statement from Kornhauser, under Fink's guidance, the VIC-20 took only six months to capture a significant share of the personal computer market and made a significant contribution to Commodore's 63 percent increase in fiscal 1982 sales.

"The category of small personal computers is so new that there are few, if any, organizations that can provide the expert marketing assistance needed to sell successfully," Board Chairman Henry Kornhauser said. "Alan Fink provides the vital link in the marketing chain by bringing to clients the manufacturer's viewpoint."

## Today's engineers and technicians obsolete in 5 years

Today's design engineers and technicians may be technologically obsolete in three to five years, according to Douglas Bonham, director of Heath/Zenith Educational Division of the Heath Company, Benton Harbor, MI.

Bonham told the *T.H.E. Journal* technology is changing so fast in the electronics and computer fields that engineers face obsolescence as quickly as the products they design.

In addition, electronics is invading virtually every field where mechanical and electrical equipment used to dominate, forcing many engineers to learn entirely new disciplines. ■

# FUN HOUSE

By Richard Ramella

# Season's Greetings

As kids and kids in spirit enter the Fun House this month, you can see that I've decorated the place for the holiday season.

Christmas falls as usual on December 25, and Hanukkah starts on December 11 this year. Though the two holidays celebrate different events, there are similarities. Both include decorations, giving gifts, and playing games, somewhat apart from their religious meanings.

So let's have a Fun House party with these programs:

The 12 Days of Christmas is a Color Computer program that celebrates that peculiar story of gift-giving gone wild.

Dreidel is a Level II listing for an ancient Hanukkah game of a spinning top and shifting fortunes. It's pronounced *draydul*.

Peglegs is my holiday gift to you—a simple, but tricky computer version of a game you may recognize. There are listings for Level II and the Color Computer.

This month's Nickel Bargain Bin is the solution to the

*Twelve Days of Christmas*

```
100 REM * 12 DAYS OF CHRISTMAS / TRS-80 COLOR BASIC 4K *
110 CLS(0)
120 DIM A$(24)
130 DATA A PARTRIDGE IN A PEAR TREE., TWO TURTLE DOVES
140 DATA THREE FRENCH HENS, FOUR CALLING BIRDS
150 DATA FIVE GOLDEN RINGS, SIX GEESE A-LAYING
160 DATA SEVEN SWANS A-SWIMMING, EIGHT MAIDS A-MILKING
170 DATA NINE PIPERS PIPING, TEN DRUMMERS DRUMMING
180 DATA ELEVEN LORDS A-LEAPING, TWELVE LADIES DANCING
190 DATA 12TH,11TH,10TH,9TH,8TH
200 DATA 7TH,6TH,5TH,4TH,3RD,2ND,1ST
210 Z$="204 204 185 193 197 185"
220 Y$="197 204 204 210 197 193 176 185 185 176 176 176 176"
230 X$="204 204 204 210 210 210 200 200 200 204 204 204"
240 W$="147 147 147 147 176 176 176 176 170"
250 V$="176 185 193 197 185 193 193 193"
260 U$="204 197 193 185 176"
270 T$="197 197 159 170 176"
280 S$="185 176 170 159 147"
290 N=1
300 FOR A=1 TO 24
310 READ A$(A)
320 NEXT A
330 X=31
340 Z=0
350 FOR Y=0 TO 15
360 FOR L=X-Z TO X+Z
370 K=RND(20)
380 IF K<6 THEN SET(L,Y,RND(8)) ELSE SET(L,Y,6)
390 NEXT L
400 Z=Z+1
410 NEXT Y
420 PRINT @ 322,"THE TWELVE DAYS OF CHRISTMAS ";
430 IF G=50 GOTO 430
440 FOR T=1 TO 1500
450 NEXT T
460 CLS(0)
470 FOR A=12 TO 1 STEP -1
480 PRINT
490 PRINT
500 PRINT "ON THE "A$(A+12)" DAY OF CHRISTMAS "
510 A$=W$
520 GOSUB 790
530 PRINT "MY TRUE LOVE GAVE TO ME:"
540 A$=V$
550 GOSUB 790
560 FOR B=N TO 1 STEP -1
570 IF B=1 OR B=2 THEN B$=" " ELSE B$=", "
580 IF N>1 AND B=1 THEN PRINT "AND ";
590 PRINT A$(B);
600 IF B=1 THEN PRINT B$; ELSE PRINT B$
610 IF B>5 THEN A$=Z$
620 IF B=5 THEN A$=X$
630 IF B>1 AND B<5 AND N<5 THEN A$=Z$
640 IF B=4 AND N>4 THEN A$=U$
650 IF B=3 AND N>4 THEN A$=T$
660 IF B=2 AND N>4 THEN A$=S$
670 IF B=1 THEN A$=Y$
680 GOSUB 790
690 U=U+B
700 NEXT B
710 PRINT @ 0,"TOTAL GIFTS:";U
720 N=N+1
730 FOR X=1 TO 500
740 NEXT X
750 CLS(0)
760 NEXT A
770 G=50
780 GOTO 330
790 IF N>1 AND B=1 THEN SOUND 193,3
800 FOR J=1 TO LEN(A$) STEP 4
810 B$=MID$(A$,J,3)
820 P=VAL(B$)
830 SOUND P,3
840 NEXT J
850 RETURN
860 END
```


Help Protect the Planet From A Wide Variety of Cosmic Crafts That Are Really . . . .
WEERD!
By Arthur Gleckler
In your quest for survival, battle your way through dozens of different types of strange-looking space vehicles. Watch out! Quick! Put up your shields before they annihilate you!
BIG FIVE
SOFTWARE
P.O. Box 9078-185 · Van Nuys, CA 91409 · (213) 782-6861
"Weerd" is now available from your local dealer. If your dealer does not have it then send us a check or money order for $19.95 plus $2.00 for shipping & handling. Outside USA (except Canada) add $6.00 for shipping. California residents add 6½% sales tax. Be sure to specify cassette or diskette.
"Weerd" is 100% machine language and includes sound effects playable through the cassette port. Cassette version requires 16K Model 1 or Model 3 TRS-80™. Disk version requires 32K Model 1 or Model 3 TRS-80™. Disk version features voice sound effects and permanent high scores. Both versions are compatible with the TRISSTICK™.
TRS-80 is a T.M. of Tandy Corp.

Peglegs game.

So, fa-la-la-la-la, and let's get started before the hot cider gets cold.

### The 12 Days of Christmas

If a wild gift-giver duplicated the events of this Christmas song, someone would wind up with 364 items by the 12th day. I found that out when I built a running total into this program. It displays the entire song with the melody if you want to sing along. That's all the program does, but that's a lot, considering the length of the listing.

### Dreidel

Hanukkah is a Jewish holiday that celebrates the long-ago rededication of the Temple of Jerusalem. And Dreidel is a game that children have been playing on Hanukkah for hundreds of years.

This program is given only in Level II; it simulates a game of Dreidel with the computer playing against a human.

A dreidel is a top with four sides. On each side is a letter from the Hebrew alphabet. These letters are shown with this article (see Fig. 1).

Each player starts with the same amount of Hanukkah gelt—gelt means money—or you can use counters ranging from peanuts to marbles.

To start, each player puts five pieces of Hanukkah gelt in the main pile. Each player spins the dreidel. These are results of the letters that are face up when the top stops spinning:

- NUN—Nothing happens.
- GIMMEL—Take all the pot.
- HE—Take half the pot.
- SHIN—Give half of what you have to the pot.

When the pot is empty, each player must put five counters in it. The loser is the player who cannot put five counters into the empty pot.

This program includes a spinning dreidel and enough automatic features to make it easy to play.

### Peglegs

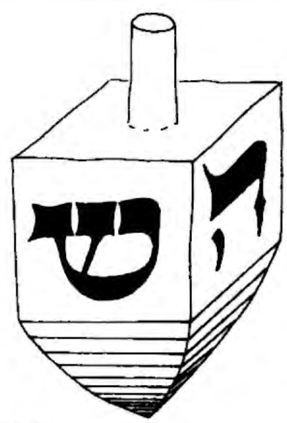
And now it's time to open your gift.

Peglegs is a computer version of a game that has been around for a few hundred years. I offer program listings for Level II and Color Basic.

To start, a cross-shaped board draws on the screen and fills with 32 pegs. The space in the center of the board is empty.

Your goal is to remove as many pegs as you can from the board by jumping pegs.

*Dreidel*



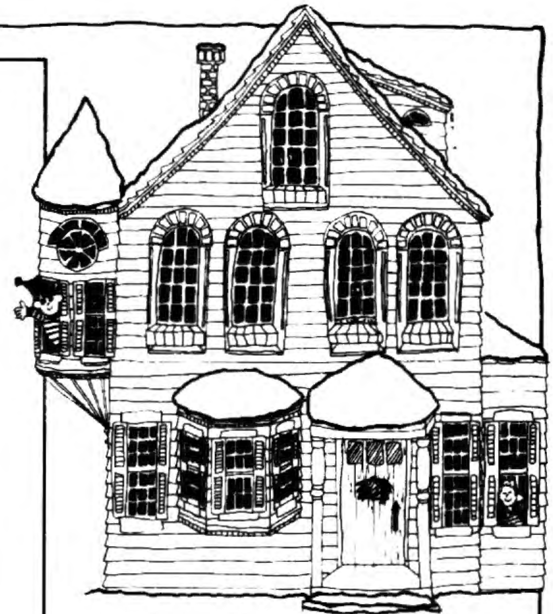
```
100 REM *DREIDEL / TRS 80 LEVEL II *
110 CLS
120 CLEAR 150
130 S$=CHR$(128)
140 T$=CHR$(131)
150 U$=CHR$(171)
160 V$=CHR$(179)
170 W$=CHR$(181)
180 X$=CHR$(186)
190 Y$=CHR$(187)
200 Z$=CHR$(191)
210 P=582
220 A=596
230 M=608
240 PRINT "LET'S PLAY DREIDEL."
250 PRINT
260 PRINT "THE LIMIT IS 100..."
270 INPUT "HOW MUCH HANUKKAH GELT FOR EACH";B
280 IF B>100 THEN PRINT: PRINT: GOTO 260
290 C=B
300 G=B
310 D=0
320 CLS
330 E$=S$+Z$+STRING$(5,T$)+Z$+S$
340 A$(1)=S$+Z$+S$+S$+V$+U$+S$+Z$+S$
350 A$(2)=S$+Z$+S$+S$+V$+Y$+S$+Z$+S$
360 A$(3)=S$+Z$+S$+Z$+T$+Z$+S$+Z$+S$
370 A$(4)=S$+Z$+S$+W$+Z$+X$+S$+Z$+S$
380 B$=S$+Z$+STRING$(5,S$)+Z$+S$
390 C$=S$+STRING$(7,Z$)+S$
400 D$=S$+S$+STRING$(5,Z$)+S$
410 F$=S$+S$+S$+S$+Z$+S$
420 Y=21
430 FOR X=0 TO 126
440 SET(X,Y)
450 SET(X,Y+10)
460 NEXT X
470 PRINT @ 514,"YOU";
480 PRINT @ 528,"POT";
490 PRINT @ 542,"ME";
500 X=0
510 FOR Y=21 TO 31
520 SET(X,Y)
530 SET(X+27,Y)
540 SET(X+54,Y)
550 Z=0
560 SET(X+78,Y)
570 SET(X+126,Y)
580 NEXT Y
590 PRINT @ P,C;
600 PRINT @ A,D;
610 PRINT @ M,G;
620 IF C<1 OR G<1 GOTO 660
630 IF D=0 GOTO 1060
640 Q=Q+1
650 PRINT @ 768,"";
660 IF C<1 GOSUB 1130: PRINT @ 768,"I WIN.": GOTO 1180
670 IF G<1 GOSUB 1130: PRINT @ 768,"YOU WIN.": GOTO 1180
680 IF Q=1 THEN PRINT "YOUR TURN. ";
690 IF Q=2 THEN PRINT "MY TURN. ";: FOR T=1 TO 500: NEXT T: GOTO
 740
700 PRINT "TAP ENTER TO SPIN THE DREIDEL."
710 K$=INKEY$
720 IF K$<>"" GOTO 740
730 GOTO 710
```

*Dreidel Continues*


*Figure 1*

*Dreidel Continued*

```
740 GOSUB 1150
750 FOR X=1 TO RND(3)
760 IF X/2<>INT(X/2) THEN FOR Z=Z TO RND(45)-Z+10
770 IF X/2=INT(X/2) THEN FOR Z=Z TO RND(10) STEP -1
780 U=RND(4)
790 PRINT @ Z,E$;
800 PRINT @ Z+64,A$(U);
810 PRINT @ Z+128,B$;
820 PRINT @ Z+192,C$;
830 PRINT @ Z+256,D$;
840 PRINT @ Z+320,F$;
850 NEXT Z
860 NEXT X
870 PRINT @ 624,"";
880 IF U=1 THEN PRINT "NUN";
890 IF U=2 THEN PRINT "GIMMEL";
900 IF U=3 THEN PRINT "HE";
910 IF U=4 THEN PRINT "SHIN";
920 PRINT @ 768,"";
930 IF U=1 THEN PRINT "NOTHING."
940 IF U=2 THEN PRINT "TAKE ALL."
950 IF U=3 THEN PRINT "TAKE HALF."
960 IF U=4 THEN PRINT "GIVE HALF."
970 IF U=2 AND Q=1 THEN C=C+D: D=0
980 IF U=2 AND Q=2 THEN G=G+D: D=0
990 IF U=3 AND Q=1 THEN I=INT(D/2): C=C+I: D=D-I
1000 IF U=3 AND Q=2 THEN I=INT(D/2): G=G+I: D=D-I
1010 IF U=4 AND Q=1 THEN I=INT(C/2): C=C-I: D=D+I
1020 IF U=4 AND Q=2 THEN I=INT(G/2): G=G-I: D=D+I
1030 GOSUB 1130
1040 IF Q=2 THEN Q=0
1050 GOTO 590
1060 PRINT @ 768,"";
1070 PRINT "WHEN POT IS EMPTY, WE EACH PUT IN 5."
1080 C=C-5
1090 G=G-5
1100 D=D+10
1110 GOSUB 1130
1120 GOTO 590
1130 FOR T=1 TO 500
1140 NEXT T
1150 PRINT @ 768,STRING$(43," ");
1160 PRINT @ 624,STRING$(10," ");
1170 RETURN
1180 GOTO 1180
1190 END
```



You're a winner if you have two pegs left and you're a double winner if you have only one peg left. You're a triple winner if only one peg remains and its position is the space in the middle of the board.

Now, how do you move around and jump? The Color Computer and Level II versions are slightly different, but this should make both of them clear: In both versions, you move into jumping position by tapping D for down, U for up, L for left, and R for right.

In the color version, your position is marked in blue if you are on a space not occupied by a peg. If you are on one of the occupied spaces, the position will blink on and off.

In the Level II version, your

*Peglegs*

```
100 REM * PEGLEGS / TRS-80 LEVEL II 4K *
110 CLS
120 Q=32
130 U$="JU"
140 D$="JD"
150 L$="JL"
160 R$="JR"
170 Y=3
180 FOR X=18 TO 36
190 SET(X,Y)
200 SET(X,Y+22)
210 NEXT X
220 X=6
230 FOR Y=9 TO 19
240 SET(X,Y)
250 SET(X+42,Y)
260 NEXT Y
270 X=18
280 FOR Y=3 TO 24
290 IF Y<9 OR Y>18 THEN SET(X,Y): SET(X+18,Y)
300 NEXT Y
310 Y=9
320 FOR X=6 TO 48
330 IF X<19 OR X>35 THEN SET(X,Y): SET(X,Y+10)
340 NEXT X
350 FOR Y=5 TO 24 STEP 3
360 FOR X=21 TO 36 STEP 6
370 SET(X,Y)
380 NEXT X
390 NEXT Y
400 FOR Y=11 TO 17 STEP 3
410 FOR X=9 TO 45 STEP 6
420 SET(X,Y)
430 NEXT X
440 NEXT Y
450 RESET(27,14)
460 A=27
470 B=5
480 A$=A$+INKEY$
485 PRINT @ 653,A$;"  ";
490 PRINT @ 480,"SCORE";Q;
```

*Peglegs Continues*

# INFLO INC.

*Presents*

## 2 MEGABYTES DISK STORAGE
## $1,000.00

| | |
|---|---|
| INFLO 2 MEG SYSTEM FOR TRS-80, LNW, or PMC 80 | $1000.00 |

## THE INFLO 3.2 MEGABYTE HIGH SPEED SYSTEM FOR:

| | |
|---|---|
| TRS 80 MOD I or III or PMC 80 | $1550.00 |
| Late LNW 80 or LNW Model II | $1300.00 |
| NEW PRICING LNW Model II | $1995.00 |

## FULL LINE OF LNW PRODUCTS AND SOFTWARE

| | |
|---|---|
| CHARM | $39.95 |
| AUTOPLOT/LNW | $99.50 |
| LNWBASIC | $69.95 |
| HIRES 1&2 | $20.00 |

CALL YOUR LOCAL DEALER

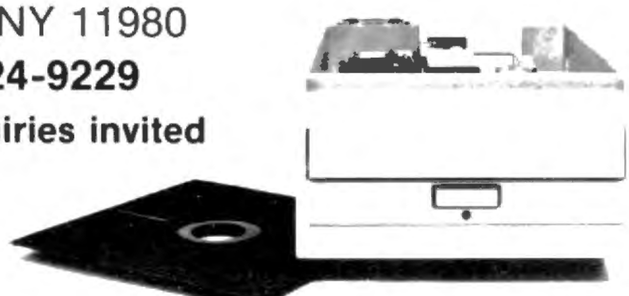
| | | |
|---|---|---|
| LA ...... (504) 293-3400 | MA ..... (413) 583-6383 | IL ....... (312) 782-5300 |
| NJ ...... (201) 751-8444 | DC ..... (703) 527-9060 | WI ...... (414) 261-1780 |
| FL ...... (305) 443-7749 | NY ...... (516) 924-9229 | OR ..... (503) 232-5552 |

THE LNW USERS GROUP BI-MONTHLY NEWSLETTER IS NOW IN CIRCULATION;
FOR MORE INFO ON THIS GROUP CALL THE LNW BBS **(516) 924-8115**

# Inflo Inc.

244 Mill Rd.
Yaphank, NY 11980
**(516) 924-9229**

**Dealers inquiries invited**



~530

LNW is a trademark of
LNW Research Corp

TRS-80 is a trademark of
Tandy Corp

BBS Number is for Modem use only

See List of Advertisers on Page 451

*Peglegs Continued*

```
500 IF Q=2 THEN PRINT @ 154,"WINNER";
510 IF Q=1 THEN PRINT @ 90,"DOUBLE";
520 IF Q=1 AND A=27 AND B=14 THEN PRINT @ 90,"TRIPLE";
530 IF POINT(A,B)=-1 THEN RESET(A,B): FOR T=1 TO 10: NEXT T: SET
(A,B)
540 IF POINT(A,B)=0 THEN SET(A,B):FOR T=1 TO 10: NEXT T: RESET(A
,B)
550 IF A$="U" AND POINT(A,B-2)<>-1 THEN B=B-3: A$=""
560 IF A$="D" AND POINT(A,B+2)<>-1 THEN B=B+3: A$=""
570 IF A$="L" AND POINT(A-3,B)<>-1 THEN A=A-6: A$=""
580 IF A$="R" AND POINT(A+3,B)<>-1 THEN A=A+6: A$=""
590 IF B-5<4 GOTO 610
600 IF A$=U$ AND POINT(A,B-5)=0 AND POINT(A,B-6)=0 AND POINT(A,B
-3)=-1 GOTO 690
610 IF B+5>24 GOTO 630
620 IF A$=D$ AND POINT(A,B+5)=0 AND POINT(A,B+6)=0 AND POINT(A,B
+3)=-1 GOTO 690
630 IF A-9<6 GOTO 650
```

position simply blinks, wherever you are.

To jump a peg in the color version, just press one of the four direction arrows on the keyboard. To jump a peg in the Level II version, tap JD for jump down, JU for jump up, JL for jump left, and JR for jump right.

You cannot make any illegal moves, and there are other safeguards built in. You don't have to press enter with this program; the arrow keys do the job.

If you don't understand this yet, run the program and you soon will.

If you can't win, take a look at the Nickel Bargain Bin offer at the end of the column.

Peglegs is tough to beat, but I have a strategy and it's yours for a nickel and a self addressed, stamped envelope. If you want the answer, send the request with a note bearing the words "PEGLEGS SOLUTION" to me, Richard Ramella, 1493 Mountain View Ave., Chico, CA 95926. *Do not* send the request to this magazine; I am responsible for the offer.

Next month is January, isn't it? No better time than the first of the year to think about time—calendars, timepieces, leap years, and such. So, until 1983 . . . ■

*Color Peglegs*

```
100 REM * PEG LEGS / TRS-80 COLOR BASIC 4K *
110 CLS(0)
120 Q=32
130 U$=CHR$(94)
140 D$=CHR$(10)
150 L$=CHR$(8)
160 R$=CHR$(9)
170 Y=3
180 FOR X=18 TO 36
190 SET(X,Y,5)
200 SET(X,Y+22,5)
210 NEXT X
220 X=6
230 FOR Y=9 TO 19
240 SET(X,Y,5)
250 SET(X+42,Y,5)
260 NEXT Y
270 X=18
280 FOR Y=3 TO 24
290 IF Y<9 OR Y>18 THEN SET(X,Y,5): SET(X+18,Y,5)
300 NEXT Y
310 Y=9
320 FOR X=6 TO 48
330 IF X<19 OR X>35 THEN SET(X,Y,5): SET(X,Y+10,5)
340 NEXT X
350 FOR Y=5 TO 24 STEP 3
360 FOR X=21 TO 36 STEP 6
370 SET(X,Y,8)
380 NEXT X
390 NEXT Y
400 FOR Y=11 TO 17 STEP 3
410 FOR X=9 TO 45 STEP 6
420 SET(X,Y,8)
430 NEXT X
440 NEXT Y
450 RESET(27,14)
460 A=27
470 B=5
480 A$=INKEY$
490 PRINT @ 502,"SCORE";Q;
500 IF Q=2 THEN PRINT @ 32,"WINNER";
510 IF Q=1 THEN PRINT @ 0,"DOUBLE";
520 IF Q=1 AND A=27 AND B=14 THEN PRINT @ 0,"TRIPLE";
530 IF POINT(A,B)=8 THEN RESET(A,B):FOR T=1 TO 10: NEXT T: SET(A
,B,8)
540 IF POINT(A,B)<>8 THEN SET(A,B,7): FOR T=1 TO 40: NEXT T: RES
ET(A,B)
550 IF A$="U" AND POINT(A,B-2)<>5 THEN B=B-3
560 IF A$="D" AND POINT(A,B+2)<>5 THEN B=B+3
570 IF A$="L" AND POINT(A-3,B)<>5 THEN A=A-6
580 IF A$="R" AND POINT(A+3,B)<>5 THEN A=A+6
590 IF B-5<4 GOTO 610
600 IF A$=U$ AND POINT(A,B-5)<>5 AND POINT(A,B-6)<>8 AND POINT(A
,B-3)=8 GOTO 690
610 IF B+5>24 GOTO 630
620 IF A$=D$ AND POINT(A,B+5)<>5 AND POINT(A,B+6)<>8 AND POINT(A
,B+3)=8 GOTO 690
630 IF A-9<6 GOTO 650
640 IF A$=L$ AND POINT(A-9,B)=0 AND POINT(A-12,B)=0 AND POINT(A-
6,B)=-1 GOTO 690
650 IF A+9>49 GOTO 665
660 IF A$=R$ AND POINT(A+9,B)=0 AND POINT(A+12,B)=0 AND POINT(A+
6,B)=-1 GOTO 690
665 IF LEN(A$)>1 THEN A$=""
670 GOTO 480
690 RESET(A,B)
700 IF A$=U$ THEN RESET(A,B-3): B=B-6
710 IF A$=D$ THEN RESET(A,B+3): B=B+6
720 IF A$=L$ THEN RESET(A-6,B): A=A-12
730 IF A$=R$ THEN RESET(A+6,B): A=A+12
740 SET(A,B)
745 A$=""
750 Q=Q-1
760 GOTO480
770 END
```

from HOWE SOFTWARE

# FOR THE TRS-80 MODEL I AND MODEL III

# System Diagnostic

**For Cassette or Disk**

## IS YOUR COMPUTER WORKING CORRECTLY? ARE YOU SURE?

**Tests every component of your TRS-80 for proper operation.**

**DISK $99.95**

**CASSETTE $94.95**

NEW

ROM: Checksum test
RAM: Four separate tests including every address and data value
Video Display: Character generator, video RAM, and video signal tests
Keyboard: Every key contact tested
Line Printer: Character test
Cassette Recorder: Read/write/verify data
RS-232-C Interface: Read sense switches (Model I), connector fault, data transmission, framing, data loop, baud rate generator
Disk Drives: Disk controller, drive select and restore, track seek and verify read, read/write/verify all tracks and sectors, formatting, disk drive timer, disk head cleaner
Model I: single or double density, 35, 40 or 80 track drives
Model 3: single or double density, 35, 40 or 80 track drives
single or double sided drives

--- One program adapts to any system configuration and hardware.
--- Individual tests of each device with operator monitoring and intervention.
--- Continuous system tests run continually for hours, with diagnostic reports optionally written on line printer.
--- Complete instructions and documentation.

**SPECIFY TRS-80™ MODEL I OR MODEL III**

## SMART TERMINAL

Enables your TRS-80 to be used as a data communications terminal to a time-sharing system, computer bulletin board, or another computer, via the RS-232-C interface.

- MEMORY BUFFER holds data for transmission or data received from other computer.
- CASSETTE or DISK may be used to load or save data from memory.
- AUTOMATIC TRANSMISSION of data from memory.
- AUTOMATIC STORAGE of incoming data at user's option.
- TRANSMIT or RECEIVE WITH VERIFICATION options included for communication between two TRS-80s using Smart Terminal.
- Full CONTROL KEYS, including control key mapping into any ASCII character. True BREAK key. Lower case supported on Model I.
- Buffered LINE PRINTER ECHO for incoming data.
- Disk and cassette files fully compatible with ELECTRIC PENCIL™ and SCRIPSIT™ programs.
- BAUD RATE and RS-232-C CHARACTERISTICS can be reset from within the program.
- SAVE PROGRAM option creates "personalized" back-up copy of program with all options set as specified by user.
- ONE PROGRAM supports both cassette and disk systems. Program is compatible with PMC-80 and other TRS-80 "work alike" computers.

**Model I or III Version** **supplied on cassette $69.75**
**supplied on diskette $74.95**
**Model II (CP/M) Version** **$79.95**

## MON-3 and MON-4

The TRS-80 Monitor Programs #3 and #4 are powerful utility programs enabling you to interact directly with the TRS-80 in Machine Language. They are as useful for beginners as for advanced programmers.

- BEGINNERS can learn to interact directly with the computer in Machine Language.
- 40-PAGE MANUAL provided with each program.
- SIMPLE commands, easy to use.

The Features Of The Monitor Programs Enable You To The Following.....

- DISPLAY memory in different ways.
- DISASSEMBLE memory to see Machine Language commands.
- MOVE and COMPARE memory areas.
- SEARCH through memory to find specific values.
- MODIFY memory in various ways.
- RELOCATE object programs.
- PRINT output on video display or line printer.
- READ and WRITE object tapes in SYSTEM Format.
- UNLOAD programs using low RAM on disk.
- SAVE and READ disk files (MON-4 Only).
- INPUT and OUTPUT of disk sectors (MON-4 Only).
- SEND and RECEIVE data over RS-232-C Interface (MON-4 Only).
- Create SYMBOLIC Tapes (MON-3) or Files (MON-4) of Disassembled output for Editor/Assembler program.

**MON-3 (For Cassette Systems) $39.95**
**MON-4 (For Disk Systems) $49.95**

## MAILING LIST

Maintains mailing lists of up to 1326 names (48K version). Add, change, delete, or find names. Machine language sort according to information in ANY field (first or last name, address, city, state, zip code). Three or four line labels printed in 1, 2, 3, or 4 columns, in master list, or on video display.

**TRS-80 Model 1/3 Disk Version $69.95**
**IBM PC Disk Version $79.95**

## HOME BUDGET

Keeps track of your monthly and year-to-date income and expenses. Income and expenses classified by code numbers for identification of categories. Data includes date, code number, amounts and check number (optional). Computes monthly and year-to-date summaries showing income tax deductions. All output printed on video display or line printer at user's option. Complete instructions for customizing to suit your own budget.

**TRS-80 Model 1/3 Cassette Version $29.95**
**TRS-80 Model 1/3 Disk Version $49.95**
**IMB PC Disk Version $59.95**

## SMALL BUSINESS ACCOUNTING

Based on Dome Bookkeeping Record #612, this program keeps track of income, expenditures, and payroll for a small business. Receipts and expenditures can be entered on a daily, weekly, or monthly basis. Program computes monthly, through last month, and year to date summaries. Payroll section (included in disk version only) keeps record of employees and paychecks with up to six categories of payroll deductions. Computes employee payroll records and year-to-date payroll totals. Complete instructions for customizing to suit your own business.

**TRS-80 Model 1/3 Cassette Version $29.95**
**TRS-80 Model 1/3 Disk Version $49.95**
**IBM PC Disk Version $59.95**

COMPUTRONICS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

VISA AMERICAN EXPRESS master charge

**24 HOUR ORDER LINE**
**(914) 425-1535**

**NEW TOLL-FREE ORDER LINE**
(OUTSIDE OF N.Y. STATE)
**(800) 431-2818**

- All orders processed within 24 Hours
- 30-Day money back guarantee
- Add $3.00 for shipping in UPS Areas
- Add $4.00 for C.O.D. or NON-UPS Areas
- Add $5.00 to Canada or Mexico
- Add exact postage to all other countries

## HSDS HARD DISK DRIVE SOFTWARE

- MAKES TRSDOS COMPATIBLE WITH MOST HARD DISK DRIVES
- ADDS MANY EXTRA FEATURES TO TRSDOS ... $400.00

## COMPROC (COMMAND PROCESSOR)

- AUTO YOUR DISK TO PERFORM ANY SEQUENCE OF INSTRUCTIONS THAT YOU NORMALLY GIVE FROM THE KEYBOARD (FOR EXAMPLE: INSERT THE DISKETTE, PRESS THE RESET BUTTON, YOUR COMMAND FILE COULD AUTOMATICALLY SHOW YOU THE DIRECTORY, SHOW THE FREE SPACE ON THE DIKSETTE, LOAD A MACHINE LANGUAGE SUBROUTINE, LOAD BASIC, LOAD AND RUN A BASIC PROGRAM AND SELECT A GIVEN ITEM ON YOUR MENU. ALL WITHOUT TOUCHING THE KEYBOARD!)

MODEL-I VERSION ... $20.00
MODEL-III VERSION ... $30.00
**NOT AVAILABLE FOR MODEL-II**

## DISCAT (DISKETTE CATALOG SYSTEM)

- THIS COMPREHENSIVE DISKETTE CATALOGUING/INDEXING UTILITY ALLOWS THE USER TO KEEP TRACK OF THOUSANDS OF PROGRAMS IN A CATEGORIZED LIBRARY FILE INCLUDES PROGRAM NAMES AND EXTENSIONS, PROGRAM LENGTH, DISKETTE NUMBERS AND FREE SPACE ON EACH DISKETTE. KEEP A COMPLETE CATALOG OF THE DIRECTORIES ON ALL YOUR DISKETTES IN ALPHABETICAL ORDER (SORTED ON EACH DISKETTE) OR COMPLETE ALPHABETICAL LIST OF PROGRAMS ON ALL YOUR DISKETTES)

MODEL-I VERSION ... $50.00
MODEL-III VERSION ... $50.00
MODEL-II VERSION (SEE MODEL-II UTILITY PACKAGE)

## BLINK (BASIC LINK FACILITY)

- LINK FROM **BASIC** PROGRAM TO ANOTHER SAVING ALL VARIABLES
- THE CHAINED PROGRAM MAY EITHER REPLACE THE ORIGINAL PROGRAM OR CAN BE MERGED BY STATEMENT NUMBER

MODEL-I VERSION ... $25.00
MODEL III VERSION ... $30.00
MODEL-II VERSION (SEE MODEL-II UTILITY PACKAGE) ... $50.00

## INFINITE BASIC

- ADDS OVER 80 COMMANDS TO BASIC
- SORTING, STRING CENTERING, ROTATION, TRUNCATION, JUSTIFICATION, DATA COMPRESSION, STRING TRANSLATION, COPYING SCREEN DISPLAY, SCROLLING, MATRIX OPERATIONS, SIMULTANEOUS EQUATIONS (THROUGH MATRIX INVERSION), DYNAMIC ARRAY RESHAPING

MODEL-I VERSION ... $50.00
MODEL-III VERSION ... $60.00
**NOT AVAILABLE ON MODEL-II**

## INFINITE BUSINESS

- ADD ON PACKAGE TO **INFINITE BASIC** (REQUIRES INFINITE BASIC)
- ADDS PACKED DECIMAL ARITHMETIC WITH 127 DIGIT ACCURACY (+ 0 ...)
- COMPLETE PRINTER PAGINATION CONTROLS, AUTO HEADERS, FOOTERS AND PAGE NUMBERS
- BINARY SEARCH OF SORTED AND UNSORTED ARRAYS (INSTANT SEARCH OF AN ELEMENT WITHIN AN ARRAY)
- HASH CODES

MODEL-I VERSION ... $30.00
MODEL-III VERSION ... $30.00
**NOT AVAILABLE ON MODEL-II**

## REMODEL-PROLOAD

- THE ULTIMATE RENUMBERING PROGRAM. RENUMBERS ALL OR PART OF A PROGRAM (ALLOWS PARTIAL RENUMBERING IN MIDDLE OF PROGRAMS)
- PARTIAL OR COMPLETE MERGE OF TWO CASSETTE PROGRAMS

MODEL-I VERSION ... $35.00
MODEL III VERSION ... $35.00
**NOT AVAILABLE ON MODEL-II**

## COPSYS

- COPY AND VERIFY ALL MACHINE LANGUAGE (SYSTEM) TAPES WRITTEN IN STANDARD FORMAT. IF YOU BUY A MACHINE LANGUAGE PROGRAM, **COPSYS** ALLOWS YOU TO EASILY COPY THE PROGRAM ONTO ANOTHER CASSETTE AS A BACKUP

MODEL-I VERSION ... $15.00
MODEL-III VERSION ... $20.00
**NOT AVAILABLE ON MODEL-II**

**FOR DEALER INFORMATION CALL:** RACET COMPUTES

1330 N. GLASSEL, SUITE M, ORANGE, CA 92667
(714) 997-4950

## MODEL II SPEEDUP—FAST DISK I/O

**THIS IS AN ENHANCEMENT FOR TRSDOS 2.0 THAT WILL RADICALLY DECREASE DISK ACCESS TIME.**

- DISKS BOOT FASTER TO DOS
- IMPROVE DISK I/O UNDER BASIC

NEW

MODEL II ONLY ... $99.95

## MODEL II FASTBACK — FULL DISK BACKUP IN 55 SECONDS

**IN BUSINESS TIME IS MONEY, AND ONE BACKUP IS WORTH A THOUSAND TEARS.**

- WORKS ON SYSTEMS WITH 2 OR MORE DRIVES
- CAN REPLACE YOUR EXISTING TRSDOS 1.2 or 2.0 BACKUP UTILITY

NEW

MODEL II ONLY ... $75.00

## MODEL-II UTILITY PACKAGE

- ESSENTIAL FOR EVERY MOD-II OWNER
- RECOVER AND REPAIR FILES AND DIRECTORIES (BY JUST ENTERING A SINGLE COMMAND)
- XCOPY: SIMILAR TO COPY BUT CAN COPY ANY NUMBER OF FILES AT ONE TIME FASTER AND MORE ACCURATE THAN COPY SINCE RECORDS ARE COPIED IN GROUPS RATHER THAN ONE RECORDS AT A TIME. USING XCOPY YOU CAN COPY FILES THAT CAN NOT BE COPIED USING THE COPY COMMAND
- SZAP: PROVIDES THE CAPABILITY TO READ AND MODIFY ANY SECTOR ON A DISKETTE
- XHIT: CAN BE USED TO REPAIR A DISKETTE DIRECTORY
- DCS: DIRECTOR CATALOG SYSTEM IS A UTILITY FOR THE MANAGEMENT OF USER DISKETTES. SETS OF A MULTIPLE DISKETTE DIRECTORY FILE (WITH UP TO 1200 INDIVIDUAL FILE NAMES). ALLOWS SELECTIVELY LISTED OR PRINTED LISTS OF DIRECTORY FILES IN COMBINED SORTED ORDER (FOR EXAMPLE: LISTED ALPHABETICALLY BY DISKETTE, OR A COMPOSITE ALPHABETICAL LIST OF ALL YOUR DISKETTES)
- DEBUG II: ADDS SEVERAL FEATURES TO THE PRESENT TRSDOS DEBUG UTILITY INCLUDING SINGLE INSTRUCTION CYCLE, AUTO (LOOP) BREAKPOINTS, SUBROUTINE CALLING, BREAK-KEY DETECTION AND MANY OTHERS

MODEL II ONLY ... $150.00

## MODEL-II DEVELOPMENT SYSTEM

- THIS PACKAGE IS A MUST FOR ASSEMBLY LANGUAGE PROGRAMMERS
- INCLUDES THE **MICROSOFT EDITOR ASSEMBLER PLUS** WITH ENHANCEMENTS FOR THE MODEL-II
- A COMPLETE DISASSEMBLER
- SUPERZAP FOR READING AND MODIFY ANY SELECTOR ON A DISKETTE

MODEL-II ONLY ... $125.00

## MOD-II BASIC CROSS REFERENCE UTILITY

- LIST OR PRINT A SORTED CROSS REFERENCE TO ALL NUMBERS OR VARIABLES WITHIN A PROGRAM
- LIST OF PRINT ALL LINE NUMBERS CONTAINING A SPECIFIED STRING OF CHARACTERS

MODEL II ONLY ... $50.00

***** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE *****

**END USERS CALL:** COMPUTRONICS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

VISA

AMERICAN EXPRESS

24 HOUR ORDER LINE
(914) 425-1535

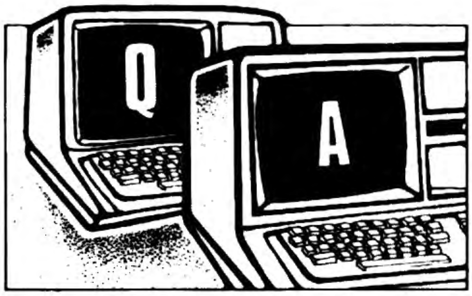
NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

**ADD \$3.00 FOR SHIPPING IN UPS AREAS**
**ADD \$4.00 FOR C.O.D. OR NON-UPS AREAS**
**ADD \$5.00 TO CANADA AND MEXICO**
**ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO**

**DEALER INQUIRIES WELCOME**

# FEEDBACK LOOP

by Terry Kepner




## Problems and solutions

*Send any questions or problems dealing with any aspect of TRS-80 microcomputing to "Feedback Loop,"* 80 Micro, *80 Pine Street, Peterborough, NH 03458.*

*I would like to share some comments on Lazy Writer. Contrary to your response to T.B. of Hartford, CT (Feedback Loop, September 1982), Lazy Writer does provide full proportional spacing for several printers, including the DW II.*

*Also, about the reply to G.B. of Hopkins, MN (also September), Lazy Writer can save formatted text to disk as it would appear on paper; it also has built-in RS-232 communications.*

*S.B.*
*Frederick, MD*

You're right, Lazy Writer does provide proportional spacing. But this is an optional extra ($29.95 for the DW II), which makes its price substantially more than NewScript ($124.95 versus $204.95).

I didn't know Lazy Writer could format text to the disk drives, or that it had an RS-232 driver built-in; for Mr. G.B., that makes it a worthwhile purchase.

*I'm writing in response to your answer to E.S. of Los Angeles (September issue) concerning copying disk files from Model III double density to Model I single density. I believe there's a fourth alternative besides the three you mention—Super Utility Plus.*

*This utility allows the Model III double-disk user to separately format (in several different DOS configurations) each drive and copy files from one to the other. The program costs $49.95 (cheap and includes many other powerful disk manipulation utilities).*

*I'm not a machine-language programmer, but I do appreciate Super Utility Plus's ease of operation. I've used it to correct disk errors, rename disks, and make back-ups of protected disks.*

*T.B.*
*Spokane, WA*

Super Utility Plus is definitely one of the most powerful disk utility programs on the market for the Model I and Model III computers. I agree with Paul Wiener, every serious user of a TRS-80 Model I or III should have it. But it won't necessarily transfer all programs from one DOS to another. Super Utility Plus is manufactured and sold by Powersoft (11500 Stemmons Fwy, Suite 125, Dallas, TX 75229, 800-527-7432) for $74.95. The Alternate Source will be selling it at its original price ($49.95) until January 1, 1983!

*I read with interest the letter from P.D. of Nashville (September issue). Coincidentally, we have the same equipment—a Model II with a Line Printer III—and up until a year ago we had the same problem.*

*Unfortunately, Radio Shack wasn't able to help me. I'm convinced the problem lies with the printer's design. Otherwise I think the LP III is an excellent machine.*

*Here's what happens: When the printer is programmed to print a character, tension is put on the ribbon. But when you're form feeding, only line feeds go to the printer. When the form is feeding through, the ribbon is loose. The trouble comes when the perforation between the forms passes in front of the print head, catches the loose ribbon, and pulls it away from the print head. At $13 per ribbon, that's an expensive problem.*

*Once I understood the problem, the solution was easy. I program the printer to print a dot at column 80 one line before the perforation and one line after the perforation. This puts enough tension on the ribbon to keep it out of the way. I've added this feature to all our programs with 100-percent success. I can now leave my printer on all night long printing five-part forms with carbons.*

*For unalterable canned programs, you have to devise a way to put enough tension on the paper so the perforations don't pull the ribbon, while making sure the tension won't tear the tractor-feed holes.*

*T.S.*
*Chicago, IL*

Sounds like an excellent, nontechnical solution for modifiable programs. Does anyone have a working solution for the nonmodifiable programs?

*In your reply to R.M. of Poland, OH, you listed two machine-language manuals for the 6809 chip. There's a third book available:* 6809 Assembly Language Programming *by Lance A. Leventhal, published by Osborne-McGraw Hill Publications.*

*P.W.*
*Jaffrey, NH*

Thanks for letting us know.

*I am considering a switch from my Stringy Floppy and tape system to disks, but I need more information before I will buy. Can you tell me the advantages and disadvantages of 40 versus 80-track drives, double-headed drives, and double-density drives. I intend to use the system for business activities such as inventory and accounts receivable.*

*W.G.*
*West Berlin, NJ*

The primary advantage of 80 over 40-track drives is space—there's twice as much of it. The disadvantage is some incompatibility with software on 35 and 40-track disks. Most of the incompatibility problems come about because 80-track drives actually put two tracks in the same space normally occupied by only one track of a 35-track disk.

An 80-track drive would treat a 40-track disk as if it were an 80-track disk where every other track was a duplicate of the preceding track. This presents a problem when you try to read a disk because the boot-up sector of a disk usually tells the DOS where the directory track is located. The 40-track disk tells the DOS the directory is on track 17, so the 80-track drive skips over to *its* track 17 (actually track 8 of the 40-track disk) and gets lost trying to find the directory.

Several DOSes on the market (MULTIDOS, DOSPLUS, to name two) include a command, Skip, that takes care of this problem, allowing you to read the 40-track disk. Writing to the 40-track disk is also okay if you don't try to read the disk in a 40-track drive later (which will read the new 80-track information alongside the old 40-track information, and get confused).

The problem still remains for those programs on protected disks that can't be treated as data disks. These special DOS disks require that you boot them up in a 40-track drive. To leave yourself the most flexibility, I suggest you get one 40-track drive to use with those special programs.

The double-headed drive's advantage is, once again, space. Double-headed drives store twice as much data in the same amount of space. The disadvantages are that they cost more (single-sided disks can be used, but the second side isn't guaranteed to be good and you could experience lost data problems), and most DOSes let you use only three drives on a system with a double-sided drive, with the fourth drive line being used to switch between side A and side B (but who cares when you end up with 50 percent more room than a standard four-drive system).

Double density lets you store almost twice as much information as with single-density drives. I don't know of any real disadvantages to double density. In fact a double-density board will improve the reliability of disk I/O in single density on a Model I just by being installed (double density requires the use of a data-separator circuit for the FDC chip controlling the disk drives). This data separator reduces, if not eliminates, DATA-LOST errors.

If you were to get one double-sided, double-density, 40-track drive and two 80-track double-sided, double-density drives, you would have about 1.4 megabytes (18*40*2 + 18*80*4 = 5760 sectors) of information on-line, compared with 400,000 bytes on a standard four-drive, 40-track, single-density system.

*I've been considering buying the Radio Shack PC-2 Pocket Computer. I haven't because Tandy doesn't have an editor/assembler, or even an instruction set, for Assembly-language programming—despite giving an example of a short machine-code routine in their manual. I was told their machine is identical to the Sharp Pocket Computer and that Sharp doesn't want to release the instruction set either. I asked a Sharp salesman, but he didn't know anything. Can you shed any light on the matter?*

*P.S. I'm a great fan of LCD displays, and am curious to know about the possibility of pocket computers in the future with larger displays (more than one line of text). Maybe portable is a better word than pocket.*

*J.W.*
*Albuquerque, NM*

The PC-2 is the Sharp Pocket Computer with Tandy's label on it. The Sharp salesman can't be blamed for not knowing because deals of this nature are conducted at a higher level of management.

An editor/assembler isn't available for either Pocket Computer. You are correct about why Tandy doesn't sell one for their PC-2. If Sharp, the licensing company, allowed them, Tandy most assuredly would sell an editor/assembler. After all, they've released one for all their other computers.

Trade journals have been talking about new LCD multiline displays for microcomputers the past few months. It shouldn't be long before we see a portable computer similar to the Kaycomp or the Osborne 1, except that it will have a light-weight LCD display instead of a 15 or 20-pound video monitor.

*My system is a Model I, Level II ALPS keyboard with 16K, an LNW expansion interface with AeroComp Doubler, two MPI B-52 drives, and a Centronics 737 printer.*

*I'm unable to read most disks formatted on other systems. When I try to boot them, the disk spins, stops, and the screen retains its previous image. If I connect someone else's drive to my system (using my power supply and cabling), one out of every five drives tried will load. Disks made on my system won't load on most other systems. I have swapped drives, power supplies, and cabling. The only item that seems to make a difference is the drive.*

*I've used my drives and interface on other systems and they work just fine.*

*I think there's a tremendous disparity in the speed and head-alignment adjustments for mini-drives. I haven't found anyone who can check and align my drives, so I don't have any real proof yet. Also, there seems to be some drive/interface interplay that I don't fully understand; they work fine separated, but not together.*

*M.B.*
*Laguna Beach, CA*

Ouch! It does sound like your problem is with the disk drives and their alignment. There are two commercially available programs that can help you identify the problem. The most versatile is Floppy Doctor by Dave Stambaugh (The Micro Clinic, 17375 Brookhurst, #114, Fountain Valley, CA 92708, $24.95/Model I, $29.95/Model III). Floppy Doctor is a complete diagnostics package. It thoroughly checks your computer system: RAM, ROM, and drives (35, 40, 77, and 80 track, single and double density). The drives are checked for proper motor speed, head alignment, read/write accuracy, head-seek accuracy, and several other important areas. If you don't already have this program, get it. It will tell you if the problem is with your drives, disks (I test for flaws on any new disks I buy), computer, or software. Knowing which tests your drive fails can sometimes pinpoint the exact cause for your drive's problems.

The other diagnostic package is DDT-Disc Drive Timer from DiscoTech (1150 Coddingtown Center, P.O. Box 11129, Santa Rosa, CA 95406, $29.95). This utility uses a graphic chart to show you your drive's rotation speed, and tells you how to adjust that speed to within normal system tolerances.

There is another possibility—check your disk drives to see if they both have the termination pullup resistor pack installed. Only the last drive on your drive cable should have it. If you're not sure where the resistor pack is, take both drives to a computer repair center and have the technician tell you if the resistor pack is installed properly.

*I've been so flustered by Radio Shack's EDTASM that I've almost given up. I have a problem saving the object output from the assembly onto*

# Santa's Gone Soft!


# With Help From Instant Software

### BUSINESS AND APPLICATIONS

| | | |
|---|---|---|
| 0222RD | Electronic Breadboard 32K, Mod I & III, disk | $59.95 |
| 0228R | Electronic Breadboard 16K, Mod I & III, tape | $49.95 |
| 0428AD | Electronic Breadboard 32K, Apple, disk | $59.95 |
| 0369RD3 | Easy Calc 48K, Mod III only, disk | $49.95 |
| 0269RD | Easy Calc 48K, Mod I only, disk | $49.95 |
| 0103R | Personal Bill Paying 16K, Mod I & III, tape | $14.95 |
| 5005RD | Master Directory 32K, Mod I only, disk | $29.95 |
| 0152RD | Business Analysis 32K, Mod I & III, disk | $99.95 |
| 0140R | Business Analysis 32K, Mod I & III, tape | $75.00 |
| 0380RD | Scriptr 48K, Mod I only, Disk | $34.95 |

### EDUCATION

| | | |
|---|---|---|
| 0099R | Typing Teacher 16K, Mod I & III, tape | $17.95 |
| 0100R | Video Speed Reading 16K, Mod I & III, tape | $14.95 |
| 0309R | Triangle Trig 16K, Mod I & III, tape | $19.95 |
| 0071RD | Geography Explorer USA 32K, Mod I & III, disk | $49.95 |
| 0276RD | Geography Explorer Mid East, 32K, Mod I & III, disk | $49.95 |
| 0111R | Word Watch 16K, Mod I & III, tape | $14.95 |
| 0257R | Math Master 16K, Mod I & III, tape | $17.95 |
| 0127R | Surveyors Apprentice 16K, Mod I & III, tape | $14.95 |

### GAMES

| | | |
|---|---|---|
| 0332R | Space Shuttle 16K, Mod I & III, tape | $19.95 |
| 0017R | Air Flight Simulation 16K, Mod I & III, tape | $14.95 |
| 0148A | Air Flight Simulation 16K, Apple, tape | $14.95 |
| 0159R | Jet Fighter Pilot 16K, Mod I & III, tape | $14.95 |
| 0329AD | Jet Fighter Pilot 32K, Apple, disk | $24.95 |
| 0223R | Cosmic Patrol 16K, Mod I & III, tape | $19.95 |
| 0224RD | Cosmic Patrol 16K, Mod I & III, disk | $24.95 |
| 0237R | Danger in Orbit 16K, Mod I & III, tape | $19.95 |
| 0247RD | Danger in Orbit 16K, Mod I & III, disk | $24.95 |
| 0051R | Ball Turret Gunner 16K, Mod I & III, tape | $14.95 |
| 0240R | Alien Attack Force 16K, Mod I & III, tape | $14.95 |
| 0171R | Flight Path 16K, Mod I & III, tape | $14.95 |
| 0043R | Santa Paravia 16K, Mod I & III, tape | $14.95 |
| 0229AD | Santa Paravia 48K, Apple, disk | $24.95 |
| 0273TI | Santa Paravia 16K, TI 99/4, tape | $14.95 |
| 0174A | Santa Paravia 48K, Apple tape | $14.95 |
| 0312R | Swamp War 16K, Mod I & III, tape | $14.95 |
| 0141R | Battleground 16K, Mod I & III, tape | $14.95 |
| 0378RD | Master Reversi 16K, Mod I only, disk | $24.95 |
| 0117R | Night Flight 16K, Mod I & III, tape | $14.95 |
| 0183A | Skybombers III 32K, Apple, tape | $14.95 |
| 0271AD | Skybombers II 32K, Apple, disk | $24.95 |
| 5020RC | Tower of Fear 16K, Color Computer, tape | $19.95 |
| 0084R | Music Master 4K, Mod I & III, tape | $14.95 |
| 0106R | Air Mail Pilot 16K, Mod I & III, tape | $14.95 |
| 0205RD | Flying Circus 32K, Mod I only, disk | $39.95 |
| 0163AD | Paddle Fun 32K, Apple, disk | $19.95 |
| 0219R | House of 30 Gables 16K, Mod I & III, tape | $14.95 |
| 0325RD | Mission, Mud 32K, Mod I & III, disk | $19.95 |
| 0387RD | Sparrow Commander 32K, Mod I & III, disk | $19.95 |
| 0386RD | Kitchen Sink 32K, Mod I & III, disk | $19.95 |
| 0161AD | Apple Fun 32K, Apple, disk | $19.95 |

### UTILITIES

| | | |
|---|---|---|
| 0230R | TLDIS 16K, Mod I & III, tape | $19.95 |
| 0231RD | DLDIS 32K, Mod I only, disk | $24.95 |
| 5700RD | Super>>Terminal 32K, Mod I & III, disk | $95.00 |
| 5003R | Ultra-Mon 16K, Mod I & III, tape | $24.95 |
| 0232R | The Disassembler 16K, Mod I & III, tape | $14.95 |
| 0245R | Programmer's Primer 16K, Mod I & III, tape | $14.95 |
| 0139RD | Disk Scope 32K, Mod I only, disk | $24.95 |
| 0184R | TRS-80 Tests 16K, Mod I only, tape | $14.95 |

## TO ORDER:

**See your local Instant Software dealer or call toll-free 1-800-258-5473 Dept. A-65 In New Hampshire 1-603-924-9471 Dept. A-65**

**Mon.-Fri. 8:00 am-4:30 pm E.S.T.**

**Include $2.50 for Shipping & Handling**

***TRS-80 is a trademark of Radio Shack a division of Tandy Corporation**

****APPLE is a trademark of Apple Computer Inc.**

# Instant Software™

**Peterborough, N.H. 03458 USA** **A subsidiary of Wayne Green Inc.**

PRICES SUBJECT TO CHANGE

*cassette. EDTASM works fine except for this one area. I have both Version 1.0 and Version 1.1 for tape systems, but both have this problem.*

*My system is a Level II, 16K computer with the dual cassette interface from Sel-tronics. I removed the interface from the system, but still have the problem. I tried the computer store, Fort Worth, and friends with no luck. I'm considering getting EDTASM Plus, but am worried about the cassette operation.*

*B.B.*
*San Jose, CA*

It sounds like the cassette interface in your keyboard is at fault. Most of the early Model Is had a flaw in the design of the cassette interface that made it very difficult for the computer to read from and write to cassette tape. For a time Radio Shack offered a free modification to the cassette interface that corrected this problem (the XRX cassette modification). The modification is still available, but now they usually charge a nominal fee for the parts and labor.

Before you charge off to the nearest Tandy computer retail center, check these points out first:

Are your tape recorder heads clean and properly aligned? Borrow a friend's recorder and see if you have better luck saving and loading. If you do, the problem is most likely with your tape recorder and not EDTASM or your computer.

Have you tried using your copy of EDTASM on another computer (such as the local Radio Shack store's or a friend's)? If your EDTASM works on other computers, the problem is with your system. If the EDTASM fails to work at the store, show the salesman and get a replacement copy.

If you have chronic problems with cassette I/O, regardless of the program or tape recorder you're using, the problem is definitely with your cassette interface and should be taken to a repair center.

*I have a PMC-80 with a built-in cassette recorder. In the August issue of* 80 Micro *there's a game called Naval Wars. To connect the computers together you need the mike, remote and earphone plugs exposed, which aren't available on the PMC-80. If you plug a plug into the computer a second tape player can be accessed by the normal I/O statements. Is it possible to convert the program to work on the PMC-80?*

*D.H.*
*Conway, PA*

After looking at the program, I don't see why you can't just replace the INPUT#-1 and PRINT#-1 commands with INPUT#-2 and PRINT#-2 and use the second tape player plug on the PMC-80. If you're going to be playing with another PMC-80, both machines will need the modified program. If the other machine is a TRS-80, use the modified program in your machine only.

*I have a Model III 48K cassette-operated computer, with a Radio Shack LP VIII. I use the computer for word processing with Scripsit, which is where I have my problem. The LP VIII is capable of changing type fonts, underlining, and printing boldface characters, none of which are supported by Scripsit.*

*A friend told me about a cassette program, "Scrips Plus," that corrects this problem, but I can't find the software company, Quality Software, who's supposed to be one of your advertisers.*

*Another problem I have is line renumbering in Basic. On my PC-1 this is easy to do. The need often arises to either replicate or duplicate some programming without retyping. I have seen renumbering mentioned in ads; are there any subroutines that you've published in the past that can change line numbers easily?*

*D.P.*
*Houston, TX*

The Scripsit modifying program you want is called Scriplus, for Scripsit. The company that sells it has changed its name from Quality Software to PowerSoft (11500 Stemmons Fwy, Suite 125, Dallas TX 75229), which is why you couldn't find them in the magazine. Scriplus retails for $24.95 (cassette or disk) and lets you use all your printer's features: backspace, underline, boldface, and so on.

There's another Scripsit-modifying program available called Scriptr. It's sold by Pioneer Software (1746 NW 55th Ave., #204, Lauderhill, FL 33313) and retails for $40 (cassette and disk).

For more details on both of these programs, read pages 101 and 102 of the September 1982 issue of *80 Micro*.

I think you have a slight misconception about renumbering programs. The PC-1 is unusual in its method of renumbering Basic lines. Most Basic line-renumbering programs change the line numbers of your Basic program to a new sequence. They don't move or duplicate the Basic lines. Several of the DOSes on the market (such as DOSPLUS and MULTIDOS) include special commands in their DOS Basic that let you move or duplicate Basic lines. But since you don't have disk drives, that isn't a solution (and the only renumbering program I can find in the magazine is in the March 1981 issue, page 256).

What you can do is get a machine-language program that lets you use screen-editing techniques to duplicate Basic lines to new locations (almost exactly the same way the PC-1 duplicates lines. You just type a new line number over the old one, press enter, and you've duplicated it.) There are two of these available. The first one is called IRV and is sold by The Programmer's Guild, (P.O. Box 22, Peterborough, NH 03458 (603) 924-6065) and retails for $24.95 (Model I cassette version), $29.95 for disk. The Model III version is called Superkeys and retails for $49.95 (disk only). It's available from Advanced Operating Systems (450 St. John Road, Suite 792, Michigan City, IN 46360, (219) 879-4693).

The second program is sold by Interpro (P.O. Box 4211, Manchester, NH 03108, (603) 669-0477) and retails for $24.95 (specify disk or tape). It automatically adjusts itself to your computer's memory size (16K, 32K, or 48K), and computer type (Model I or III).

Both programs let you assign messages to the keyboard keys (so you can define the shift, D key combination to print "Delete," the shift, X key combination to print Data ***************, and so forth) and let you use the standard Level II Edit commands together with their screen-editing capabilities to concatenate or change Basic lines on the screen.

Of the two programs, Key Commander is more sophisticated and versatile; it's also easier to use.

*We're looking for information on the Color Computer: books, programs, hardware, software, and computer clubs. We're also interested in a program called Smart Package, which is supposed to allow direct linking to other computers without the use of Compu-*

*Serve. Can you help?*

*R.&K.J.*
*Madera, CA*

Your best source of information on the Color Computer is *80 Micro* (this isn't just a plug for this magazine, it's the truth). *80 Micro* has more companies advertising products (books, programs, hardware, and such) for the Color Computer than either of the two publications specializing in the Color Computer (*Color Computer News*, and *The Rainbow*). Check September's Feedback Loop for two machine-language books on the Color Computer, and this month for a third one.

One very good source is B. Dalton Books, located in many shopping malls. B. Dalton has the largest and most complete selection of computer books of any store I've ever been in, including computer stores.

New computer club listings can be found in *80 Micro* issues, with an updated list printed each year of all the clubs known. *80 Micro*'s November 1981 issue included as complete a listing as possible. It will be updated in the Anniversary issue. In addition, the *Radio Shack Computer Newsletter* also lists new clubs.

The Smart package is probably a communications program, although I can find none with that name. Several similar programs are available: Datapak (Cer-Comp, 5566 Richochet Ave., Las Vegas, NV 89110, $24.95 cassette, $49.95 disk). Colorcom/E (Spectrum Projects, 93-15 86th Drive, Woodhaven, NY 11421, $49.95), Micro-Text (The Micro Works, P.O. Box 1110, Del Mar, CA 92014, $59.95 ROMPAK), Colorterm (Martin Consulting, 94 Macalester Bay, Winnipeg, Manitoba, R3T 2XT Canada, $34.95), and Super Color Terminal (Nelson Software Systems, P.O. Box 19096, Minneapolis, MN 55419, $39.95 tape, $49.95 ROMPAK, $69.95 disk).

These programs make the world of telephone and computer communications available to the Color Computer user. CompuServe, Timenet, and The Source are all data-base computer networks you can tie into using a communications terminal program. When you buy the CompuServe package from Radio Shack, you get one hour of time on CompuServe and a rather primitive communications program. CompuServe isn't necessary to communicate with other computers—only the terminal program (and a modem/phone hardware hookup) are needed for that. There are several hundred computer bulletin board services in the world. Many of their phone numbers have been listed in past issues of *80 Micro*.

*I want to upgrade my TRS-80 Level I computer without paying Radio Shack prices. I've worked with electronics, but not computers. Is there any alternative from another company, or a do-it-yourself ROM pack? I've put in my own 16K RAM.*

*L.H.*
*Cortland, NY*

The only source of Level II ROMs is Radio Shack, which sells them for $99 and charges $25 to install them. If you don't mind voiding your Radio Shack warranty, you can buy the ROMs from Radio Shack's National Parts Division (900 East Northside Drive, Fort Worth, TX 76102, (817) 870-5662). The two-chip Level II ROMs cost $66.73 (the A-B chip, serial number AXX3031, is $41.58, the C chip, serial number AXX3030, is $15.15). If you have MasterCharge or Visa, you can call direct and order the parts by phone. There's also a $1.50 shipping and handling charge.

*I bought my Model III in November 1980, and after the warranty expired, I upgraded it to 48K and then added a non-Radio Shack disk drive, all of which operate flawlessly. The problem is with my keyboard. About eight months ago the I key stuck in the on position. A half hour with a screwdriver and ohmmeter convinced me that the only problem was a stuck keyboard switch—one bad key module. I tried but couldn't get a replacement module. After several phone calls I gave up and swapped the failing key module with the period module from the numeric key pad.*

*I continued looking, but couldn't find either a single key module or a keyboard at any price. I was told that I could order a module, but after three months I gave up. Finally I was told that Radio Shack doesn't sell keyboards over the counter; I would have to bring in my machine for service and they would replace the necessary parts.*

*I'm a little leery of taking my machine with its $700 dollars of alien hardware to Radio Shack (I've heard the horror stories of people getting their computers back with their homebrew modifications in a bag). And why should I surrender my machine to a repair center and pay for parts and labor (not to mention down time), when I could easily repair it myself?*

*R.C.*
*Hyattsville, MD*

Yes, you can buy a new keyboard from Radio Shack. All you do is order it from National Parts (see previous letter for the address information). The Model III keyboard's serial number is AXX0205, and costs $68.56. It shouldn't take more than two weeks for the part to be delivered. (They usually promise store delivery in three work days.)

*I'm having a few minor problems with my TRS-80 Model I monitor. The image on my display shows a small ripple that travels slowly from the bottom of the screen to the top, and then starts again. Also, the whole image on the screen is shifted to the right approximately ¼" to ½", making the rightmost part of the image fuzzier than the rest. Finally, the monitor sometimes squeals from the inside. The noise is similar to what you hear from a color tv, only louder. Can you help me with these?*

*H.L.*
*South Bend, IN*

Anyone having difficulties with a Model I/III computer system should buy Dennis Kitsz's book, *The Custom TRS-80 & Other Mysteries* (available from the *80 Micro* Bookshelf, advertised in this magazine. Ask for catalog number BK1218). The book retails for $29.95.

All three problems are covered in *The Custom TRS-80*, pages 253–254. The first problem isn't the fault of your computer, but is the result of radio frequency interference (RFI) from another device. The most common source of this problem is the triac used in light dimmers. The second most common source of interference comes from fluorescent light bulbs, followed by failing neon bulbs (usually stereo lights); in a business, check your copier. This is one of the hardest problems to eliminate because the problem originates from outside the computer.

The second problem can be corrected by opening the keyboard unit (Warn-

ing: Opening the computer will void your Radio Shack warranty) and adjusting the horizontal and vertical control potentiometers on the right side of the keyboard, facing down with respect to the normal operating position of the keyboard.

The last problem is only a nuisance. The high-voltage lines are experiencing a slight arcing problem. The solution is to open the case (same warning as above), *after* letting it sit unplugged for an hour to let the power supply capacitors discharge safely, and clean the insides. If you find the hissing irritating, take it to the repair shop and have them put more dope on the high voltage lines. If you know where these lines are yourself, and you have the right kind of dope, you can do this yourself.

*My TRS-80 screen starts to scroll and gets wavy when I fill it with graphics (i.e., white-out the screen). What's wrong? How can I fix it?*

*J.H.*
*Grosse Ponte Park, MI*

First, get Dennis Kitsz's book (the same one mentioned before). Second, the video monitor has to be modified to fix this problem. Unplug the monitor an hour before you begin. (Warning: Not only will opening the case void your warranty, but if you don't let the monitor sit unplugged for an hour before starting, the voltages you'll find inside could kill you.) After removing the case, locate the resistor marked R14 on the small circuit board closest to the side of the case (some monitors only have one). The resistor should be marked 3.3K ohm (orange, orange, red bands marked on the resistor). Solder a 1.5K ohm resistor on top of the 3.3K resistor (making them parallel to one another). Put the cabinet back together. This should take care of your problem.

*While reading some* 80 Micro *back issues, I noticed several programs with different version numbers, such as Star Trek 4.0. How are the version numbers arrived at?*

*M.C.*
*Concord, NH*

They make them up. Star Trek 4.0 was written several years before Adventure International's Star Trek 3.5. Generally, unless the program is written by the same author, the version number is meaningless. If the same author is using different version numbers, the numbers correspond to the program's release sequence, thus Scripsit Version 1.1 is an improvement of Scripsit Version 1.0

*My almost new, yet out of warranty, Radio Shack Modem I recently died. Repair charge: $56.*

*Radio Shack claims the damage was caused by a power surge and is not the fault of their equipment. I would like to know if this is a plausible explanation or am I being sandbagged. Also, how can I prevent this from recurring?*

*E.M.*
*Birmingham, AL*

Yes, this is possible. I have had the unfortunate luck to see most of a TRS-80 computer destroyed by a line-voltage power surge. Radio Shack sells a power-line filter/multiple outlet box for $49 that will limit the amount of ac power line interference your system gets. Many other companies, advertisers in this magazine, offer surge protectors and spike suppressors for your computer. Since you've already been zapped once, you'd better get one of these right away.

*I've decided to add lowercase capability to my Model I, but I don't have any instructions or a schematic. I do have an SPST switch and a 2102 RAM chip. Can you help?*

*I'm also considering upgrading my system to include disk drives and double density, but I'm worried about compatibility with other computers. If I have 40-track, double-density drives, will I be able to exchange programs on disks with someone who has only Radio Shack 35-track, single-density drives? Would it help if both parties operated under the same DOS?*

*H.S.*
*Cincinnati, OH*

*The Custom TRS-80* by Dennis Kitsz has the procedure on how to install lowercase capability (as well as about half the questions in this month's column).

Compatibility isn't a problem if you use 40-track drives and a DOS capable of reading and writing in single and double density. All you have to remember to do is to format and write the exchange disk as a 35-track, single-density disk. DOSPLUS, LDOS, NEWDOS-80, and MULTIDOS all read and write to both single and double-density, 35 or 40-track disks. Of the bunch, MULTIDOS offers the most advantages for its price: MULTIDOS is the only DOS that reads and writes to disks no matter what system they were written with (although NEWDOS80 is a problem if the directory is more than one track in length). For the most compatibility, both operators should use the same DOS.

*I am a blind freelance writer and have thus far been unable to use my husband's 48K Model I tape-based computer. Our quest for help to get the computer to read out in an audio medium has been long and frustrating. Other blind users I know of use Apples. My ultimate goal is to have a word processor I can use.*

*Do you know of an easier way to accommodate an audio medium other than dissecting and modifying each program to be used with it? Can you suggest a way to turn our 80 into a word processor I can use?*

*N.K.*
*Forked River, NJ*

My first thought was to call IJG Computer Services (1260 West Foothill Blvd., Upland, CA 91786, (714) 946-5805) and ask about the progress of Talking Pencil, a word processor for the visually handicapped. Unfortunately, while they have plans to eventually release such a product, the processor is barely out of the planning stages.

The next step was to call the Votrax Corp. (500 Stephenson Hwy, Troy, MI 48084, (313) 588-0341). Votrax makes a device called Type 'N Talk, which takes ASCII text input from an RS-232 port and converts it into speech (it works quite well). However, it requires software to direct the data to the RS-232 from the main program (as well as control codes to determine when and what to send). Unfortunately, again, the results were negative; no one at Votrax was aware of a word processor using the Votrax unit.

Other phone calls also failed to provide a solution. Since I couldn't seem to help you, does anyone else know of a solution for N.K.'s problem? ■

*Terry Kepner is a free-lance writer and programmer, and the vice president of Interpro. He's been writing about microcomputers since 1979.*

# DOUBLE FEATURE
## FOR YOUR TRS-80


# HERE'S YOUR TICKET TO MANY THRILLING HOURS OF ENJOYMENT

**TWO BRAND NEW GRAPHIC ADVENTURES IN ONE PACKAGE — FOR YOUR TRS-80**

*By Roger Schrag*

## SPOOK HOUSE

Horrors! A deranged madman has locked your unconscious form inside of a deserted creep house at a crazy carnival. You awake to some sobering facts: a bomb is set to explode, and only 30 real-time minutes stand between you and eternity. Somewhere, somehow you must locate and defuse the explosive — but we'd be less than honest if we thought you had more than a ghost of a chance! Brrrr — a real cliffhanger!

## TOXIC DUMPSITE

Something's gone very, very wrong at the Toxic Dumpsite where the treatment and burial of deadly contaminants take place. The entire plant will explode like the Fourth of July in less than 30 minutes — unless you can avoid the many traps and protection systems and shut the plant down in time. Of course, time is the one luxury you haven't got . . . !

This is it — the blockbusting debut of the long-awaited Graphic Adventures! Now, your TRS-80 can hobnob with the best of 'em. Over 50 full screens of sharp, quality graphics breathe a realism and vitality into Adventuring, the likes of which you've yet to see on the TRS-80. Both Graphic Adventures feature game save, extensive vocabulary, real-time action (yes — we said real-time!) and accept FULL sentences! Exclusive area showings brought to you by Adventure International. Now playing at a TRS-80 near you!

**TRS-80 48K Models 1 & 3 disk 012-0164 — $39.95**

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.

Adventure INTERNATIONAL

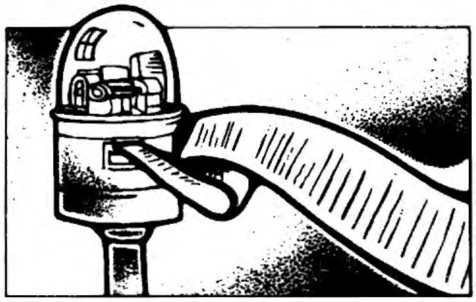
PRICES SUBJECT TO CHANGE

~19

Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194

# MONEY DOS

by J. M. Keynes



## Explore your options

I'm sure the options market was invented by brokerage firms because it allows those with small bankrolls to gamble. If you wish to trade options here are the rules.

An option is the right to buy or sell 100 stock shares at a set price called the "striking price." All listed options have an expiration date. At any time before the expiration date they may be bought and sold each market day. All options expire at 3 p.m. EST on the third Friday of the expiration month.

A "call" is an option to buy 100 shares of the stock. A "put" is an option to sell 100 stock shares.

An "exercise" converts an option to the actual stock. The option holder (buyer) can convert to stock at any time, provided he has the money to pay for it. For example, if you bought a November 30 call on XYZ last June for $200, and today XYZ is at $37 per share and you want to get the stock, your option, $3,000, plus a commission gets you 100 stock shares. You have exercised your option. In practice, few options are exercised. Most investors who own XYZ options on November 30 would sell in the market for about $700.

"In the money" refers to an option, like the XYZ above, which has some tangible value. That is, the stock is above the striking price.

"Out of the money" is an option with no tangible value; that is, the current stock price is below the striking price. The price it commands will depend on how much time is left before expiration and how close the stock price is to the exercise price.

You are "covered" when you sell an option on a stock you own. This is the least risky investment in the market. (See one of my previous columns where I go into covered option writing in depth.) You are also covered if your options are properly spread. I'll get into that later.

"Naked" is when you sell an option on stock you don't own, and if the holder exercises it, you are forced to buy the stock at the current market price in order to deliver to the option holder. Being naked implies unlimited risk. If you sell an option on ASA with a striking price of 30 and ASA goes to 90, you are stuck. Of course, if you have any judgment, you will have covered your naked position by buying the option back.

The strategy of simply buying an option is obvious. If you think a stock is going up, you buy a call. If you are bearish, you buy a put. Program Listing 1 assists you in determining if an option is over or underpriced.

Another strategy that I employ frequently is "spreading." This is when you buy one option and sell another on the same stock.

XYZ is at 23. You think it will go to 30 by February, so you buy the February 25 outright for about $200 per option and concurrently sell the February 30 for about $75. You put up the same $200 for the February 25, but you get back $75 from the sale of the February 30, making your net risk $125. This is an example of "vertical spread."

In many cases I prefer spreads. It is true that you give up the potential for an enormous move in a short time, but over many years, I have determined that few stocks rise more than 33 percent in any 90-day period. Therefore, I prefer the spread because I think the odds favor it. Figure 1 clearly shows the vast advantage spreading has for even a 45 percent move-up in the stock. Should the stock remain under 25, both calls will expire worthless. If you are considering buying a call that is way out of the money (buying a 50 call on a stock that is now at 40), the vertical spread portion of Fig. 2 will make spreading a compelling strategy.

Using the "calendar spreads" strategy requires the least investment. An example would be selling a February (nearby) and buying a May (deferred) option. As this is written, there are over 25 stocks on which this can be done for under ½ or $50 per spread. For example: Today you can buy the Mesa Petroleum April 20 and sell the MSA January 20 for a debit of only ¼ or $25 per spread. MSA closed today at 14. The risk is that what you put up as the April option must always be worth more than the January. Should MSA be at 20 on the third Friday of January, the option you sold would expire worthless while the one you bought has

Compare outright purchase of an option vs concurrently selling a higher priced one.

The calculations assume equal dollar investment in each. Profit or loss at expiration. Current stock price 23, investment $200.

| AT EXPIRATION | OUTRIGHT | SPREAD | % STOCK GAIN |
|---|---|---|---|
| $25 PER SHARE | −200 | −200 | 8 |
| $26 PER SHARE | −100 | −40 | 13 |
| $27 PER SHARE | 0 | 120 | 17 |
| $28 PER SHARE | 100 | 280 | 21 |
| $29 PER SHARE | 200 | 440 | 26 |
| $30 PER SHARE | 300 | 600 | 30 |
| $31 PER SHARE | 400 | 600 | 34 |
| $32 PER SHARE | 500 | 600 | 39 |
| $33 PER SHARE | 600 | 600 | 43 |
| $34 PER SHARE | 700 | 600 | 47 |
| $35 PER SHARE | 800 | 600 | 52 |
| $36 PER SHARE | 900 | 600 | 56 |

*Figure 1*

## Quality Computer Services Presents ...

# Hard Disk Sub-Systems 20 Megabytes for Under $3500!

**Our Systems Are Compatible with Your**

- TRS-80® Model I, II, III, XVI
- Apple II®
- IBM® PC
- NEC™
- S-100™
- Altos™
- NorthStar™
- Superbrain™
- Panasonic™
- Epson™
- Heath/Zenith™
- and Others

**Now Available!**

**Optional 10 Meg. Tape Back-up**


Now you can expand the capability of your microcomputer with cost-effective mass storage from Q.C.S. Our integrated family of 6-, 12-, 20-, and 40-million-byte systems lets you operate your micro with the speed, convenience, and proven reliability of systems costing thousands of dollars more! And as your needs grow, Q.C.S mass storage can grow too. You can start with an entry-level 6-million-byte system and expand 80 million bytes ... without scrapping your existing programs or operating system.

### The completely assembled and tested system includes:

- Switching power supply
- Winchester drive
- Host adapter card
- Sealed case
- Expansion (enclosure, controller, and power supply designed for future backup hardware or increased storage)
- Software
- Auto-attach CP/M
- Hard disk utilities
- Realtime clock
- Formatting program (assigns alternates for any weak sectors detected during formatting—assuring the lowest possible error rate)
- Warranty (a full one-year warranty on all parts and workmanship)

### Also available:

- Attach program for TRSDOS and OASIS
- **New CP/M dynamic generator:** configured software; menu-driven for 6-, 12-, 20-, 40-, 80-megabyte drives

### The following systems software can be utilized:

**TRS I, III**—attach software for DOS 4.0,* LDOS
**TRS II, XVI**—CP/M Racet for TRSDOS,* Turbo DOS, OASIS
**TRS Models II** and **XVI** use 6 Meg for CP/M and 6 Meg for TRSDOS.
**Apple**—CP/M and DOS 3.3
**NEC**—CP/M
**S-100**—CP/M
**Heath/Zenith**—CP/M
**IBM**—PC DOS, USCD, etc.

* Available at an additional cost.

See your local dealer or call:

**QCS** Qu-Se inc **Quality Computer Services** 48

178 Main Street, Metuchen, New Jersey 08840
Telephone 201/548-2135

**Toll-free number: (800) 631-5944**

Quantity discounts available. Distributor, Dealer and OEM inquiries invited.

90 days to go and would be worth about 2 to 2¼—a profit of perhaps 1,000 percent. Even if MSA is only at 17, the April 20 should be worth at least ½ for a 100 percent profit. If MSA is unchanged at 14, it is likely you will be able to recover most of your investment. The program in Fig. 2 tells you what the option would be worth with 90 days to go if the stock is unchanged.

Several times each year a situation develops that can provide the clever investor with an opportunity for big profits with essentially zero risk. Examine the following illustration. Teledyne recently sold for 75. For a brief period you could have bought one October 90 and one October 110 and sold *two* October 100s for a net credit of 3/8. That is, the proceeds from the sale of the two October 100 calls paid in full for the October 90 and 110 with 3/8 left over, nearly paying the commission. You put up nothing. (Note: One of the October 100 calls is uncovered and requires margin money, or if you have any marginable securities in your account, they will stand good for the margin required.)

Check out this arithmetic. If on the third Friday of October, TDY sells for less than 90, all the options expire worthless. The investor lost nothing. If TDY is between 90 and 110, the investor profits. The maximum profit is achieved at 100, for then the 100 and the 110 options are worthless and the 90 is worth 10, or $1,000. If TDY is above 110, the investor loses only a commission as the combined value of the 90 and 110 equals the value of two 100 options.

You're not laughing now! The order to the broker was, "TDY, buy calls 1 October 90 and 1 October 110. Sell calls 2 October 100 at 3/8 credit." This is one area where established brokers may get executions that the discounters can't touch. Your broker will be surprised that such opportunities exist.

There is another strategy that I have successfully employed. Today Warner Communications (WCI) closed at 37. There is an option (actually a warrant on the American Exchange) that gives one the right to buy one share of WCI at $55 per share anytime up to April 1, 1986. It closed at 13 3/8. The more conservative investor might consider buying the stock and selling (shorting) the war-

*Program Listing*

```
15 FOR I=1 TO 1800: NEXT
20 CLEAR: REM IF YOU WANT ONLY THE VERTICAL SPREAD PROGRAM
TYPE RUN 5000
30 IF PEEK(293)<>73 THEN POKE 16553,255: REM SEE IF MODEL 1
40 POKE16419,196: CLEAR1000: DIM A(300): CLS: T$="###.##": Z=100
: A(110)=0: GG=1: DIM S$(100)
50 INPUT "CURRENT PRIME RATE (14% AS 14)";I
60 INPUT "CALENDAR SPREAD=1  VERTICAL SPREAD=2  ELSE ENTER";CS:
CLS
70 PRINT "COMPARE ESTIMATED PRICES WITH ACTUAL OPTION PRICES.
80 INPUT "SYMBOL OF STOCK";SS$
90 CLS: PRINT @459,"I'M LOOKING IN DATA BANK FOR ";SS$
100 READS$,V
110 IFS$="END" THEN CLS: PRINT SS$;" IS NOT IN DATA BASE, ENTER
VOLITILITY.": PRINT "ENTER A NUMBER FROM 5 TO 60": PRINT "5 FOR
UTILITIES UP TO 60 FOR MOST VOLATILE";: INPUT V: GOTO 130
120 IFS$<>SS$ THEN 100
130 CLS: INPUT "ACTUAL PRICE OF FIRST OPTION.";HP
140 INPUT "ACTUAL PRICE OF SECOND OPTION";LP
150  PRINT "FIRST OPTION (LOWER STRIKING PRICE, OR LONGER OPTION
): "
160 PRINT "TOUCH A 1 FOR PUT OR A 2 FOR CALL": GOSUB900
170 IFVAL(Q$)>2,CLS: PRINT "BAD ENTRY, TRY AGAIN": GOTO 160
180 A(105)=VAL(Q$)
190  INPUT "   # DAYS TO EXPIRATION                   ";D1: M=D1/30
200  INPUT "   STRIKING PRICE                         ";A(101)
210  PRINT "SECOND OPTION (HIGHER STRIKING PRICE, OR SHORTER OPT
ION): "
220 PRINT "TOUCH A 1 FOR PUT OR 2 FOR A CALL": GOSUB900
230 IFVAL(Q$)>2 THEN CLS: PRINT "BAD ENTRY, TRY AGAIN."
240 A(106)=VAL(Q$)
250 INPUT "    # DAYS TO EXPIRATION                   ";D2: N=D2/30
260  INPUT "   STRIKING PRICE                         ";A(102)
270  INPUT "CURRENT STOCK PRICE                       ";P
290  INPUT "QUARTERLY DIVIDEND (40 CENTS AS .40)   ";D: D=D*4
310  PRINT
320 CLS: PRINT @410,"I'M THINKING"
330  IF A(105)=1 THEN  A$="PUT "
340  IF A(105)<>1 THEN  A$="CALL"
350  IF A(106)=1 THEN  B$="PUT "
360  IF A(106)<>1 THEN  B$="CALL"
370   I=I/Z: V=V/Z: D=D/Z: T=M/12: U=N/12
380   A(61)=T*.5:  A(62)=0
390    A(60)=(T/A(61)-A(61))*.5
400  IF(A(60)=0)+(A(60)=A(62))THEN420
410    A(61)=A(61)+A(60):  A(62)=A(60): GOTO 390
420  IF U=0 THEN A(63)=0
430   A(63)=U*.5:  A(62)=0: A(60)=(U/A(63)-A(63))*.5: GOTO 450
440   A(60)=(U/A(63)-A(63))*.5
450  IF(A(60)=0)+(A(60)=A(62)) THEN 470
460   A(63)=A(63)+A(60):  A(62)=A(60): GOTO 440
470  IF A(102)=0 THENA(102)=5
480  IF A(101)=0 THENA(101)=5
490  FOR J=101 TO 102
500  IF A(J+4)=1 THEN Q=A(J)+(A(J)-P)
510  IF A(J+4)=1 THEN A(30)=Q/A(J)
520  IF A(J+4)<>1 THEN A(30)=P/A(J)
530  IF J=102 THEN A(61)=A(63)
540  IF J=102 THEN T=U
550   A(36)=LOG(A(30))
560   E=A(36)+(I-D+.5*V*V)*T
570   E=E/(V*A(61)): F=E-V*A(61): A(40)=(D*T)*(-1): A(45)=EXP(A(
40))
580   A(48)=A(45): A(50)=E
590  GOSUB810
600  IF A(J+4)=1 THEN 620
610   A(J+2)=A(48)*P*A(52): GOTO 630
620   A(J+2)=A(48)*Q*A(52)
630   A(40)=(I*T)*(-1): A(45)=EXP(A(40)): A(48)=A(45): A(50)=F
640 GOSUB810
650   A(J+2)=A(J+2)-A(48)*A(J)*A(52)
660  IF A(J+4)=1 THEN  A(J+2)=A(J+2)*.8498
670 IF II=1 THEN  GOTO  980
680 NEXTJ
690 CLS: PRINT @330,"OPTION EVALUATIONS ON STOCK SYMBOL ";SS$
700 PRINT @420,"ESTIMATED    ACTUAL
710 PRINT @448,"";: PRINT " # 1 VALUE OF";D1;"DAY, $";A(101);A$;
" =  $";: PRINT USING T$;A(103);: PRINT "     ";: PRINT USING T$
;HP
720  PRINT " # 2 VALUE OF";D2;"DAY, $";A(102);B$;" =  $";: PRINT
 USING T$;A(104);: PRINT "     ";: PRINT USING T$;LP
730 PRINT STRING$(64,131)
740 X1=HP-A(103): X2=LP-A(104): O1=INT(100-((A(103)-X1)/A(103)*1
00))
```

*Listing continues*

rant. You put up the difference (23 5/8). As long as WCI is above about 24, you can't lose. Anything above 24 is net profit. Maximum profit occurs at 55 or above. A similar situation exists in American Express (AXP).

If you want to buy a stock that has listed options, you may be able to artificially create (so to speak) the stock by using options, and in the process, save money. Last week Teledyne TDY was at 79. The April 80 call and the put were selling for 12. I wanted 1,000 shares of TDY, but rather than put up $79,000, I bought 10 April 80 calls and sold 10 April 80 puts. I have essentially all the profit potential and risk of owning the stock. The difference is that I put up no money. My $79,000 can be earning interest for the next eight months (about $6,300). At expiration, if I still want to own the stock, I can exercise my call option and buy it for 80 even though it may be at 110 (would you believe 130?). If I choose to sell the call I must pay capital gains taxes. There is no tax if I exercise.



*Listing continued*

```
750 O2=INT(100-((A(104)-X2)/A(104)*100))
760 IFO1>0 THEN R$="OVERVALUED": GOSUB910 ELSE R$="UNDERVALUED":
 GOSUB910
770 GG=GG+1: IFO2>0 THEN R$="OVERVALUED": GOSUB910 ELSE R$="UNDE
RVALUED": GOSUB910
780 IFM>3 OR N>3 THEN GOTO  970
790 IFOP=1 THEN  930
800 PRINT @975,"";: INPUT "PRESS ENTER TO CONTINUE";E$: CLS: IFC
S=2 THEN H=A(101): L=A(102): A=P: GOTO  5100 ELSE 20
810   A(53)=1/(1+.231642*ABS(A(50)))
820   A(40)=-A(50)*A(50)/2: A(45)=EXP(A(40)): A(51)=.398942*A(45
)
830   A(55)=1.330027:  A(56)=1.82126:  A(57)=1.78148:  A(58)=.35
6564
840   A(59)=.319382
850   A(52)=(((A(55)*A(53)-A(56))*A(53)+A(57))*A(53)-A(58))*A(53
)
860   A(52)=A(52)+A(59): A(52)=1-A(51)*A(53)*A(52)
870  IF A(50)>=0 THEN890
880   A(52)=1-A(52)
890  RETURN
900 Q$=INKEY$: IFQ$="" THEN 900 ELSE RETURN
910 IFO1>66 OR O2>66 THEN OP=1
920 PRINT "OPTION # ";GG;" IS ";R$;" BY ";ABS(O1);"%": O1=O2: RE
TURN
930 U$=INKEY$: IFU$="" THEN PRINT @0,"ONE OVERPRICED BY MORE THA
N 66%.": GOTO 940 ELSE IF CS=2 THEN H=A(101): L=A(102): A=P: GOT
O 5100 ELSE 40
940 PRINT "SOMEBODY THINKS THEY KNOW SOMETHING...TAKEOVER?????"
950 PRINT "SPREAD OPPORTUNITY, OR PERHAPS AN EXCELLENT COVERED W
RITE???"
960 PRINT "TO CONTINUE TOUCH ANY KEY": GOTO 930
970 IFM>3 THEN M=3: N=3: T=M/12: Y$="1ST": II=1: GOTO 380  ELSE
N=3: U=N/12: Y$="2ND": II=1: GOTO 380
980 IF CS=1 THEN  II=0: PRINT @832,"WITH 90 DAYS LEFT THE ";Y$;"
 OPTION SHOULD BE WORTH "; ELSE 790
990 IFY$="1ST" THEN PRINT USING T$;A(103) : GOTO  790  ELSE PRIN
T  A(104): GOTO 790
5000 REM
          LINE 5000 TO 5260 CAN BE A STAND ALONE PROGRAM FOR
          EVALUATING OUTRIGHT PURCHASES VS VERTICAL SPREADS
          IN CALLS  PROFIT/LOSS WILL LESS FAVORABLE AS

         COMMISSIONS HAVE BEEN OMITTED
5010 REM
          THE CALCULATIONS ASSUME AN EQUAL INVESTMENT IN
          BOTH THE OUTRIGHT AND SPREAD POSITION.  IN
          PRACTICE THIS MAY NOT BE POSSIBLE BUT IS SHOWN
          FOR ILLUSTRATIVE PURPOSES.
5020 CLEAR: CLS
5030 PRINT "COMPARISON OF OUTRIGHT PURCHASE VS VERTICAL SPREAD"
5040 PRINT "EG.  BUY A 30 CALL AND SELL A 35
5050 PRINT : INPUT "PRICE OF STOCK IN DECIMAL";A
5060 INPUT "PRICE IN DECIMAL OF HIGHER PRICED OPTION";HP
5070 INPUT "STRIKING PRICE";H
5080 INPUT "PRICE IN DECIMAL OF LOWER PRICED OPTION";LP
5090 INPUT "STRIKING PRICE";L
5100 S3=HP-LP
5110 U=HP-LP: U=HP/U
5120 CLS: INPUT "WANT PRINT OUT OF OUTRIGHT VS VERTICAL ENTER A
1";PO
5130 CLS: PRINT "PROFIT/LOSS AT EXPIRATION. STOCK PRICE ";A;"INV
ESTMENT $";HP*100: PRINT "THE CALCULATIONS ASSUME AN EQUAL INVES
TMENT IN EACH STRATEGY
5140 IFPO=1 THEN  LPRINT "THE CALCULATIONS ASSUME EQUAL DOLLAR I
NVESTMENT IN EACH.": LPRINT "PROFIT OR LOSS AT EXPIRATION.  CURR
ENT STOCK PRICE ";A;" INVESTMENT $";HP*100: LPRINT "."
5150  PRINT @129,"AT EXPIRATION     OUTRIGHT","SPREAD","% STOCK
GAIN
5160 IFPO=1 THEN  LPRINT "AT EXPIRATION        OUTRIGHT","SPREAD"
,"% STOCK GAIN
5170 FOR I=H TO H+11
5180 J=(I-A)/A*100
5190 GOSUB5240
5200 PRINT "$";I;" PER SHARE    ";INT(O*100),INT(P*U*100),INT(J)

5210 IFPO=1 THEN  LPRINT "$";I;" PER SHARE    ";INT(O*100),INT(P
*U*100),INT(J)
5220 NEXT
5230 INPUT "GO BACK TO MAIN PROGRAM=1 ANOTHER VERTICAL SPREAD=2"
;V: IFV=1 THEN  40 ELSE 5020
5240 O=Q-HP
5250 IFI<L+1 THEN  P=Q-S3
```

*Listing continues*

*Listing continued*

```
5260 Q=Q+1: RETURN
8000 REM
DATA STATEMENTS CONTAIN NUMBERS WHICH CORRESPOND TO THE
VOLATILITY OF THE STOCK.  THESE DO NOT REMAIN CONSTANT
FOR ALL STOCKS AS SOME INCREASE OR DECREASE WITH TIME
HENCE, THE NUMBERS MUST BE UPDATED OCCASIONALLY
8010 DATA AMF,45,ASA,51,ABT,40,AMD,55,AET,35,APD,32,ALD,38,AH,37
,AA,30,AMX,52,AMH,58,AHC,51,AMB,35,ABC,34,AC,35,ACY,35,AEP,5,AXP
,28,AHP,30,AHS,35,AMI,51,T,9,AMP,30,AAC,55
8020 DATA BUD,42,APA,60,ADZ,40,AS,35,AR,45,ASH,37,ARC,34,AUD,42,
AV,34,AVT,40,AVP,32
8030 DATA BKO,35,BDW,53,BLY,50,BAC,25,BCR,38,BOL,35,BAX,28,BRY,2
3,BDX,28,BS,33
8040 DATA BDK,30,BBL,30,BA,35,BCC,32,BMY,27,BFI,35,BC,38,BY,35,B
NI,33,BGH,29,CBS,28
8050 DATA CI,28,CSX,29,CCB,28,CAT,27,CZ,28,CEA,47,CHA,31,CHR,60,
CMB,29,CHU,32,CMZ,35
8060 DATA FNC,32,CS,40,CNV,33,CLX,31,CGP,57,KO,26,CL,27,COT,35,C
SP,33,CBU,58,CWE,5,CQ,32,CMY,37,CSC,50,CVN,50,ED,5,CTC,5,CDA,35,
CBE,30,GLW,29,CYR,50,ZB,32
8070 DATA DKI,5,DGN,47,DPT,50,DPC,49,DH,28,DE,28,DAL,31,DEN,34,D
IA,33,DEO,34,DEC,32,DIS,29,DGS,47,DGZ,47,DOW,31,DI,32,DD,29,DUK,
5
8080 DATA EGG,32,ESY,45,EFU,39,EK,27,ECK,29,ELG,33,EDS,34,EMR,26
,EC,38,ENS,32,ESM,31,XON,25
8090 DATA FDX,47,FCF,44,FLE,42,FGN,57,FLR,35,F,31,FWC,45,FTX,42,
FQA,40
8100 DATA GAF,42,GCA,55,GTE,23,GAO,43,GD,43,GE,24,GF,23,GRL,38,G
M,27,GPC,28,GX,49,GP,30,GET,33,GS,27,GLM,49,GT,27,GLD,31,GRA,28,
GWF,45,G,29,GW,31,GO,30
8110 DATA HAL,32,HRS,33,HPC,32,HBL,29,HWP,29,HLT,35,HIT,30,HIA,4
1,HM,49,HON,30,HCA,33,HI,27,HNG,31,HT,35,HUM,43,EFH,47
8120 DATA INZ,29,IBM,26,IFF,30,HR,38,IGL,29,IP,30,ITT,26,JNJ,25,
JOY,32,KM,31,KAB,43,KMG,33,KPH,55,LTV,50,LSI,34,LVI,32,LMK,44,LL
T,28,LIT,34,LK,54,LOR,35,LLX,33,LPX,35
8130 DATA MAI,46,MCA,31,MCZ,31,MAN,34,MCA,33,MHS,33,ML,31,MKY,47
,MAT,49,MDE,36,MCD,28,MD,36,MDT,36,MRK,27,MER,36,MSA,50,MSU,5,MM
M,26,MND,48
8140 DATA MOB,31,MDS,53,MTC,29,JPM,26,MOT,32,MUR,47,NCR,33,NL,35
,NLT,42,DR,28,NME,44,NSM,45,NOM,40,NEM,42,NBL,45,NSC,29,NOC,43,N
WA,35,NWT,31,NSI,28,NVO,49
8150 DATA OAK,55,OXY,33,ODR,43,OCF,31,OI,29,PPG,30,PDN,49,PKD,45
,PC,48,JCP,30,PZL,41,PEP,28,PKN,36,PTO,33,PFE,28,PD,42
8160 DATA PSB,43,MO,27,P,33,PBI,30,PCO,35,PPP,41,PRD,41,PRM,57,P
G,25
8170 DATA RCA,31,RAL,27,RYC,34,RTN,32,RB,48,RTA,58,REV,31,RJR,25
,RLM,31,ROK,33,RM,52,RDC,48,RD,26,RDR,35
8180 DATA SAB,44,SA,25,SFF,31,SGP,29,SLB,31,SFA,37,SPP,32,SRL,33
,S,27,SED,50,SUO,33,SGN,35,SMF,45,SKY,35,SII,35,SKB,29,SNE,34
8190 DATA SO,5,SX,32,SRO,58,LUV,43,SY,29,SQB,31,SD,30,SN,31,SOH,
35,STY,32,STK,52,SBK,35,SUN,33,SOC,39,SYS,32,SYN,33
8200 DATA TIE,63,TRW,28,TAN,40,TEK,32,TDY,35,TGT,29,TSO,50,TX,27
,TXN,30,TEI,58,TEZ,58,TXO,34,TDW,53,TGR,45,TL,31,TOS,55,TOY,48
8210 DATA TW,48,TA,28,TIC,30,TY,25,TYM,49
8220 DATA UAL,43,UK,26,UCL,33,UNP,33,U,50,UH,48,X,31,UTX,30,UPJ,
29,VLO,51,VIA,39,VEL,8
8230 DATA WMT,33,JWC,30,WAN,50,WCI,34,WLA,27,WEN,48,WSN,47,WU,33
,WX,30,WY,32,WKR,44,WMB,35,WMS,53,Z,30,XRX,28,ZOS,49,ZE,51
8240 DATA END,00
```

TDY goes on my books at 80 with a purchase date the same as the exercise date. If TDY declines below 80 (a ghastly thought), I am obliged to buy the stock at 80, but I was going to do that in the first place. In any case I'll be many thousands ahead by using the above strategy. Such put/call combinations are available on most optionable stocks.

> *"The key is knowing which options will outperform the market and which will be the laggards. I use a complicated system that ranks all options from one to five."*

There is one advanced strategy that has profited handsomely for five years and, you will find this hard to believe, it doesn't matter if the market goes up or down! Buy $10,000 worth of options on a group of stocks that are likely to outperform the averages and concurrently sell $10,000 worth of options on a group that is likely to underperform the market. If the market rises, the options you bought will rise faster than the options you sold. Should the market decline, all the options will probably expire worthless. By using the same strategy with puts, you can profit regardless of market direction.

This strategy is most useful if you have marginable securities in your account to meet the margin requirements for the naked options. The key is knowing which options will outperform the market and which will be the laggards. I use a complicated system that ranks all options from one to five. For five years it has not had a losing quarter. The options ranked one have *always* vastly outperformed those ranked five. The computations are updated weekly. If you want a printout, send an S.A.S.E. to MONEYDOS.

I have not covered all the strategies in trading options, but I hope you find MONEYDOS helpful. If you have suggestions for future columns write to me c/o *80 Micro*. ■

# COLOR WORKS


**NOW**—create color graphics and play color games on your TRS-80* MOD 1 & MOD 3. CHROMAtrs Kit **only $99****

**CHROMAtrs assembled & tested **$169**

# *WITH 15 VIVID COLORS*

**Now be first!** CHROMAtrs lets you create color graphics with capabilities never before possible on TRS-80*. You'll be able to play color games with such brilliant effects your friends will be green with envy — or any other color of your choice — and all this on your own TRS-80* MOD 1 & MOD 3!

**CHROMAtrs works with:**

- 15 vivid colors
- Transparent and high resolution graphics (192x256)
- Atari® joystick and paddle interface
- Audio and video outputs
- 3 Dimensional simulation through 35 prioritized display planes

**CHROMAtrs comes with:**

- Routines to provide high speed color graphics capability through Basic
- Operating manuals
- One complimentary game
- 16K RAM on board

The TRS-80* memory is not used and there is no modification necessary (warranty not voided). This uncomplicated system is so simple to connect you'll be amazing your friends with your color powers immediately.

## South Shore Computer Concepts, Inc.

Please send

- ☐ CHROMAtrs KIT ................ $99
- ☐ CHROMAtrs assembled & tested . $169
- ☐ TRS-80* MOD 1 cable ........... $12
- ☐ TRS-80* MOD 3 cable ........... $14
- ☐ Modulator with cables & switch box . $25

Sub Total ..... ______

N.Y.S. residents sales tax 7¼% . . . ______

Handling & shipping ............... $5

Total amount order ..............

*TRS-80* is a product of Tandy Corp.*

Check one:

Check or M.O. enclosed

______ Master Charge Visa

Account # expiration

Signature

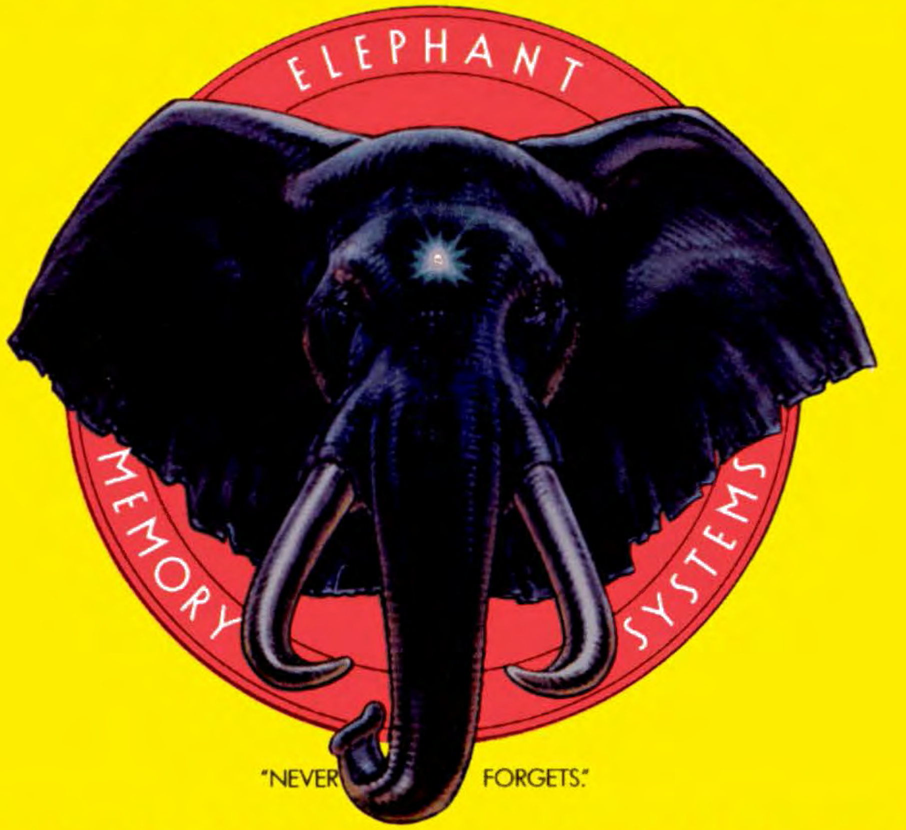
Name (Print)

Address

City State Zip

South Shore Computer Concepts, Inc. • 1590 Broadway, Hewlett, N.Y. 11557 • Phone Orders Taken (516) 569-4390 ✓171

# REMEMBER:




# MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

# ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us. 472

Distributed Exclusively by **Leading Edge Products, Inc.,** 225 Turnpike Street, Canton, Massachusetts 02021

Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.

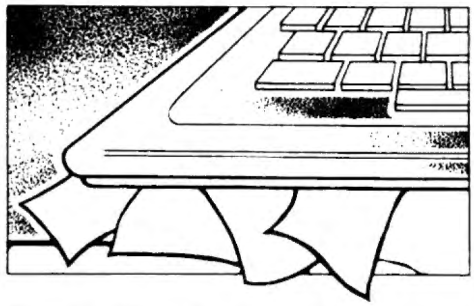
# 80 LIST OF ADVERTISERS


| RS Number | | Page |
|---|---|---|
| 286 | ABC Data Products | 350 |
| 476 | ABC Sales | 309 |
| 173 | ABS Suppliers | 409 |
| 38 | A M Electronics | 379 |
| * | AR Systems | 292 |
| 443 | A String Systems | 225 |
| 591 | ATC Software | 482 |
| 107 | Aardvark-80 | 239 |
| 564 | Abacus Associates | 478 |
| 26 | Access Unlimited | 330, 331 |
| 197 | Ace Computer Products of FL | 49 |
| 79 | Acorn Software | 255 |
| 536 | Adcock & Johnson | 156 |
| 562 | Advanced Systems Concepts Inc | 472 |
| 19 | Adventure International | 441 |
| 22 | Adventure International | 284 |
| 82 | Aerocomp Inc | 296, 297 |
| 183 | Alamo Computer Co | 327 |
| 75 | Alcor Systems | 347 |
| * | Aldevco | 138, 246, 469 |
| 136 | Algorix | 290 |
| 178 | All Systems Go | 413 |
| 473 | Alpha Byte Stores | 13 |
| 17 | Alpha Products Co | 14, 15, 17 |
| 69 | Alternate Source, The | 313 |
| 217 | American Small Business Computers | 87 |
| 218 | American Small Business Computers | 89 |
| 219 | American Small Business Computers | 91 |
| 220 | American Small Business Computers | 93 |
| 221 | American Small Business Computers | 95 |
| 559 | Animcalc | 482 |
| 141 | Anitek | 128 |
| 268 | Ann Arbor Precision | 188 |
| 40 | Apparal Inc | 351 |
| * | Applied Creative Technology Inc | 467 |
| 574 | Applied 80 Softwares | 470 |
| 6 | ARCsoft Publishers | 268 |
| 514 | Ashland Computer Systems | 241 |
| 65 | Aspen/Random House Co | 61 208 |
| 297 | Autel Electronic Company | 128 |
| 239 | Avalon Hill Game Company, The | 121 |
| 334 | BASF | 340 |
| 152 | BCcompco | 70 |
| 300 | B.T. Enterprises | 75, 245 |
| 74 | B.V. Engineering | 469 |
| 184 | Baudy House Computer Products | 58, 199 |
| 243 | Bay Technical Associates Inc. | 329 |
| 134 | Beard Software Development Co. | 143 |
| 387 | Better Software Associates | 358 |
| * | Bible Research Systems | 197 |
| 29 | Big Five Software | 420 |
| 106 | Binary Devices | 208 |
| 381 | Bodex | 225 |
| * | Bottom Line, The | 100 |
| 20 | Business Division, The | 338, 339 |
| 590 | Bytewriter | 472 |
| 181 | CMD Micro | 142 |
| 570 | CMD Micro Computer Services Ltd | 477 |
| 100 | CPU Shop, The | 223 |
| 138 | C&S Electronics Ltd | 234 |
| 64 | California Word Exchange | 335 |
| 466 | Check-Mate | 139 |
| 326 | Cheever Microware | 473 |
| 527 | Chicago Computer | 295 |
| 103 | Chromasette Magazine | 202 |
| 584 | Climate Assessment Technology Inc | 472 |
| 102 | Cload Magazine Inc | 203 |
| 121 | Cognitec | 55 |
| 576 | Colorquest | 474 |
| * | Color Software Services | 328 |
| 361 | Communications Systems Lab | 311 |
| 252 | CompuAdd Corp | 306 |
| 550 | CompuGift Inc | 470 |
| 455 | Compukit | 137 |
| 235 | CompuServe | 170, 171 |
| 120 | Computer Applications Unlimited | 155, 399 |
| 139 | Computer Case Company | 319 |
| 133 | Computer Discount of America | 281 |
| 568 | Computer Island | 477 |
| 420 | Computer Peripheral | 382 |
| 367 | Computer Place, The | 258 |
| 18 | Computer Plus | 19 |
| 257 | Computer Services of Danbury | 319 |
| 109 | Computer Shack | 129, 131 |
| 201 | Computer Shack | 133 |
| 453 | Computer Shack | 135 |
| 39 | Computer Shopper | 417 |
| 369 | Computerware | 310 |
| 27 | Computex | 274 275, 383 |
| 553 | Computex | 480 |
| 511 | Compuware Corp | 415 |
| 104 | ComSoft | 167 |
| 8 | Conex Electro System Inc | 32 |
| 68 | Contract Services Associates | 169 |
| 110 | Control Craft Inc | 391 466 |
| 105 | Coosol Inc | 385 |
| 52 | Cornsoft Group, The | 28 |
| 308 | Cornsoft Group, The | 29 |
| 319 | Cornsoft Group, The | 30 |
| 519 | Cornsoft Group, The | 31 |
| 45 | Cornucopia Software | 47 |
| 58 | Cosmopolitan Electronics Corp | 279 |
| 187 | Cottage Software | 365 |
| 493 | Crabbe Associates | 184 |
| 190 | Creative Computer | 162 |
| 55 | DA&D Sales Inc | 149 |
| 410 | D B Software | 301 |
| 517 | DCS Software | 271 |
| 413 | DLP Company | 409 |
| 506 | D/Punch Company | 409 |
| * | DSC Publishing | 318 |
| 452 | Data Mart | 165 |
| 118 | Data Services Inc | 278 |
| 99 | Data Systems | 316 |
| 253 | Datacom Computer Sales & Supplies | 409 |
| 394 | Decision Science Software | 208 |
| 580 | Demi Software | 470 |
| 403 | Digiforensics | 473 |
| 37 | DISKovery Unlimited | 463 |
| 5 | Diskette Junction | 59 |
| * | Disk Supply | 282 |
| 82 | Displayed Video | 66 67 374 375 |
| * | Don Fielding | 198 |
| 353 | Dresselhaus Computer Products | 362 |
| 216 | EAP Company | 401 |
| 271 | EBG & Associates | 302 |
| 401 | E & G Umior Enterprises | 292 |
| 28 | EPYX/Automated Simulations | 355 |
| 49 | ESI Lynx | 175 |
| 560 | EZR Micro Inc | 478 |
| 513 | East Coast Data | 415 |
| 545 | Ebert Personal Computers | 455 |
| * | Eighty Micro | 50, 163, 372 373 415 467 469 481 |
| * | Eighty-Micro Anniversary Issue | 35 |
| * | Eighty Micro Back Issues | 368 |
| * | Eighty Micro Bookshelf | 393-395 |
| * | Eighty Micro Christmas | 323 |
| * | Eighty Micro Subscription | 305 |
| * | Eighty Micro Subscription Problems | 469 |
| 329 | Eighty US Journal | 134 |
| 318 | Electronic Protection Devices | 324 |
| 159 | Electronic Specialists | 345 |
| * | Encyclopedia | 226 227 |
| 283 | Endicott Software | 469 |
| 97 | Epson America Inc | 144 145 |
| 177 | Equine Management Innquations | 65 |
| 93 | Exatron | 38 |
| * | Excelonix | 304 |
| 166 | FGA Software | 45 |
| 36 | Fantastic Software | 289 |
| 214 | Fort Worth Computers | 385 |
| 262 | Frank Hogg Laboratory Inc. | 321 |
| 442 | Freedom Financial Enterprises of WA Inc | 283 |
| 567 | Freedom Financial Enterprises of WA Inc | 477 |
| 360 | Freedom Micro-Systems Inc. | 361 |
| 391 | Funsoft Inc. | 124 |
| 66 | Futuresoft | 97 |
| 492 | GSB Electronics | 208 |
| 186 | Good-Lyddon Data Systems | 301 |
| 9 | H & E Computronics Inc | 425-432 |
| 409 | H F Signalling Inc | 469 |
| 487 | HPB Vector Co | 315, 409 |
| 244 | Hacks | 238 |
| 47 | Heath/Zenith Company | 164 |
| 46 | Hexagon Systems | 270 |
| 365 | Holman D P Services | 359 |
| 153 | Holmes Engineering Inc | 325 |
| 320 | Howard Bowe | 359 |
| 552 | Howard W Sams & Company Inc | 476 |
| 175 | Howe Software | 160 |
| 185 | Hurricane Laboratories | 147 |
| 211 | ICR Futuresoft | 98 |
| * | IJG | 267 389, 435 |
| * | InCider | 99 |
| 589 | Indecks Inc | 480 |
| 494 | Independent Software | 409 |
| 424 | Individual Systems Inc | 446 |
| 530 | Inflo Inc | 423 |
| 566 | Inflo Inc | 470 |
| * | Infocom | 280, 281 |
| * | Instant Software Inc. | 436 |
| 351 | Institute for Scientific Analysis | 294 |
| 357 | Integrated Devices Inc. | 214 |
| 565 | Interface Inc | 478 |
| * | International Computer Products | 184 |
| 577 | International Computer Products | 480 |
| 516 | International Software Alliance | 113 |
| 336 | Intracolor Communications | 59 |
| 491 | J Eli & Associates | 415 |
| 202 | JES Graphics | 196 |
| 273 | J F Consulting | 23 |
| 101 | J&M Systems | 377 |
| 42 | JPC Products Company | 439 |
| 254 | Jimscot Inc. | 204 |
| 256 | Joe Computer | 111 |
| 515 | John Wiley & Sons Inc | 295 |
| 347 | Jonell Systems | 469 |
| 485 | Kalglo Electronics Company Inc | 197 |
| 323 | Katahdin Software | 409 |
| * | Kengore Corp | 251 |
| 558 | Kensoft | 482 |
| 586 | Kraft Systems Company | 478 |
| 51 | Krell Software Corp | 196 |
| 331 | KSoft | 272 |
| 358 | Kwik Software | 293 |
| 444 | L J Kutten | 469 |
| 30 | LNW Research Corp | CIV |
| 31 | LNW Research Corp | 7 |
| 32 | LNW Research Corp | 151 |
| 33 | LNW Research Corp | 143 |
| 462 | LSI Systems | 27 |
| 392 | Lawrence Epstein Associates | 127 |
| 191 | Lawyer's Microcomputer, The | 136 |
| 472 | Leading Edge Products Inc | 450 |
| 402 | Learning Tools Inc. | 141 |
| 350 | Ledford Computer Services | 250 |
| 155 | LEDS Publishing Company Inc | 270 |
| 358 | Lemons Tech Services | 293 |
| * | Level IV Products Inc | 112, 301 |
| 131 | Libra Laboratories Inc | 231 |
| 135 | Lindbergh Systems | 46 |
| 575 | M-P Software | 482 |
| 90 | MWB Industries Inc | 241 |
| 196 | Maine Software Library, The | 212 |
| 457 | Management Systems Software | 401 |
| 259 | Manhattan Software Inc | 113 |
| 588 | Manhattan Software Inc | 476 |
| 92 | Maple Leaf Systems | 409 |
| 241 | Mark Data Products | 122 |
| 146 | Master Electronics Inc | 271 |
| 469 | Maximum Micro | 386 |
| 83 | Med Systems Software | 42, 57, 103 |
| 250 | Melbourne House Software Inc | 123 |
| 461 | Memory Merchant | 447 |
| 524 | Menlo Systems | 359 |
| 188 | Mercer Systems Inc | 140 |
| 507 | Meridian | 307 |
| 13 | Meta Technologies Corp | 9 11 |
| 149 | Micro Architect | 383 |
| 247 | Microcomputer Technology Inc | CII |
| 459 | Micro Data Supplies | 108 |
| 76 | Micro-Design Inc | 407 |
| 77 | Micro-Design Inc | 407 |
| 78 | Micro-Design Inc | 407 |
| 463 | Micro-Design Inc | 232 |
| 132 | Micro-80 Inc | 158 378 |
| 583 | Micro-80 Inc | 482 |
| 161 | Micro-Grip Ltd | 254 |
| 419 | Microhatch | 401 |
| 157 | Micro Images | 221 |
| 464 | Micro Labs | 233 |
| 465 | Micro Labs | 233 |
| 60 | Micro Mainframe | 397 |
| 96 | Micro Management Systems | 367 |
| 342 | Micromatic Programming Co | 381 |
| 162 | Micro Mega | 311 |
| 117 | Micromint Inc | 197 |
| 270 | Micro Moonlighter Newsletter | 113 |
| 578 | Micro-Peripheral Corp | 476 |
| 546 | Micro Projects Engineering | 229 |
| 312 | Microsette | 248 |
| 282 | Microsoft CPD | 287 |
| * | Micro Solutions Inc | 180, 380 |
| 384 | Micro Systems Software Inc | 348, 471 |
| 339 | Micro-Tax | 136 |
| 380 | Micro Tech Exports Inc | 364 |
| 581 | Micro Tech Exports Inc | 474 |
| 555 | Microworks, The | 474 |
| 125 | Microworks, The | 293 |
| 72 | Mikee Electronics Corp | 328 |
| 137 | Miller Microcomputer Services | 482 |
| 123 | Misosys | 352 466 |
| 67 | Mumford Micro Systems | 411 |
| 539 | N C Software Inc | 415 |
| 533 | NEBS | 381 |
| 451 | NISS | 51 |
| 571 | NTS Software | 480 |
| 572 | Nanon Systems Corp | 470 |
| 128 | Nelson Software Systems Inc | 119 357 |
| 496 | Nepenthe Programs | 246 |
| 255 | New Classics Software | 247 |
| 232 | Nocona Electronics | 10 243 |
| 525 | Ocean Computer Products | 301 |
| 582 | Office of the Australian Trade Commissioner | 476 |
| 50 | Omikron | 52 |
| 195 | Omnitek Computers International Inc | 272 |
| 206 | Options-80 | 138 |
| 151 | Orion Instruments | 318 |
| 25 | PC Newsletter | 368 |
| 433 | PCD Systems | 337 |
| 242 | PMC Software | 249 |
| 321 | PAB Software Inc | 391 |
| 207 | Pacific Exchanges | 138, 292, 294, 336, 401 |
| 122 | Pan American Electronics | 349 |
| 333 | Peggytronics | 160 |
| 324 | Pel/Tek | 473 |
| * | Percom Data Company | 3, 24 |
| 1 | Percom Data Company | 25 |
| 2 | Percom Data Company | 25 |
| 3 | Percom Data Company | 25 |
| 272 | Peripherals Unlimited | 444 |
| 124 | Perry Oil & Gas Inc | 359 |
| 557 | Personal Computer Owners | 474 |
| * | Phase I Systems Inc | 457 |
| 490 | Pion Inc | 159 |
| 160 | Pioneer Software | 240 |
| 377 | Plug Bug Systems | 392 |
| 73 | Plus Computer Technology | 353 |
| 315 | Powersoft | 205 |
| 378 | Powersoft | 209 |
| 61 | Powersoft | 207, 471 |
| 11 | Practical Peripherals | CIII |
| 63 | Precision Prototypes | 73 |
| 477 | Priority One Electronics | 253 |
| 260 | Pro/Am Software | 241 |
| 288 | Process Control Technology | 241 |
| 80 | Program Store, The | 114-117 |
| 245 | Programmer's Institute | 224 |
| 475 | Programmer's Institute | 225 |
| 418 | Programs for the 80's | 310 |
| 143 | Progressive Electronics | 316 |
| 91 | Prosoft | 21, 335, 479 |
| 48 | Quality Computer Services | 443 |
| 34 | Quality Software & Consulting Inc | 65 |
| 194 | Quant Systems | 142 |
| * | R N da Silva | 319 |
| 41 | Racet Computes Ltd | 405 |
| 468 | Racet Computes Ltd | 405 |
| 269 | Radix Technologies | 60 |
| 264 | Rainbow P & P Corp | 59 |
| 265 | Rainbow P & P Corp | 143 |
| 460 | Rainbow Software Services | 178 |
| 296 | Rainbow, The | 315 |
| 130 | Rand's | 244 |
| 345 | Real Software | 198 469 |
| * | Realty Software Company | 401 |
| 554 | Realty Software Company | 480 |
| 275 | Red Baron Computers | 371 |
| 7 | Redgate Publishing Company | 78 79 |
| 498 | Red Wing Software | 415 |
| 114 | Retware | 242 |
| 579 | Retware David C Whitney Assoc Inc | 478 |
| 145 | REMarkable Software | 126 |
| 129 | REMsoft Inc | 294 |
| 573 | Repair Service Inc | 476 |
| 303 | Richcraft Engineering Ltd | 415 |
| 480 | Rimes Computer Products | 77 |
| 512 | Rockware Data Corp | 391 |
| 343 | Rogo Computer Products | 415 |
| 213 | 68 Micro Journal | 401 |
| 35 | SIO | 369 |
| 499 | S M S Products | 391 |
| 266 | STSC Inc | 453 |
| 142 | Sales Data Inc | 149 |
| * | Sandpiper Software | 319 |
| 203 | Scientific Engineering Lab | 401 |
| 510 | Scott Tasso | 401 |
| 372 | SemiDisk Systems | 43 |
| 474 | Seneca Electronics | 346 |
| 526 | Serious Personal Computer | 246 |
| 363 | Shawmut Systems | 349 |
| 495 | Shelboy Enterprises | 250 |
| 301 | SimSim Inc | 45 448 |
| 12 | Simutek | 303 |
| 164 | Simutek | 459 |
| 163 | Simutek | 179 181 183 |
| 458 | Single Source Solution | 269 |
| 587 | Small People Software Corp | 482 |
| * | Snappware | 211, 213 215, 217, 219 |
| 399 | Soft Images/Decision Systems | 105 |
| 527 | Soft Images/Decision Systems | 107 |
| 84 | Soft Sector Marketing | 185 |
| 85 | Soft Sector Marketing | 187 |
| 86 | Soft Sector Marketing | 189 |
| 87 | Soft Sector Marketing | 191 |
| 88 | Soft Sector Marketing | 193 |
| 89 | Soft Sector Marketing | 195 |
| 379 | Softcore Software Co, The | 85 |
| 43 | Software Concepts | 157 |
| 170 | Software Models | 314 |
| 537 | Software Options | 273 |
| 327 | Software Support | 33 |
| 322 | Solectronics | 369 |
| 171 | South Shore Computers | 449 |
| 119 | Spectral Associates | 363 365 |
| 569 | Spectrum Projects | 482 |
| 261 | Standard and Poor's Corp | 83 |
| 205 | Starbuck Data Co | 336 |
| 432 | Star Kits | 356 |
| 346 | Star Value Software | 365 |
| 71 | Sublogic Communications Corp | 403 |
| 285 | Sun Research Inc | 329 |
| 456 | Sunlock Systems Products | 265 |
| 94 | Swallow Software | 111 |
| 585 | Sybex | 470 |
| * | Synergistic Solar Inc | 336 |
| 174 | Syracuse R & D Center | 401 |
| 408 | T & D Software | 149 |
| 389 | T C E Programs | 225 |
| 449 | TSASA Inc | 111 |
| 189 | Tab Sales | 377 |
| 70 | Taranto & Associates Inc | 69, 71 |
| 440 | Tech Data Corp | 37, 184, 409 |
| 504 | Telexpress Inc. | 257 |
| 59 | Texas Computer Systems | 317 |
| 371 | Think Software | 45 |
| 144 | Thomas V Lenz | 304 |
| 338 | Tidewater Data Systems | 473 |
| 81 | Total Access | 332 |
| 44 | Transaction Storage Systems | 299 |
| 258 | Trend Software Company | 383 |
| 489 | Triple-D Software | 369 |
| 332 | United Software Associates | 111 |
| 551 | Universal Data Research Inc | 478 |
| 417 | Universal Data Research Inc | 415 |
| 509 | Usage Software Group | 265 |
| * | VR Data Corp | 475 |
| 169 | Van Horn Office Supply | 237 |
| 523 | Verbatim | 172 |
| 518 | Vespa Computer Outlet | 235 |
| 549 | Volks Microcomputer Systems | 127 |
| 556 | Wayne Green Books | 465 |
| * | Wayne Green Books | 266, 227, 259, 387, 465 |
| * | Wayne Green Inc | 467 |
| 179 | Western Operations | 212 |
| 212 | William A Fink | 342 |
| 192 | Wilson Software Division | 456 |
| 488 | Winn & Harrison | 208 |
| 158 | XYZT Computer Dimensions Inc | 386 |
| 563 | XYZT Computer Dimensions Inc | 482 |
| 156 | York 10 Computerware | 161 |
| 172 | Young Ideas Inc | 127, 369 |


**For further information from our advertisers, please use the Reader Service card.**

# NOTES FROM BENEATH the KEYBOARD

by Paul Wiener



## Disk I/O with V. Hester

This month's "Notes from Beneath the Keyboard" is special. Kim Watt once told me what a luxury it was to work with Vernon Hester. He said you didn't need a disassembler with Vernon around, because Vernon could read a page of op-codes as easily as mnemonics. He also said, "Vernon Hester is the king of TRS-80 disk I/O. I know some people have attributed that honor to me, but Vernon's the real one."

If you use any of Vernon Hester's software, you probably share my belief that Vernon is one of the grandmasters of TRS-80 disk I/O. This month, I'm broadcasting the voice of the master. I think you'll agree that Vernon's programming perspicacity rivals his teaching talent. Vernon will tell us a hitherto unpublished method of disk I/O. At first glance, the technique might seem similar to other one-byte I/O techniques. However, Vernon's offering is unique because it uses ROM calls to eliminate the greatest stumbling block to compatibility with a variety of DOSes.

**Vernon:** A lot of people have learned how to read and write sectors from disk. There has been a problem writing the last sector, or end-of-file (EOF) byte.

That gets tricky when you're working with a multitude of DOSes. What no one has detailed is exactly how one who doesn't want to get too involved in a DOS can write a simple routine that will correctly write a file, byte for byte, from RAM to disk, or read a file from disk to RAM, with the proper end of file, and with the proper end byte.

Those calls are available in the ROM. If you make those calls in the ROM, the particular DOS being used will do its own end-of-file and next-byte manipulation, and correctly write the proper end of file and end-of-file byte.

**Paul:** When you talk about end of file, and end-of-file byte, you're referring to potential compatibility problems because of the way in which different DOSes handle those things, right?

**Vernon:** Right.

**Paul:** Now, are those the major or only incompatibility problems? If someone's application program has disk I/O and those two things come out right, is it fairly certain everything else will?

**Vernon:** Yes. What a lot of people have probably confused a one-byte read or a one-byte write with is a logical record length of one. The programmer still has control over what his logical record length is. Unless you have a specific need, such as random files from Basic, I don't see why you shouldn't have a logical record length of (256(?)). But other than that, most programs work on a logical record length of 256.

**Paul:** Does a logical record length of 256 ensure that a 256-byte buffer is filled before a sector is actually flushed to disk?

**Vernon:** Yes.

**Paul:** So that keeps things relatively quick.

**Vernon:** Yes.

**Paul:** Some people are concerned that this type of byte I/O is slow. You said once before that it does use system-verify automatically, so writes can take a little longer. But otherwise, it's about the same as any technique that writes a sector at a time. Is that correct?

**Vernon:** Yes. It depends on what you're trying to read or write. If you're trying to read or write a file that fits in RAM, it's not going to take much time. Especially reading it, because it only reads and writes full sectors; it doesn't read or write partial sectors.

In other words, the way DOSes work, you can only read in a sector. What DOS does, in a one-byte read, is to read an entire sector the first time it accesses that sector. The accumulator will contain the sector's first byte. Then, if you call the one-byte read again, it doesn't read the sector again, it positions the buffer pointer one byte up. Once that sector is finished being read in, DOS will read another sector.

**Paul:** That's done automatically with the same ROM call. When you do the call, you don't have to know if the buffer pointer is in the middle of a sector or the end of a sector. You do it the same way?

**Vernon:** You do it the same way. The DOS handles it. Every time you read a byte, although the accumulator contains that byte, it also contains a zero-flag condition. If the zero flag is not set by a read, you jump to a little routine I call End of File. If the accumulator contains a 1C hex, it's read in everything and you are at the end of file; otherwise, you have an error.

What you do is call the one-byte read. You jump-relative if the byte read is nonzero to your get-error routine. Otherwise, you move the byte in the accumulator to wherever in memory you're reading. For example, I point the HL register at the memory I want to read. Then I load the target of the HL register (the location where the HL is pointing) from the accumulator, increment the HL, and jump right back to the one-byte read routine. When I end up with an error, I compare with 1C hex. If you have a 1C hex, you have read in what you want to read in. Otherwise you can set bit 6 and jump to 4409 hex. That's a DOS error-handling routine. The only reason you set bit 6 is so you won't see DOS ERROR = error number.

**Paul:** Do you get the DOS error message at all then?

**Vernon:** If you jump to 4409 you will.

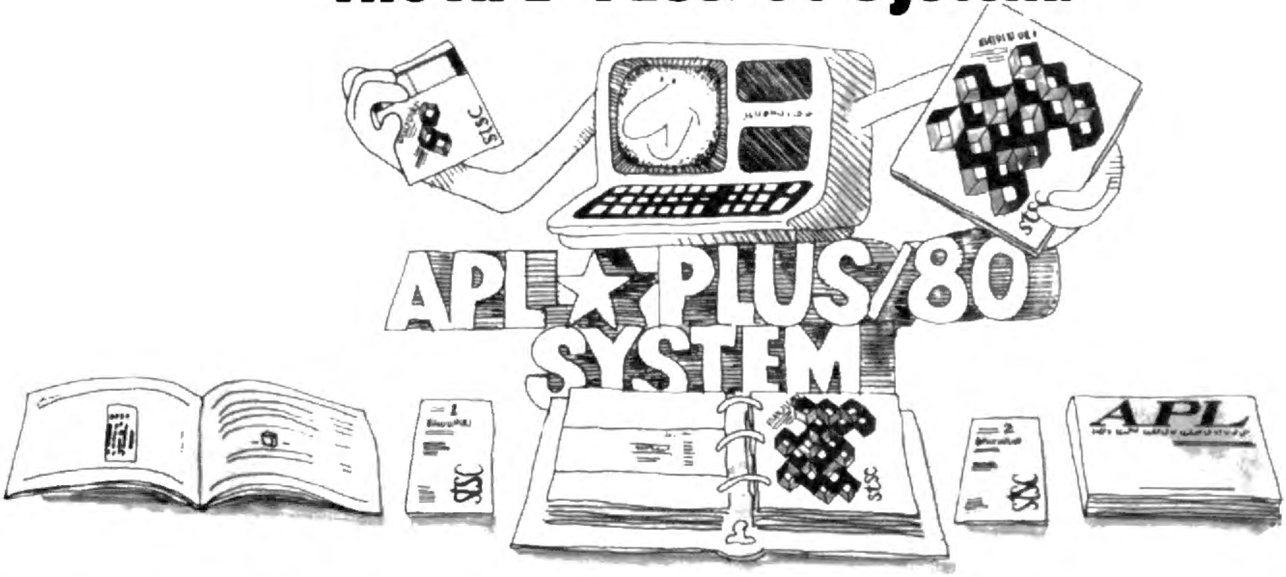
**Paul:** With bit 6 set you get an error message but not the error number?

**Vernon:** Right. If you don't have bit 6 set, you'll get DOS ERROR = 24. Most DOSes, when you're in the DOS ready mode and you input an illegal or bad file name before it jumps to 4409, will set bit 6. To set bit 6 I do an OR 40H. It's a little faster.

I've written a routine that will both read and write a user program, text, or whatever.

At line 14000 you have a label called Start: Define label FBUF plus 20H. It's the first byte to write. Model III TRSDOS has a 50-decimal-byte DCB. In

# A Practical and Educational Gift--
# The APL★PLUS®/80 System.




**All you get when you buy some software systems is a thin manual and a handshake for good luck. STSC's APL★PLUS/80 System is different. We give you a set of documentation that's so complete you won't need to read another thing to develop and run solutions on your TRS-80® Model III.**

We know that our APL★PLUS/80 System can improve your productivity–if you know how to use its power. We start you off with a beginner's operation guide, if you're new to APL and to the TRS-80, and a beginner's tutorial, if you're familiar with the TRS-80 but not with the language. Then we include the most widely used APL textbook, so that once you get a taste for APL, you can study it in detail.

### Challenge to BASIC

Build a data file with 15 records of 21 random numbers between 1 and 100 chosen without duplicates (in a record) and arranged in tables with 3 rows and 7 columns. Write a subroutine that reads a specified record and prints the table with row and column totals in fields 6 columns wide. Here is an APL solution:

```
    ∇ RANDOMTABLES;I
[1]   'RTABLES' ⎕FCREATE 10 ⋄ I←1
[2]  L:(3 7⍴21?100)⎕FAPPEND 10 ⋄ →(15≥I←I+1)⍴L
    ∇
    ∇ SHOWTABLE N;M
[1]   'I6' ⎕FMT M,[1]+/[1]M←M,+/M←⎕FREAD 10,N
    ∇
      RANDOMTABLES ⋄ SHOWTABLE 12
    89     1    45    30    84    50   100   399
    52     8    93    13    99    40    77   382
    14    69    29    96     3    44    68   323
   155    78   167   139   186   134   245  1104
```

Here's your complete documentation package:

- Introduction to the APL★PLUS/80 System.
- Product Overview: APL★PLUS Systems.
- *APL★PLUS/80 Computer Operation User's Guide.*
- *APL is Easy!*
- *APL: An Interactive Approach,* Leonard Gilman and Allen J. Rose (Wiley, 1974).
- *APL★PLUS/80 Formatting User's Guide.*
- *APL★PLUS/80 Shared Files User's Guide.*
- *APL★PLUS/80 Programmer's Reference Manual.*
- Two APL★PLUS/80 Reference Cards.

We don't want to scare you away with our extensive documentation. You don't need to read all the manuals to write useful applications in APL, because APL is easy. APL is also fun and we're sure that when you know a little APL, you'll want to know a whole lot more–and *that's* why we're giving you the most complete set of documentation on the market.

If you'd like to use APL on your TRS-80 Model III to develop and deliver solutions faster and better for yourself or for your clients, send for the APL★PLUS/80 Application Development System. Mail in the coupon with your payment and we'll send you everything you need to run APL on your TRS-80 right away. If you'd like more details, check the box on the coupon and we'll send you our free information package which includes a detailed explanation of the Challenge to BASIC.

We're STSC, Inc., the leading supplier of APL software and services in the United States. Our APL★PLUS systems have been serving the business and professional world for more than 12 years.

APL★PLUS/80 runs under TRSDOS® 1.3 or LDOS® 5.1 on a 48K RAM TRS-80 Model III with two disk drives. The APL★PLUS/80 System comes with a custom APL-character ROM and a self-adhesive keyboard label set to convert your TRS-80 to the APL character set.

Attn: APL★PLUS/80 Distribution
**STSC, Inc.,** 2115 East Jefferson Street ✓266
Rockville, Maryland 20852 (301) 984-5000 (orders only)

**Yes, send me the APL★PLUS/80 System, postpaid.**

☐ My check for $295 is enclosed. *Add sales tax where applicable.
☐ Charge my MasterCard Account #
Bank #
☐ Charge my VISA Account #
Expiration date:
*State sales tax applicable in MD (5%), FL (4%), and CT (7½%).

**I'd like to know more about the APL★PLUS/80 System.**

☐ Send me your free information package.

Name
Address
City State Zip
Phone ( )

APL★PLUS is a service mark and trademark of STSC, Inc., registered in the United States Patent and Trademark Office. 80-1282
TRS-80 and TRSDOS are registered trademarks of Tandy Corporation. LDOS is a registered trademark of Logical Systems Corporation

that case you define that FBUF plus 32H.
**Paul:** So, if you're not pressed to save a few bytes and you're not sure what DOS is going to be used, you can allow for a 50-byte DCB buffer and preserve all-around compatibility, right?
**Vernon:** Right. At line 17000, we define the label Mode. Mode will tell you whether you want to read or write. When you enter this program, you jump to or call the label Disk. Before you do that, Mode must contain the proper value to specify a read or write. A few other locations must be correctly loaded as well. Start and Finish (lines 14000 and 15000) must be loaded for a write, and Where (line 16000) must be loaded for a read. The name of the file to read or write must be set up in memory. I'll get to all that as we follow the program flow.

Line 18000 defines your I/O buffer. I arbitrarily used 5300H as one page above the file DCB buffer (the label FBUF, line 13000).

At line 19000, I set the origin up. You have plenty of room here. The label I have is Disk. Load DE with FBUF. That points DE at the DCB. Then we call FILK. That checks for a valid file name.
**Paul:** Does the user have the file name in a buffer when this routine starts?
**Vernon:** The file name can be anywhere in memory when he enters the routine. The HL had to be pointing at the file name's first character. The file name has to be terminated by 03 or 0DH. If HL contains 6000H, the name should be there (at 6000H) terminated by 03 or 0DH.
**Paul:** And that's what's getting checked when you call FILK?
**Vernon:** Right. In other words, if HL is pointing to 1234567, that's an invalid file name. There would be a nonzero condition on return from FILK.

The next instruction is jump-relative on nonzero to BADNAM. The comment there is "nonzero if bad name."

The next instruction is load HL with the I/O buffer. That's where to read or write sectors from or to. I set that to 5300H.

Line 24000 is load B with zero. That sets a logical record length (LRL) of 256.
**Paul:** What do you use for logical record lengths other than 256?
**Vernon:** Whatever you want to use. Load B with 24 and you'll have a logical record length of 24.
**Paul:** And loading B with one would give you a logical record length of one?
**Vernon:** Yes.
**Paul:** And 255 would be 255?
**Vernon:** Right. There's a gotcha on LRLs other than 256, though.
**Paul:** What's that?
**Vernon:** You need two buffers.
**Paul:** Maybe for simplicity, you should keep it as 256 then.
**Vernon:** Okay. Load A with the contents of Mode to determine if you're going to read or write. If it's a zero, you want to write. If it's a nonzero, you want to read. The way you check that is with the next instruction at 26000 and 27000: Or A. Jump-relative to RDISK if A is nonzero.

If you get to line 28000, you know you're going to write a file, so we call INIT. That will initialize a (new) file, or overwrite an existing one (with the same file name). This is what the HL is originally pointing to—this text (file name). You can have whatever valid file name you want. You can have file name, slash, extension, decimal point, password, colon, and drive number if you wish. And if you don't have a drive number, it's going to search for the nonwrite-protected disks, starting at drive zero.

The next instruction (at line 30000) is "jump-relative to UERR if nonzero." That occurs if there is an error.

The next instruction is load HL with the contents of Start. That's the first

```
                00000 ;RWD
0013            01000 ROMRD   DEFL    13H             ;ROM INPUT ONE BYTE
                02000                                 ;FROM DEVICE
001B            03000 ROMWT   DEFL    1BH             ;ROM OUTPUT ONE BYTE
                04000                                 ;TO DEVICE
402D            05000 DOS     DEFL    402DH           ;ENTRY TO 'DOS READY'
4409            06000 ERROR   DEFL    4409H           ;ERROR INTERPRETER
                07000                                 ;A CONTAINS ERROR CODE
441C            08000 FILK    DEFL    441CH           ;MOVE FILE FROM @ (HL) TO
                09000                                 ;@ (DE), CHECK FOR VALIDITY
4420            10000 INIT    DEFL    4420H           ;INITIALIZE FILESPEC
4424            11000 OPEN    DEFL    4424H           ;OPEN EXISTING FILESPEC
4428            12000 CLOSE   DEFL    4428H           ;CLOSE FILE
5200            13000 FBUF    DEFL    5200H           ;FILE DCB
5232            14000 START   DEFL    FBUF+32H        ;FIRST BYTE TO WRITE
5234            15000 FINISH  DEFL    START+2         ;LAST BYTE TO WRITE
5236            16000 WHERE   DEFL    FINISH+2        ;WHERE TO READ INTO
5238            17000 MODE    DEFL    WHERE+2         ;NON-ZERO = READ
5300            18000 IOBUF   DEFL    5300H           ;SECTOR BUFFER
5239            19000         ORG     MODE+1
5239 110052     20000 DISK    LD      DE,FBUF         ;PUT DE @ DCB
523C CD1C44     21000         CALL    FILK            ;CHECK FOR VALID FILENAME
523F 2027       22000         JR      NZ,BADNAM       ;NON-ZERO IF BAD NAME
5241 210053     23000         LD      HL,IOBUF        ;WHERE TO READ/WRITE SECTORS

5244 0600       24000         LD      B,0             ;LRL=256
5246 3A3852     25000         LD      A,(MODE)        ;WHAT TYPE
5249 B7         26000         OR      A
524A 2023       27000         JR      NZ,RDISK        ;NON-ZERO = READ
524C CD2044     28000         CALL    INIT            ;INIT FILE
                29000                                 ;OVERWRITE IF EXISTING
524F 2019       30000         JR      NZ,UERR         ;IF NON-ZERO SHOW ERROR
5251 2A3252     31000         LD      HL,(START)      ;FIRST BYTE TO WRITE
5254 ED4B3452   32000         LD      BC,(FINISH)     ;LAST BYTE TO WRITE
5258 7E         33000 MORE    LD      A,(HL)          ;GET BYTE
5259 CD1B00     34000         CALL    ROMWT           ;PUT IT INTO BUFFER
                35000                                 ;DOS WILL WRITE SECTOR
                36000                                 ;TO DISK WHEN FULL.
525C 200C       37000         JR      NZ,UERR         ;IF NON-ZERO SHOW ERROR
525E E5         38000         PUSH    HL              ;SAVE POSITION
525F ED42       39000         SBC     HL,BC           ;ARE WE DONE?
5261 E1         40000         POP     HL              ;RESTORE POSITION
5262 23         41000         INC     HL              ;INCREMENT ONE POSITION
                42000                                 ;INC HL DOES NOT EFFECT
                43000                                 ;"Z" FLAG!
5263 20F3       44000         JR      NZ,MORE         ;IF NON-ZERO MORE TO WRITE
5265 C32844     45000         JP      CLOSE           ;CLOSE FILE
5268 3E13       46000 BADNAM  LD      A,13H           ;13H = IMPROPER FILE NAME
526A F640       47000 UERR    OR      40H             ;MASK TO PROHIBIT
                48000                                 ;'DOS ERROR = XX'
526C C30944     49000         JP      ERROR           ;GET ERROR MESSAGE
526F CD2444     50000 RDISK   CALL    OPEN            ;OPEN EXISTING FILE
5272 20F6       51000         JR      NZ,UERR         ;IF NON-ZERO SHOW ERROR
5274 2A3652     52000         LD      HL,(WHERE)      ;GET PLACE TO PUT IT
5277 CD1300     53000 GDISK   CALL    ROMRD           ;GET BYTE
527A 2004       54000         JR      NZ,EOF          ;IF END OR ERROR = NON-ZERO
527C 77         55000         LD      (HL),A          ;PUT THE BYTE
                56000                                 ;WHERE YOU WANT IT
527D 23         57000         INC     HL              ;INCREMENT POSITION
527E 18F7       58000         JR      GDISK           ;GET MORE
5280 FE1C       59000 EOF     CP      1CH             ;IF 1CH NATURAL EOF
5282 C8         60000         RET     Z               ;GO HOME IF OK
5283 18E5       61000         JR      UERR            ;SHOW ERROR
5239            62000         END     DISK
```

*Program Listing*

byte you want to write to. Start contains the address of the first byte to which you want to write.

The instruction at 32000 loads the BC register with the contents of Finish. That's the last byte you want to write.

Now you have a label. This label is More, and it's "load A with the contents of HL." The comment there is "Get the byte."

The next instruction is call ROMWT (ROM Write). That puts it in the buffer.

The next instruction (line 37000) is "jump-relative to UERR if nonzero." If something went wrong, you go to the error-handling routine.

Next you push HL to save the position to which you just wrote. Then at line 39000, subtract-with-carry the HL from BC.

**Paul:** To see how much you have left?

**Vernon:** Right—well if it's zero or not. Then you POP the HL to restore the pointer. Then increment the HL to position it at the next byte. Now remember, this increment HL does not affect the flags at all. If I'd increment H or increment L it would affect the flags. But incrementing a full 16-bit register does not.

At line 44000, jump-relative if nonzero to More. The zero flag is still in the state it was left by the subtract-with-carry in line 39000. So you'll go through this routine until the HL reaches the BC, or in other words, until I have written the last byte to write. And after that you jump to close.

**Paul:** Let's see if I understand this. The user doesn't even have to break the data up into 256-byte chunks. It automatically gets transferred to the output buffer 256 bytes at a time?

**Vernon:** I think it gets transferred to the I/O buffer one byte at a time.

**Paul:** It gets flushed to disk and then the pointer gets reinitialized to the beginning of the I/O buffer?

**Vernon:** Right.

**Paul:** The user doesn't have to worry about any of that.

**Vernon:** Not at all.

Next (line 46000) you have the label BADNAM (Bad Name). Load A with 13H. The comment there is "13H indicates an improper or illegal file name."

The next label is UERR. The instruction is OR 40H. That's so you won't have this DOS ERROR = xxx message.

**Paul:** That's the bit 6 thing.

**Vernon:** It's the same as setting bit 6, right. The next instruction is Jump to Error, which will assemble as Jump 4409H.

Now, if you want to incorporate the routine into your own program instead of ORing with 40H, you OR with C0H. Then, whatever you've written can make a call to this routine. When it jumps to Error, it will display the error message and return. But if you OR with only 40H (you don't have bit 7 set), it will jump to 402D and you'll be back at DOS Ready. That is internal in DOS's error-handling overlay.

**Paul:** ORing C0H sets bit 7?

**Vernon:** Yes, and bit 6. If you want to see DOS ERROR = xxx and still return to the calling routine, you can OR 80H. That's the end of the error-handling routine.

Now you go back to the label RDISK at line 50000. This time you're going to read from disk. Now before you got here from above, you've already called FILK, which has positioned the file name from the target of the HL to the DCB pointed at by the DE. You've already set the logical record length to 256 by loading B with zero. Then you checked the mode. The mode was nonzero, so you jumped down to this label RDISK. What you do there is to call Open.

If you're going to read a file, it has to already exist. So unlike INIT, Open won't create a new file for you. But it will find an old one with the given file name. The next instruction is "jump if nonzero to UERR." That means there's an error. Either the file doesn't exist, or something went wrong. The next instruction is Load HL with the contents of Where. That gives the place to store the data that is read in.

The next label, at line 53000, is GDISK. The instruction is Call ROMRD. That gets a byte. The first time the program encounters this instruction, it will do two things: It will read in a sector to the I/O buffer, and it will put into the accumulator the first byte of the data read into the buffer. On subsequent calls, it will put into the accumulator the next byte on queue in the buffer.

The next instruction is "jump-relative if nonzero to EOF." Nonzero can mean two things. It either means you read in everything, or you had an error. If you don't encounter a nonzero condition, you fall through to line 55000 where you load the memory location pointed to by HL with A. That puts the byte into your designated read-in area. Then you increment HL and jump-relative back to GDISK. It keeps going through this loop until an error occurs or the EOF is reached.

At line 59000, you have the label EOF: Compare 1CH. The comment is "IF 1CH Then Natural End of File." Then return if zero—go home if okay. Otherwise, jump-relative to UERR, the error-handling routine. That's it.

Now let's talk about the program a bit. There are four labels that are very important to this program. I've established them at particular points, but they don't have to be there. They are Start, Finish, Where, and Mode. They are consecutive labels.

The user, if he wants to write something, has to know where he wants to start writing it from, and where he wants to finish writing it from. He must put a value in Start, which is a 16-bit word. For example, Start can contain 0000 and Finish can contain 2FFFH. Then we would write the ROM out to a disk file, provided Mode contains a zero.

*" Vernon Hester is the king of TRS-80 disk I/O. "*

If, on the other hand, you want to read a disk file into RAM, you have to establish the Where address of where you want to start reading it into. This routine doesn't check for TOPMEM (HIGH$) or anything like that. You can do that another way. But, if you want to start reading something into 6000H, you'd put 6000H into the contents of Where, and you'd put a nonzero into the contents of Mode.

When you call this routine, as I've said, HL should point to the file name. What you can do is to put another label in here called FILNM (file name). The first instruction is to load DE with the file buffer:

```
20000 LD DE,FBUF
```

You can then load HL with file name

```
20500 LD HL,FILNM
```

That's the only maintenance required.

**Paul:** When I want to read in a file, do I have to specify the correct logical record length (LRL)?

**Vernon:** It depends on the DOS. In this particular instance, B is loaded with zero before determining whether it's a read or write, so the LRL is fixed at 256.

**Paul:** And if the file had been created with a different LRL, like a word-processing file with an LRL of one, what would happen?

**Vernon:** It would open up the file with an LRL of 256 with any DOS except MULTIDOS. (Vernon laughs here.)

**Paul:** MULTIDOS would open the file with a length of one?

**Vernon:** Yes.

**Paul:** And with other DOSes, what would be the result of trying to read the file after opening it with a wrong LRL?

**Vernon:** All the other DOSes, I think, regardless of what the directory entry says the LRL of a file is, will open up the file with the LRL indicated by the B register.

**Paul:** What would the result of that be? Would you get an error and jump to the...

**Vernon:** No! You would just read it in as 256 bytes.

**Paul:** So essentially, once it's loaded into memory, it's all the same anyway—it doesn't matter.

**Vernon:** Right. If you followed this routine with NEWDOS80 and opened a file that had an LRL of one, it would read it in as LRL 256. The pattern on the disk isn't affected by the LRL.

**Paul:** So everything would come out okay with any DOS?

**Vernon:** Right. This subroutine is very short, not even one page.

**Paul:** What is it, about 200 bytes?

**Vernon:** Let's see. If you assemble it at 5200H, it will go to 5284, inclusive—132 bytes. ■

# For serious business microcomputing, only one operating system exactly fits.

**NOW AVAILABLE FOR TRS-80 MOD II**

Whether you're in business and do microcomputing, or in computing and sell to business, you'll like OASIS.* Not a hobby or scientific system warmed-over for business use, OASIS is the only operating system designed from the ground up for business.

**SERIOUS BUSINESS COMES IN ALL SIZES.**

Whatever your business need, OASIS has the operating system to match: 8-bit or 16-, single-user or the multi-user system that professionals tell us makes micros run like minis. And that's even truer now with new OASIS-16.** *(OASIS exact business fit #1: choice.)*

**A SYSTEM IS ONLY AS GOOD AS THE BUSINESS PROGRAMS IT RUNS.**

The acid test for any business operating system is the application software available to run on it. There's plenty for OASIS—for nearly any job.

And it's top quality, too, because our integrated tools are top quality—and there are more of them. Like a high-level BASIC Interpreter/Compiler/Editor/Debugger; ISAM/Keyed/Direct/Sequential Files; a smart Print Spooler; *Automatic* Record Locking (OASIS exclusives!); plus COBOL & 'C' Languages. These tools are mandatory for high-quality business application program development—ask any developer. *(OASIS exact business fit #2: high-quality application software.)*

**PORTABILITY PROTECTS BUSINESS SOFTWARE INVESTMENTS.**

OASIS is custom-fitted to manufacturers' hardware so application software developed to run on one OASIS equipped machine can also run on others—and is upwardly compatible from 8-bit OASIS Single-User to Multi-User, on up to OASIS-16. This kind of application software portability is exclusive with OASIS.

Device independence allows various printers and terminals to be used—with no modification to application software: OASIS *system* software compensates for differences. *(OASIS exact business fit #3: portability.)*

**ACCURATE DATA & A SECURE SYSTEM REDUCE BUSINESS RISKS.**

Data integrity—a challenge for any multi-user system—is insured by OASIS File & *Automatic* Record Locking. With it, normally all users can view a record at the same time. BUT, if the record is being updated by one user, other users are automatically denied access until the update is complete. That means data is always accurate and up-to-date. It's still another OASIS exclusive.

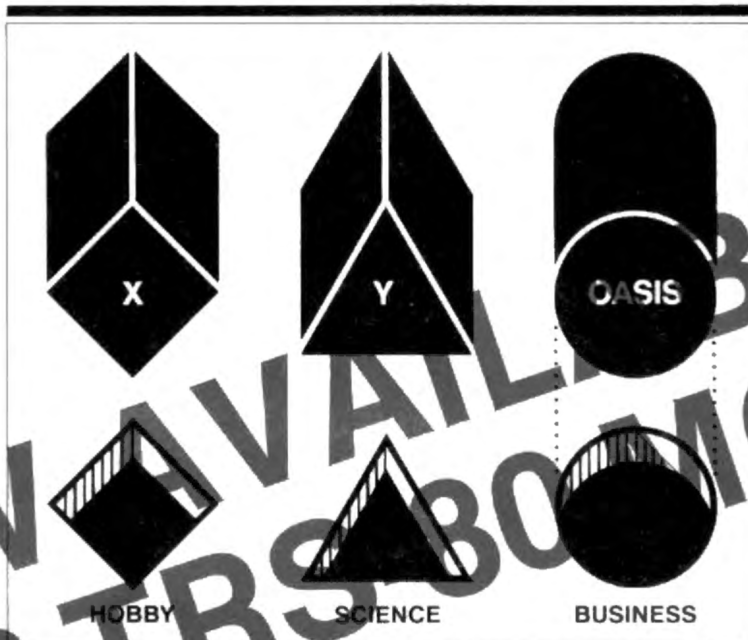
For system security, there's OASIS Logon, Password & Privilege Level. User Accounting keeps track of who used the system, when. *(OASIS exact business fit #4: data integrity/system security.)*

**A FRIENDLY SYSTEM IS GOOD BUSINESS.**

For user-friendliness, OASIS sets new standards. Example: the EXEC Job Control Language is so smart it walks users through their applications—and around the operating system.

With our 'HELP' feature, if you are unsure of the functions and options available, type 'HELP' and the command function title—OASIS displays the syntax and options available.

Of course it's all in your language—not computerese. *(OASIS exact business fit #5: user-friendliness.)*




**AND, AND, AND...**

Some of the best, most extensive documentation in the industry; a packed Application Software Directory; multi-level training; direct telephone support; worldwide sales & service; options like CONTROL Relational Data Base Management System and MASTERPLAN Financial Modeling Package; OASIS has it all. *(OASIS exact business fits #6 through #12.)*

Now you know why there's no reason to struggle trying to put a square peg in a round hole. For serious business microcomputing, there is one operating system that exactly fits: **OASIS**. Call or write us today for details.

* For Z80
** Available for 8086/8088. Available soon for 68,000, 16,000 and others

**OASIS**
STRICTLY BUSINESS

**PHASE ONE SYSTEMS, INC.**
7700 Edgewater Drive, Suite 830
Oakland, CA 94621-3051
Telephone 415/562-8085 TWX 910-366-7139

I'm serious about my business
please send me:

- ☐ OASIS-16 Manual $75
- ☐ OASIS Manual, $60
- ☐ Free Application Software Directory and put me on your mailing list

(Add $3 for shipping. California residents add sales tax.)

Name ______
St. (No Box #) ______
City ______ State ______ Zip ______
☐ Check enclosed ☐ VISA ☐ Mastercharge
☐ UPS C.O.D. ☐ American Express
Card No. ______ Exp. date ______
Signature ______

See List of Advertisers on Page 451

# COPERNICA MATHEMATICA by Bruce Douglass

## More about regression

In an analysis of variance (ANOVA), you have two or more groups of data. The hypothesis you're testing boils down to the question, "Are any groups different from the others?" The hypothesis that all groups are the same is called the "null hypothesis." The hypothesis that at least one pair of groups differ from each other is called the "alternative hypothesis." But what do we mean by different?

In the ANOVA test, you compare the means of the groups. However, just because the means of the groups are not the same does not indicate that the groups are actually different. If you sampled height in New York and in Oregon, and the mean heights were 5' 9" and 5' 10", are the heights in the two states actually different or was the sample of people chosen so small that the difference can be explained by the chance occurrence of just talking to tall people in Oregon (the only ones with their heads above water), and just talking to short people in New York (the ones below the smog)?

So you see the problem of inferential statistics. You must infer a conclusion about the means of the total population of New York and Oregon on the basis of a small sample of people in each state. A small sample can be biased for a number of reasons, including random chance. In fact, it would be very surprising if you got exactly equal numbers for the mean height even if the means were actually identical!

The job of inferential statistics is to tell us whether the groups are different, and also to give us a measure of the probability that our assessment is wrong.

ANOVA does this by making two independent measures of variability. The first is the average variability within each group, which we will call "within" variability. The second is the variability among the means of each group—the "between" variability. If the groups are the same (which implies that they originate from the same population) then these two measures of variability ought to be the same. If they are not the same, then "between" ought to be bigger than "within."

That is the essence of ANOVA and the F-ratio test used to determine how different the means are. The F-ratio test uses a measure of variability called "sum of squares" for both "within" and "between." This sum of squares is then normalized by dividing by the degrees of freedom for each, leaving the "mean square." The F ratio is the mean squares for "between" divided by the mean squares for "within." The task of determining whether the groups are different then reduces to the problem of testing to see if the F ratio is different than 1.

*"Degrees of freedom is the number of items free to vary minus the restrictions placed on the data."*

Degrees of freedom is the number of items free to vary minus the restrictions placed on the data. The more subjects you have, the greater the degrees of freedom. You might find it easier to think intuitively—the more subjects you have, the less the probability that your sample is biased by random chance, so the more reliable your conclusions drawn from your analysis will be. There are tables of F versus degrees of freedom "within" versus degrees of freedom "between." In our program, we will use a rational polynomial approximation to the distribution of the F ratio (see my February 1982 column for a description of approximation theory). The table and the approximation give you the probability that the F value you observed from the sample means was due to random chance variation and not to any real difference among the groups.

### F Ratio

What about the job of computing this F ratio? Geoffrey Keppel has written an excellent notation for ANOVA computational formulae in his book *Design and Analysis* published by Prentice-Hall. Let's consider the notation and kind of data applicable for a while.

By "score," I mean that you make a measurement on a number of subjects. For example, you might want to relate score on an IQ test to drinking coffee. In this case, you might have three groups of subjects: Group A would drink no coffee, group B would drink one cup one hour before taking the test, and group C would drink five cups within 30 minutes before the test. The results of the experiment—the F test denoting difference among the groups—would indicate whether coffee affects scores on IQ tests.

Note, by the way, the implicit assumptions of the test. You are assuming the groups' IQs are the same at the start of the experiment—you could ensure this by randomly assigning people to the groups. If the number of subjects is great enough, the probability of random bias to one or more groups decreases. That is, with enough subjects, the number of high, medium, and low-IQ people should be about the same.

You are also assuming that the only difference among the groups is the amount of coffee drunk. Any other difference (like if some of the people did not have time to eat breakfast) would muddy your conclusions.

Let's let "a" be the number of groups and "s" be the number of subjects. We will assume that all the groups have the same number of subjects. The total number of subjects in the experiment is a*s. We will use AS to denote a single score, and A to denote a group sum. To make the formulae less intimidating, we will make the following definitions:

```
10 DEFINT I-N
20 CLS:PRINT" ****** ANALYSIS OF VARIANCE PROGRAM ******"
30 PRINT"                    WRITTEN BY"
40 PRINT"               BRUCE POWEL DOUGLASS"
50 PRINT"                   1005 WEST MAIN"
60 PRINT"              VERMILLION, SD   57069"
70 PRINT
80 INPUT"ENTER THE NUMBER OF GROUPS";NG
90 INPUT"ENTER THE NUMBER OF SCORES PER GROUP";NS
100 DIM X(NG,NS)
110 REM  ****** DATA INPUT
120 FOR I=1 TO NG
130   PRINT"GROUP #";I
140   FOR J=1 TO NS:PRINT"ENTER SCORE #";J;" : ";
150   INPUT X(I,J)
160 NEXT J,I
170 REM   ******  RUN ANALYSIS
180 DIM SM(NG),SQ(NG)
190 FOR I=1 TO NG
200   FOR J=1 TO NS
210     SM(I)=SM(I)+X(I,J)
220     SQ(I)=SQ(I)+X(I,J)[2
230     T=T+X(I,J)
240   NEXT J
250   AS=AS+SQ(I):A=A+SM(I)[2
255 NEXT I:A=A/NS
260 TL=T[2/NG/NS
270 SB=A-TL:SW=AS-A:ST=AS-TL
280 DB=NG-1:DW=NG*(NS-1):DT=NG*NS-1
290 MB=SB/DB:MW=SW/DW
300 F=MB/MW
1000 ZF=F:D1=DB:D2=DW:GOSUB 1040:P1=P
1010 REM ** F RATIO PRINT FORMAT  **
1020 CLS:PRINT"  SOURCE","    SS"," DF","F RATIO"
1030 FORI=1TO127:SET(I,4):NEXTI:PRINT:PRINT
1031 PRINT"BETWEEN",SB,DB,F
1032 PRINT"WITHIN",SW,DW
1033 PRINT"TOTAL",ST,DT
1034 PRINT
1035 PRINT"P(F)=";P
1036 END
1040 'P(F) COMPUTATION
1050 IF ZF<1 THEN P=0:RETURN
1060 A=2/9/D1:B=2/9/D2
1070 P=ABS((1-B)*ZF[(1/3)-1+A)/SQR(B*ZF[(2/3)+A)
1080 IF D2<4 THEN P=P*(1+.08*P[4/D2[3)
1090 P=.5/(1+P*(.196854+P*(.115194+P*(.000344+P*.019527))))[4
1100 P=1-INT(P*4000+.5)/4000
1110 RETURN
```

*Program Listing 1. ANOVA*

$\{T\} = T^2/(a^*s)$
$\{A\} = \Sigma A^2/s$ : A is the sum of each group here
$\{AS\} = \Sigma(AS)^2$

The sum of squares for the total $SST = \{AS\} - \{T\}$; the sum of squares between the groups, $SSB = \{A\} - \{T\}$; and the sum of squares within is defined to be $SSW = \{AS\} - \{A\}$. By the way, you can check your arithmetic by the relation $SST = SSB + SSW$. Since the degrees of freedom is the number of observations minus the number of restrictions imposed by computation, the degrees of freedom for the "between" sum of squares is $(a - 1)$; we estimate the mean of the overall experiment (thereby imposing one restriction) and we have "a" groups. The degrees of freedom for the "within" sum of squares is computed easily as well. We have "s" subjects in each group, and we use these to calculate the mean. Therefore, each group contributes $(s - 1)$ degrees of freedom. Since there are "a" groups, we have $a^*(s - 1)$ for the "within" degrees of freedom. The degrees of freedom for the total sum of squares is $a^*s$ (the number of samples) minus the restrictions placed on the data by our computing the mean, or $a^*s - 1$.

We divide the sum of squares by their respective degrees of freedom to get the mean squares: $MSB = SSB/DFB$, and $MSW = SSW/DFW$. The F ratio is defined to be the ratio $F = MSB/MSW$, and it has $(a - 1)$ and $a^*(s - 1)$ degrees of freedom. We are now ready to write the program.

The ANOVA program (see Listing 1) shows the simplicity of the method without the extra clutter of saving the data to disk, or editing the data files. When you use it, you will be asked for the number of groups and the number of scores in each (your groups must have equal numbers of scores). Then you type in the data, one group at a time. The program thinks a little bit (as per our discussion above) and grinds out the data.

**Regression**

Analysis of variance is a special case of regression. In regression, you have paired or matched observations in two or more groups. The object is to find a line that best fits the relationship among the data points.

For example, there is a relationship between height and weight. The taller people are, the more they weigh. But that is not all there is to the subject—what about body fat and occupation? The amount of variability in one variable that is explained by another is called the correlation between them. This correlation is the relationship between the variables. If there is a one-to-one correspondence between the variables, then the correlation equals one; if there is no relationship (that is, the variables are totally independent), then the correlation equals zero; finally, if the relationship is corresponding and inverse, then the correlation is minus one. (This type is called a product-moment correlation.)

*"The amount of variability in one variable that is explained by another is called the correlation between them."*

Regression uses the correlation to find the best-fitting line showing the relationship between the variables—"best" in terms of least-squares best fit. If $Y_i$ is the actual value of data point i, and $Y_i'$ is the value predicted by the regression line, then the least-squares best fit finds the minimum value for the function $(Y_i - Y_i')^2$. The square of the difference between the real values and the predicted values is then a minimum. This assumes that the relationship between the variables is a straight line. The less true that assumption is, the worse the fit of the line will be to the real data.

The basic equation for a simple bivariate regression is $Y' = aX + b$. The slope of the line is "a," and "b" is the intercept, or the place where the line crosses the Y axis on a graph. Let's let "r" be the correlation between X and Y. Then the equation for r is:

$r = \Sigma xy/SQR(\ \Sigma x^2 \Sigma y^2)$

where

$\Sigma x^2 = \Sigma X^2 - (\Sigma X)^2/N$ (deviation sum of squares)
$\Sigma xy = \Sigma XY - (\Sigma X)(\Sigma Y)/N$ (sum of cross products)

N = number of points
X = raw data point for variable X
Y = raw data point for variable Y
XY = product of $X_i$ and $Y_i$

The value $r^2$ is called the common variance. It is the percentage of the total variability in Y that is explained by the presence of variable X. Note that $1 - r^2$ is the variability that X and Y do not have in common; it is called the residual variance, and stands for the variance in Y that is not explained by the presence of variable X.

In our regression equation, the line parameters "a" and "b" can be defined by:

$b = \Sigma xy/\Sigma x^2$

$a = Y_m - bX_m$

where

$Y_m$ = the mean of Y
$X_m$ = the mean of X

An inference is usually associated with some test to give a measure of how likely it is that we are wrong in drawing that inference. A common one used for regression is the F test, defined above. The F ratio defined for the bivariate regression is:

$F = r^2{*}N/(1 - r^2)$

So far, we have only discussed the case in which there is a variable to be predicted (the dependent variable Y) and a single predictor variable (also known as the independent variable X). The general case is called multivariate regression. In this case, there are many predictor variables and a single predict-

*Program Listing 2. Regress*

```
10 DEFINT I-N
15 CLS
30 PRINT"** WRITTEN BY BRUCE POWEL DOUGLASS 1981 **"
180 REM ******  INPUT ROUTINE
190 INPUT"HOW MANY INDEPENDENT VARIABLES";M:INPUT"NUMBER OF SCOR
ES PER GROUP";N
200 DIM X(M,N)
210 FOR K=1 TO N
220 PRINT,"Y("K")=X 0 ("K")"
230 FOR J=0 TO M:PRINT "INPUT X"J"("K")=";:INPUT X(J,K):NEXT J,K

240 DIM P(M),SX(M),S(M,M),R(M,M),T(2*M,M),E(M),B(M),SB(M),G(M),H
(M),Z(M+1,M+1),D(M),C(M)'RUNTIME!
250 FOR K=1 TO N:FOR J=0 TO M
260 P(J)=P(J)+X(J,K):NEXTJ
270 FOR L=0 TO M:FOR I=0 TO M
280 S(I,L)=S(I,L)+X(I,K)*X(L,K):'MATRIX FOR SUM CROSS PRODUCTS
290 NEXT I,L,K
300 FOR I=0 TO M:FOR J=0 TO M
310 S(I,J)=S(I,J)-P(I)*P(J)/N
320 NEXTJ,I
330 REM ******  BASIC STATISTICS GENERATION
340 FOR J=0 TO M:P(J)=P(J)/N:SX(J)=SQR(S(J,J)/(N-1))
350 'P=MEAN,SX=ST.DEV.,NOTE THAT Y=X(0)
360 NEXT J
370 REM ******  CALCULATE R (CORRELATION) MATRIX
380 FOR J=0 TO M:FOR I=0 TO M
390 R(I,J)=(S(I,J))/SQR(S(I,I)*S(J,J))
400 NEXT I,J:IF M=1 THEN GOTO 460
410 FOR J=1 TO M:FOR I=0 TO M
420 Z(I,J)=R(I,J):NEXT I,J:NQ=M
430 GOSUB1550:' INVERT THE CORRELATION MATRIX & PUT IT INTO Z()
440 FOR J=1 TO M:E(J)=Z(M1,J)
450 B(J)=(Z(M1,J)*SX(M1))/SX(J):NEXTJ
460 AR=P(0):IF M=1 THEN B(1)=S(1,0)/S(1,1):AR=AR-B(1)*P(1):GOTO
480
470 FORI=1TOM:AR=AR-B(I)*P(I):NEXTI'AR=INTERCEPT
480 REM ******  R SQUARED =SS(REG)/SS(T)
490 FOR I=1 TO M:RG=RG+B(I)*(S(0,I)):'RG=SS(REG),S(M1,M1)=SS(T)
500 NEXT I:RS=RG/S(0,0):IF M=1 THEN RS=R(1,0)[2:' MULTIPLE VARIA
NCE
510 RM=SQR(RS):' MR=MULTI CORRELATION
520 IF RS>=1 THEN SY=0 ELSE SY=SX(0)*SQR(1-RS):C1=1.96*SY:C2=2.6
1*SY
530 IF RS<>0 THEN F=RS*(N-M-1)/((1-RS)*M) ELSE F=0:' DF(REG)=M,D
F(RES)=N-M-1
540 ZF=F:D1=M:D2=N-M-1:GOSUB 1780:P1=P
550 CLS:'****** PRINT BASIC STATS
560 FOR J=0 TO M:FOR I=0 TO M:PRINT USING"#.####";R(I,J);:PRINT"
  ";:NEXTI:PRINT:NEXTJ:PRINT
570 PRINT@832,"ABOVE IS THE CORRELATION MATRIX WITH X1 THROUGH X
"M";
NOTE THAT Y IS AT THE 0 POSITION IN ROWS AND COLUMNS"
580 INPUT"TO CONTINUE PRESS 'ENTER'";Z
590 CLS:PRINT"SOURCE","MEAN","ST. DEV.","RY.X"
600 FORI=1TO127:SET(I,4):NEXTI:PRINT:PRINT
610 PRINT"Y",P(0),SX(0),RM
620 FOR I=1 TO M:PRINT"X("I")",P(I),SX(I),R(0,I)
630 NEXT I:PRINT:PRINT:PRINT"NOTE THAT THE RY.X FOR Y IS THE MUL
TIPLE CORRELATION":PRINT"THE VARIANCE ACCOUNTED FOR BY THE REGRE
SSION IS";RS
640 PRINT"THE VARIANCE NOT ACCOUNTED FOR IS";1-RS
650 PRINT:INPUT"TO CONTINUE, PRESS 'ENTER'";Z
660 GOSUB 1500
670 PRINT"REGRESSION",RS,M,F
680 PRINT"RESIDUAL",1-RS,N-M-1,"P(F)=";P1:PRINT
690 PRINT"THE REGRESSION EQUATION IS":PRINT"Y'=";:FOR I=1 TO M:P
RINT B(I)"X("I") + ";:NEXT I:PRINT AR
710 END
1380 REM ****** ROUTINE FOR ROW SWITCH
1390 FOR K=0 TO M
1410 W=E(K,J):E(K,J)=E(K,J+1):E(K,J+1)=W
1420 NEXT K:RETURN
1470 CLS:PRINT"SINGULAR MATRIX ENCOUNTERED.
NO UNIQUE SOLUTION EXISTS"
1490 END
1500 REM ** F RATIO PRINT FORMAT  **
1510 CLS:PRINT"SOURCE","  SS"," DF","F RATIO"
1520 FORI=1TO127:SET(I,4):NEXTI:PRINT:PRINT:RETURN
1530 END
1540 REM THIS IS THE ROW REDUCTION SUBROUTINE
```

Listing 2 continues

*Listing 2 continued*

```
1550 REM ROW REDUCTION ROUTINE
1560 FOR I=0 TO NQ:Z(I,NQ+1)=Z(I,0):Z(NQ+1,I)=Z(0,I):NEXT
1570 FOR J=1 TO NQ:IF Z(J,J)=0 THEN GOSUB 1380
1580 A=Z(J,J):FOR I=1 TO NQ+1
1590 Z(I,J)=Z(I,J)/A:NEXT I
1600 IF J=NQ THEN 1670
1610 FOR T=J+1 TO NQ
1620 IF Z(J,T)=0 THEN 1660
1630 B=Z(J,T):FOR I=1 TO NQ+1
1640 Z(I,T)=Z(I,T)-Z(I,J)*B
1650 NEXT I
1660 NEXT T,J
1670 FOR J=NQ TO 2 STEP-1:FOR T=1 TO J-1:B=Z(J,T)
1680 FOR I=NQ+1 TO J STEP-1
1690 Z(I,T)=Z(I,T)-Z(I,J)*B
1700 NEXT I,T,J
1710 FOR I=0 TO NQ:Z(I,0)=Z(I,NQ+1):Z(0,I)=Z(NQ+1,I):NEXT
1720 RETURN
1780 'P(F) COMPUTATION
1790 IF ZF<1 THEN P=0:RETURN
1800 A=2/9/D1:B=2/9/D2
1810 P=ABS((1-B)*ZF[(1/3)-1+A)/SQR(B*ZF[(2/3)+A)
1820 IF D2<4 THEN P=P*(1+.08*P[4/D2[3)
1830 P=.5/(1+P*(.196854+P*(.115194+P*(.000344+P*.019527))))[4
1840 P=1-INT(P*4000+.5)/4000
1850 RETURN
```

ed variable. For example, weight may be predicted by height, fat percentage, and activity level. Weight would be the predicted variable, and the others would be used in predicting a weight for a given height, fat percentage, and activity level. The general regression equation may be written:

$$Y' = a_1X_1 + a_2X_2 + a_3X_3 + \ldots + b$$

$$\text{or } Y' = \Sigma A_iX_i$$

with an F ratio

$$F = (P^2/k)/((1 - R^2)/(N - k - 1))$$

where

R = multiple correlation coefficient
N = the number of scores
K = the number of independent variables

The large R, the multiple correlation coefficient, is different from the small r. Each small r, or first-order correlation, stands for the covariability between $X_i$ and Y. But what if the correlation between $X_i$ and $X_j$($i \neq j$, of course) is nonzero? Then some of the same relationship explained by $X_i$ is shared by $X_j$; they explain the same portion of the variability in Y. So you cannot just add up the variances. If the correlations among the independent variables are all zero, then $R^2 = r_1{}^2 + r_2{}^2 + r_3{}^3 + \ldots$ If the correlations are nonzero, this will overestimate the multiple correlation. After all, in a multivariate regression we are interested in the total predictability of Y by all the Xs together, right?

While I don't want to derive any of this, a straightforward computational method for obtaining the multiple correlation coefficient and the b-weights for the regression equation is presented by Guttman ("A New Approach to Factor Analysis," *Mathematical Thinking in the Social Sciences,* 1954, Free Press).

First, calculate all the first-order correlations among the Xs and Y (e.g., $X_1$ and $X_4$, $X_3$ and Y, and so on). Put these in a matrix called RG. Invert the matrix. If $r_{yy}$ indicates the cell containing Y versusY in the inverted matrix RG, then $R^2 = 1 - 1/r^{yy}$. And the b weights are defined to be:

$$Bi = -r^{yi}/r^{yy} * S_y/S_i$$

where

$S_y$ = standard deviation of Y
$S_i$ = standard deviation of $X_i$

Although this probably seems a little mysterious, it works and is easy to program. You must know how to invert a matrix, but that is the only programming problem of any difficulty.

The program Regress (see Listing 2) does all this. You input the number of scores and variables, and the program computes the regression equation and the multiple-correlation coefficient. Like ANOVA, Regress is a bare-bones program; it lacks the niceties of data I/O, data editing, and a number of other statistical measures often given with the regression statistics.

I'm sure many of you will find the programs Regress and ANOVA sufficient for your needs, particularly if you add some simple data editing and file I/O routines. ■

# Load 80 goes color

Good things come to those who wait, especially at Christmas. The first Color Load 80 cassette is out this month, covering the major Color Computer programs from 1982. It sells for $14.97 and should warm any Color Computer owner's heart on Christmas morning.

Color Load 80 cassettes will be available every three months starting in March 1983. These 15-minute cassettes will pack a lot of material at 1500 Baud, and sell for only $9.97. A subscription plan will be announced in early 1983.

### Using Color Load 80

The documentation for each Color Load 80 includes a list of the programs and a short description of how to use them. My phone number and address at *80 Micro* are supplied in case of technical problems. The magazine, however, is the source of documentation for the individual programs. If you don't read the article, you won't be able to use Color Load 80.

Most programs on Color Load 80 are intended for a minimum system of 16K and Extended Color Basic, but a few may require only 4K Color Basic, and others 32K with disk drives. If the Key Box in the article doesn't list your system, then the program will not run correctly.

Model I and III Load 80s include Assembly-language files instead of machine-language. As of yet, no Editor/Assembler has become a de facto standard for the Color Computer, so Color Load 80 tapes will include machine-language files ready to load and execute using CLOADM.

### New Disk Operating System for MOD I/III

Starting with December 1982, Model I and III Load-80 disks will be supplied on TinyDOS (TDOS), the DOSPLUS kernel. This operating system has been stripped to make enough room for the programs.

Model I users can now use the Load 80 disk in Drive 0. To use the Basic files, simply enter Disk Basic and run the program of your choice. Assembly-language files will still require and Editor/Assembler. To make a backup, use the backup utility supplied on TDOS. If you have installed the LNDOUBLER 5/8 board in your expansion interface you will have to use another operating system to backup, the disk.

Load 80 disks are double-sided to allow more program room. Each side contains TDOS and can be accessed by simply flipping the disk.

Model III owners must use the TRSDOS Convert utility before using the Load 80 disk.

1 Make a backup of your TRSDOS disk and place it in Drive 0.
2 Type Purge, and then your TRSDOS disk's master password.
3 Answer Y for each file. It will be purged from the disk.
4 Insert the Load 80 disk in Drive 1. Type Convert.
5 Answer 1 for Source, 0 for Destination. The Load 80 programs will be copied onto the TRSDOS disk.
6 Flip the Load 80 disk over and do steps 4–5.

The Load 80 programs are ready for use on TRSDOS.

### Anniversary Load 80

Model I and III tapes and disks are available for the Special Anniversary *80*

| Month | Page | Article | Program | Type |
|---|---|---|---|---|
| Feb. | 202 | Colorful Titrations | TITRATE | Education |
| Mar. | 106 | Subchaser | SUBCHASE | Game |
| | 142 | Is a Rose in Color Still a Rose? | ARTIST | Game |
| | 142 | Is a Rose in Color Still a Rose? | MAZE | Game |
| | 142 | Is a Rose in Color Still a Rose? | 3-D | Game |
| | 212 | COLORMON | COLORMON | Utility |
| Aug. | 94 | The Colorful Computer (Part I) | THECOCO1* | Graphics |
| | 188 | Color Maze | MAZE | Game |
| | 202 | Four in One Plus Another | BRICKOUT | Game |
| | 202 | Four in One Plus Another | POINDEXTER | Game |
| | 202 | Four in One Plus Another | MATHTALK | Game |
| | 202 | Four in One Plus Another | PIXPRINT | Utility |
| | 202 | Four in One Plus Another | TANK-GUN | Game |
| | 260 | Space Duel | SPCDUEL | Game |
| | 264 | Square Game | SQRGAME | Game |
| | 268 | Color Breakaway | BRKAWAY | Game |
| Sep. | 140 | Basic Word Processing | WORDPROC | Application |
| | 230 | Joystick Paintbrush | PAINTBRU | Graphics |
| | 238 | Conversion | CONVRSNS* | Game |
| Oct. | 168 | Income Tax Estimator | TAXCALC | Application |
| Nov. | 164 | More Color Conversions | SUBDSTRY | Game |
| | 164 | More Color Conversions | LLANDER | Game |
| | 164 | More Color Conversions | MISSILES | Game |
| | 213 | Color Assembler | ASSEMBLR | Utility |
| | 272 | Show the Score | SHOWSCOR | Utility |
| | 304 | Test Patterns | TVTEST | Application |
| | 308 | Music Marvel | MUSIC | Application |
| Dec. | 388 | Picture-Editor | PIC-ED | Graphics |

* *All listings in the article are combined into one program. Each is accessed from a menu.*

*1982 Color Computer Load 80 Directory*

## RELOAD 80

*Micro* for $9.97 and $19.97. These tapes and disks are not part of a regular Load 80 subscription.

An index to Load 80 programs and Reload 80 columns for 1982 will be published int he January 1983 issue. Complete lists for 1981 and 1982 will be published in the Anniversary Issue. ■

| Program | Title | Page | Comments |
|---|---|---|---|
| 1 | COPYRGHT/BAS | — | None |
| 2 | VOICE/BAS | 72 | None |
| 3 | MAKEGFTX/BAS | 118 | None |
| 4 | READGFTX/BAS | 118 | None |
| 5 | SPOOLER/SRC | 130 | Needs EDTASM |
| 6 | DSPOOLER/SRC | 130 | Needs EDTASM |
| 7 | UNIPOKE/BAS | 146 | None |
| 8 | LVL2SRCH/SRC | 194 | Needs EDTASM |
| 9 | DISKMAP/SRC | 236 | Needs EDTASM |
| 10 | SEARCH/SRC | 262 | Needs EDTASM |
| 11 | SEARCH1/BAS | 262 | None |
| 12 | AUTOKEY/SRC | 280 | Needs EDTASM |
| 13 | AUTOKEY/BAS | 280 | None |
| 14 | POKEGPHS/BAS | 296 | None |
| 15 | MAZE19/BAS | 341 | None |
| 16 | SCROLL/SRC | 370 | Needs EDTASM |

*December Load 80 Directory*

# Sales Opportunity

## A few select territories available

Join a fast growing company involved in the microcomputer industry as a direct salesman. Sales involve our wide selection of books, four publications and software. We are looking for a non-smoking individual with a good sales background and a basic understanding of microcomputers. An excellent career awaits the person who wants to be "on his own." It will be the responsibility of our direct salesman to become familiar with all makes of computers and their sales outlets, amateur radio and electronic stores, book stores, and other outlets for our products in your given territory. We will also add other lines as they become available. Arrangements are being made to coordinate efforts with some other firms in the area, including a major printer manufacturer. We offer an excellent growth potential with a highly motivated, young corporation. If you or someone you know is interested in the opportunities available at Wayne Green Inc., please submit your background in detail, including sales experience, recent work history, knowledge of the microcomputing industry, educational background and references to:

Sales Manager
Wayne Green, Inc.
Peterborough, NH 03458

All replies will be immediately acknowledged and held in the strictest of confidence.

**WAYNE GREEN, INC.**
**(E-12) Pine Street**
**Peterborough, N.H. 03458**

# MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of **80 Micro**.
Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

☐ Extend my subscription one additional year for only $24.97.
☐ Payment enclosed ☐ Bill me

Canadian and Mexican $27.97 1 year only US funds. Foreign $44.97 1 year only US funds, drawn on US bank. Please allow 6-8 weeks for delivery.

**80 MICRO**

AFFIX LABEL

*If you have no label handy, print OLD address here.*

Name ______________________
Address ______________________
City __________ State __________ Zip __________

*print NEW address here.*

Name ______________________
Address ______________________
City __________ State __________ Zip __________

32DR8M

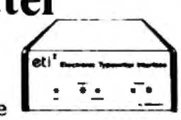
P.O. Box 981 • Farmingdale NY 11737

# Do You Use a Printer or Modem?

The average microcomputer "moves" data at 120,000 characters a second. A typical disk drive transfers it at 27,000 CPS. Most printers however plod along at 100 CPS and many modems squeak out 30 CPS. That's quite a drop in efficiency not to mention a waste of computing power!


We've broken this bottleneck with a smart little "white" box called the PRINTER OPTIMIZER® that features a 64,000 to 256,000 character memory buffer. Now you can "print" your inventory in 2 seconds instead of 10 minutes, and no longer will your computer be tied up transmitting or receiving modem transmissions. The OPTIMIZER is smart too. You'll notice a keypad and display in our picture. You can tell the OPTIMIZER to convert data, send control sequences, pause on cue, and more. For example: access all of your printer's capabilities (graphics, font size, forms control, special symbols) with the ease of selecting a station on a pushbutton car radio. Use it to adapt an XYZ printer to your ABC computer running a PDQ word processing program. Run a serial printer or modem from a parallel port. If you can afford and justify a disk drive, then you certainly need our OPTIMIZER to bring your printer, plotter or modem "up to speed" with the rest of your system.

# If You Need A Letter Quality Printer




...then consider our affordable yet high quality alternative. Our MEDIAMIX ETI²® converts an IBM Electronic Typewriter (and some other brands) into a computer printer. Every office needs a typewriter. So for example, you can buy an IBM Model 50 ET that costs about $1200 and have a superior typewriter, and then with our ETI² have a printer that produces probably the highest quality printed image available. You can even do typesetting like this ad! And for financial printing you can't beat a wide carriage IBM ET. It's a truly cost effective investment. There are a number of arguments in favor of choosing this route over a single use computer printer, and we have an article on this subject you should read.

We apply technology creatively, with insight and with respect for the non-technical end user. Feel welcome to write or call our toll free number for more information.

**APPLIED CREATIVE TECHNOLOGY, INC.**

2723 Avenue E East, Suite 717
Arlington, Texas 76011
(817)-261-6905
(800)-433-5373

COPYRIGHT 1982

See List of Advertisers on Page 451

# CALENDAR

## December

6-7 Future Computing Inc., Richardson, TX. **Home Computer Market Forum** Loews Antole, Dallas, TX.

6-8 IEEE Computer Society, Silver Spring, MD. **Winter Simulation Conference** Holiday Inn at the Embarcadero, San Diego, CA.

6-8 IEEE Computer Society, Silver Spring, MD. **VLSI & Microcomputers: Today and Tomorrow (TENCON '82)** Hong Kong.

6-10 IEEE Computer Society, Silver Spring, MD. **Tutorial Week West '82** San Diego, CA.

7-8 The Yankee Group, Boston, MA. **Strategy/Counterstrategy: The Network as a Corporate Resource** New York, NY.

7-9 IEEE Computer Society, Silver Spring, MD. **1982 Real Time Systems Symposium** Los Angeles, CA.

7-9 Technical Education Research Centers, Cambridge, MA. **Microcomputers in Education** St. Louis, MO.

10 IEEE Computer Society, Silver Spring, MD. **Computer Networking** Gaithersburg, MD.

9-12 National Computer Shows, Chestnut Hill, MA. **The Second Annual Southeast Computer Show and Office Equipment Exposition** Atlanta Civic Center.

7-8 The Yankee Group, Boston, MA. **Strategy/Counterstrategy: The Network as a Corporate Resource** Palo Alto, CA.

13-15 Technical Education Research Centers, Cambridge, MA. **Microcomputers in Education** Boulder, CO.

## January

17-18 EMA Management Associates Inc., Richmond, VA. **Using Small Computers in the Professional Design Firm** Hyatt Kuilima Northshore, Kahuku, HI.

17-20 IEEE Computer Society, Silver Spring, MD. **Optical Storage of Digital Data** Lake Tahoe, CA.

18-20 Technical Education Research Centers, Cambridge, MA. **Microcomputers in Education** Tallahassee, FL.

31-2 Future Computing Inc., Richardson, TX. **UCSD p-system Industry Forum** Loews Anatole, Dallas, TX.

## February

7-9 Technical Education Research Centers, Cambridge, MA. **Microcomputers in Education** Washington, DC.

14-17 IEEE Computer Society, Silver Spring, MD. **Computer Science Conference** Orlando, FL.

16-18 IEEE Computer Society, Silver Spring, MD. **International Solid State Circuits Conference** New York, NY.

17-19 Technical Education Research Centers, Cambridge, MA. **Microcomputers in Education** New York, NY.

21-23 American Federation of Information Processing Societies Inc., Philadelphia, PA. **Office Automation Conference** Philadelphia Civic Center.

23-25 Future Computing Inc., Richardson, TX. **Personal Computer Retail Forum** Loews Anatole, Dallas, TX.

25-27 **Computer Expo 83** Orlando, FL.

# Coming Next Month

To celebrate the new year, our January issue will include an incredible program—Creator, by Bruce Tonkin (see Proof Notes, p. 16 for more info).

Also featured will be articles telling you how to market your own software. Topics will include deciding on a publisher, copyrighting how-to, and documentation do's and don'ts.

Of course, we will continue Philip Van Praag's "Hardware Hacker" series and Margaret Grothman's "APL Primer," and feature several timely reviews and Color Computer articles. ■

COLOR COMPUTER

ONLY $18.95

TWO FOR $35.95

## ★ ★ JOYSTICKS ★ ★

Tired of broken joysticks? Ours are built to **last**, with good smooth pots, and a **strong** lever and internal mechanism. These are **superior** to the Radio Shack design. Get your joystick programs working the way they **should!**

Up to **20% OFF** on software.

We stock selections from **Prickley Pear, Tom Mix, Mark Data,** and others.

Write or call for free catalog.

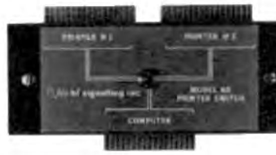
Add $2.00 shipping on Joysticks. We pay shipping on software. $1.50 additional for C.O.D.

**ENDICOTT SOFTWARE**
**P.O. Box 12543, Huntsville, AL 35802** ✓283

VISA **205/881-0506** MasterCard

---

## DO YOU HAVE 2 PRINTERS?



Now you can have **2 PRINTERS ON-LINE** at all times. Our MODEL 80 LINE PRINTER SWITCH is designed for the TRS 80 computer to allow the user to easily select between two printers by means of a conveniently located switch without changing cables. Our compact module connects to the parallel printer port of your Model I, Model II or Model III through a printer port extension cable and provides a card-edge connector for each of your two printers. Gold plated contacts ensure excellent connection reliability. Our Model 80 switch comes assembled, tested and ready to use for the low cost of $40 including shipping to any U.S. destination. A printer port extension cable is available from us at a cost of $12.50 for the Model I/III or $20 for the Model II. Call now for more information. Request application note AN-821 for Model I/III use or AN-822 for Model II use.

**hf signalling inc.** ✓409
P.O. Box 17510, Kansas City MO 64130
1-816-931-4448

VISA **TOLL FREE 1-800-831-4448** MasterCard

---

## Save Grief & Aggravation

*—Know Before You Buy—*

**READ**

**"Consumer Protection for the Microcomputer Owner"**

**COVERS:**

1. Preliminary Considerations before Buying
2. Legal Differences between Mail Order and Local Store
3. Pitfalls and Protection of Paying by Cash, Check, Credit Card, etc
4. What a Warranty Really Is
5. How to Complain Effectively
6. How to Legally Reject and Revoke Acceptance of Goods
7. Miscellaneous Hints and Suggestions

**ONLY $15 Post Paid (COD $3 Extra)**

Send Check or Money Order to

**L. J. KUTTEN, ATTORNEY**
**201 South Central**
**P.O. Box 16185**
**St. Louis (Clayton), MO 63105**
**314-721-6644**
**Mo. Residents add 4.625% sales tax**
Compuserve ID #70655,376 Source ID# TCK092

✓444

---

# This publication is available in microform.

**University Microfilms International**

Please send additional information

Name ____________________

Institution ____________________

Street ____________________

City ____________________

State __________ Zip __________

300 North Zeeb Road
Dept. P.R.
Ann Arbor, Mi. 48106
U.S.A.

30-32 Mortimer Street
Dept. P.R.
London W1N 7RA
England

---

**COLOR COMPUTER SOFTWARE**
**16-K EXTENDED BASIC**
**Adventure word games**

**· THE ALCHEMIST'S LABORATORY · $14.95**
mix the ingredients yourself

**· LOKAR'S MAGIC STAFF · $14.95**
five crystal statues, the magic staff and a castle filled with wild animals

**· NIGHT TOWER DESCENT · $14.95**
ten floors of danger, treasure and magic are between you and freedom

**· SQUEEZE · $14.95**
arcade style game where aliens close in from both sides at once

**· GRAPHIC SCREEN EDITOR · $16.95**
create pictures on the screen using joysticks or arrow keys-save on tape, erase, paint and many other features

**· COLOR SHOW DISPLAYS · $8.95**
five graphics programs that create endless changing patterns

all programs 16-K extended basic cassette only

**Send for free catalogue. We accept checks, money orders, Visa and Mastercharge (no C.O.D.s). Please add $1.00 for shipping. Send to:**

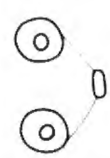
**REAL SOFTWARE CO.**
**P.O. BOX 401**
**HOPEDALE, MA 01747**
**(617) 393-6281** ✓345

Mass. residents add 5% sales tax
" dealer inquiries welcomed "

---

# 9 TRACK TAPE
## for TRS-80 Mod. II/16

ALDEVCO has available the ACCESS II ANSI std. 9 track mag. tape (1600 bpi) system tailored to the Radio Shack Model II/16 hardware and TRSDOS, providing hard disk backup AND access to mainframe data bases. Other versions are available for the S-100 bus and CPM.



**ALDEVCO**
5725 Dragon Way
Cincinnati, Ohio 45227
(513) 561-7671

---

# Subscription Problem?

80 Micro does not keep subscription records on the premises, therefore calling us only adds time and doesn't solve the problem.

Please send a description of the problem and your most recent address label to:

**80micro**
**Subscription Dept.**
**PO Box 981**
**Farmingdale, NY 11737**

Thank you and enjoy your subscription

---

## JONELL SYSTEMS
### TRS 80* MOD I

**SEABATTLE:** In assembler - fast, live-action, graphics oriented game. Play against another opponent or against computer. Hours of Fun. Req. Min 16K Level II $15.95

**BIORYTHM GENERATOR:** Generate anyone's biorythm chart for 6 mos. W/16K, 1 yr. W/32K. Video or L/P output with simultaneous graphic sine-wave presentation of all 3 functions. Flags critical days and much more. Req. min 16K Level II $19.95

**JOB ACCOUNTING:** Run your system like the big guys! Menu driven program records and files start and stop run times from system clock, elapsed time, dollar value of time, user code, date, and job name. Prints user account statements, credits payments and more. In disk basic for 32K system $14.95

**CMD 2 TAPE:** Make level II basic system tapes from CMD Disk Files. Shows load and entry addresses. Req. 32K disk system $14.95

**TAPE 2 CMD:** Make disk CMD files from level II system tapes. Shows load and entry addresses. Identifies non-system tapes. Much simpler than tapedisk utility. Req. 32K Disk System $14.95

*All programs supplied on tape. Add $5.00 to price for disk. Printed instructions furnished for each program. Give system configuration when ordering*

**JONELL SYSTEMS**
130 4th Street Altizer
Huntington, WV 25705
* Trademark of Tandy Corporation ✓347

---

## PUT YOUR COMPUTER TO WORK SAVING $$ EVERY TIME YOU BUY GROCERIES!!

## QSORT

is a fast, menu driven assembly language program that keeps track of up to 1200 store coupons. Enter your shopping list and QSORT locates every usable coupon within seconds.

A special filing system speeds physical retrieval of any coupon to a few seconds.

Comprehensive manual.

Satisfaction Guaranteed or your Money Back.

Specify Model I or III
cassette or disc ...... $39.90 ppd
(Calif. residents add 6%)

**BV ENGINEERING** ✓74
P.O. BOX 3351, RIVERSIDE, CA 92519
(714) 781-0252

### Project Management System

The Demi-Plan Project Management System is now available for Model I and III computers with 48K of memory, one or more disk drives, and a printer.

This program allows the user to define and maintain a file of project tasks and resources. This information is used to print various forms of Gantt charts, resource histograms, and task/resource cross-reference printouts. The system determines the critical path of a project and allows the project manager to introduce "what if" data to show the effect of date and resource changes on completion date.

The Demi-Plan disk sells for $49.95, with documentation alone available for $5. For more information, contact Demi-Software, 6 Lee Road, Medfield, MA 02052, (617) 359-4502.

Reader Service ✓580

### *Fortran Programs for Scientists and Engineers*

Over 60 of the most frequently used scientific algorithms, along with their program implementations written in Fortran, are presented in *Fortran Programs for Scientists and Engineers.* The material covered in this volume, including curve fitting, vector and matrix arithmetic, numerical integration, random number operations, and statistical analysis, will be of practical use to any scientist or engineer, and the programs can be readily adapted to most applications.


*Fortran Programs for Scientists and Engineers*

Each chapter contains exercises designed to test and extend reader comprehension of the material presented. For the first-time Fortran user (and the veteran who needs a quick reference), a summary of the syntax, standard functions, and reserved words of Fortran is included in the appendices.

Priced at $15.95, this book is available from Sybex, 2344 Sixth St., Berkeley, CA 94710.

Reader Service ✓585

### Sell Your Programs

Vendor-One is a Model I utility for program authors that permits your software to be sold on a custom-formatted disk that both single- and multiple-drive customers can use. The package includes a licensing agreement for unlimited distribution copies of your disk.

A Model III version will be available soon. Vendor-One sells for $30 from Applied 80 Softwares, 4316 Vermont Court, Virginia Beach, VA 23456.

Reader Service ✓574

### One-Megabyte Minifloppy Drive

The slimline minifloppy M4853 from Mitsubishi is a double-headed, 96-tpi, 5¼-inch standard drive. The unit has track-to-track access time of 3 ms, head load time of 50 ms, and a settling time of 15 ms.

The M4853 also includes a precision head load mechanism with all ferrite heads, a circular gimbal-type support for high data reliability and long media life, and an advanced direct-drive brushless motor that eliminates all concern for belt changes. Disk rotation starting time is 250 ms.

The drive's dimensions are 1.61 by 5.75 by 8 inches. Its price is $450. For additional information, contact Inflo Inc., 244 Mill Road, Yaphank, NY 11980, (516) 924-9229.

Reader Service ✓566

### Very Micro Computers

CompuGift Inc. (P.O. Box 1408, El Toro, CA 92630, (714) 768-8223) has introduced a line of desktop gift products in the shape of a 4½-by-4½-by-6-inch computer terminal. The Computer Caddy is a pen and pad holder,

*CompuGift*

while the Computer Planter and Computer Bank provide a place for flowers and coins.

The screen of each CompuGift forms a frame for your favorite picture. All items are individually gift boxed, and available in light grey, dark grey, yellow, and magenta high-quality plastic. Each is $8.99 plus $2 postage and handling.

CompuGift is the idea of a California systems analyst and his wife, a computer specialist. The company predicts first-year sales of $100,000, saying, "Everybody is computer crazy nowadays; who wouldn't like a gift like this?"

Reader Service ✓550

### Z80 System Reference Card

A new 16-page, $4.95 card summarizes and captures the entire language of the Z80 microprocessor chip for easy reference.

The card provides clock periods and microseconds for every instruction. A formula on the front page lets you calculate your own timing, regardless of your machine's MHz. In addition, all instructions are completely listed in op-code sequence for PEEKing, and mnemonic sequence for POKEing. A condition code chart and hex-to-decimal chart make programming easier. The undocumented op-codes found on some chips are included, as are decimal equivalents for every instruction.

The Z80 System Reference Card is available from Nanos Systems Corp., P.O. Box 24344, Speedway, IN 46224.

Reader Service ✓572


# Dosplus II magic.
# For you and your business.

The TRS-80 model II was created to help you take care of your business. DOSPLUS II was created to help your TRS-80 help you and your business even more.

DOSPLUS II replaces your O.E.M. dos with an ultra-sophisticated operating system that is incredibly faster, more reliable and is totally compatible with all standard business oriented software designed for use with the model II. The result is a computer/operating system that far exceeds manufacturer's specs — with more potential and more capacity to be of service to you.

DOSPLUS II operating systems are available for either your floppy or hard disk drives. The hard drive operating system is compatible with Radio Shack, QuCeS, Corvus Micro Mainframe, VR Data and other popular hard drive sub-systems.

DOSPLUS II...All the quality and dependability you'd expect from DOSPLUS.

**DOSPLUS II** DOSPLUS first in quality! First in the industry!

**POWERSOFT** ✓61
A Division of Breeze/QSD, Inc.
11500 Stemmons Fwy., Dallas, Texas 75229
To order call toll free 1-800-527-7432
For product information (214) 484-2976

**MICRO SYSTEMS SOFTWARE, INC.**
4301-18 Oak Circle
Boca Raton, FL 33431
To order call toll free 1-800-327-8724
For product information (305) 983-3390



✓See List of Advertisers on Page 451

## Farm Weather Center

Specialized software for deriving individual crop and field estimates of crop development and soil moisture balance is now available for the 48K Model III with one disk drive.

With simple rain gauge observations and daily readings of maximum/minimum temperature as input, the programs yield graphic displays of results along with summary reports and statistics through menu selections.

Prices start at $125 for one crop (either corn, soybeans, or spring wheat), while all three are available for $200. For more information, write Climate Assessment Technology Inc., 11550 Fuqua St., Suite 355, Houston, TX 77034.

Reader Service ✓584

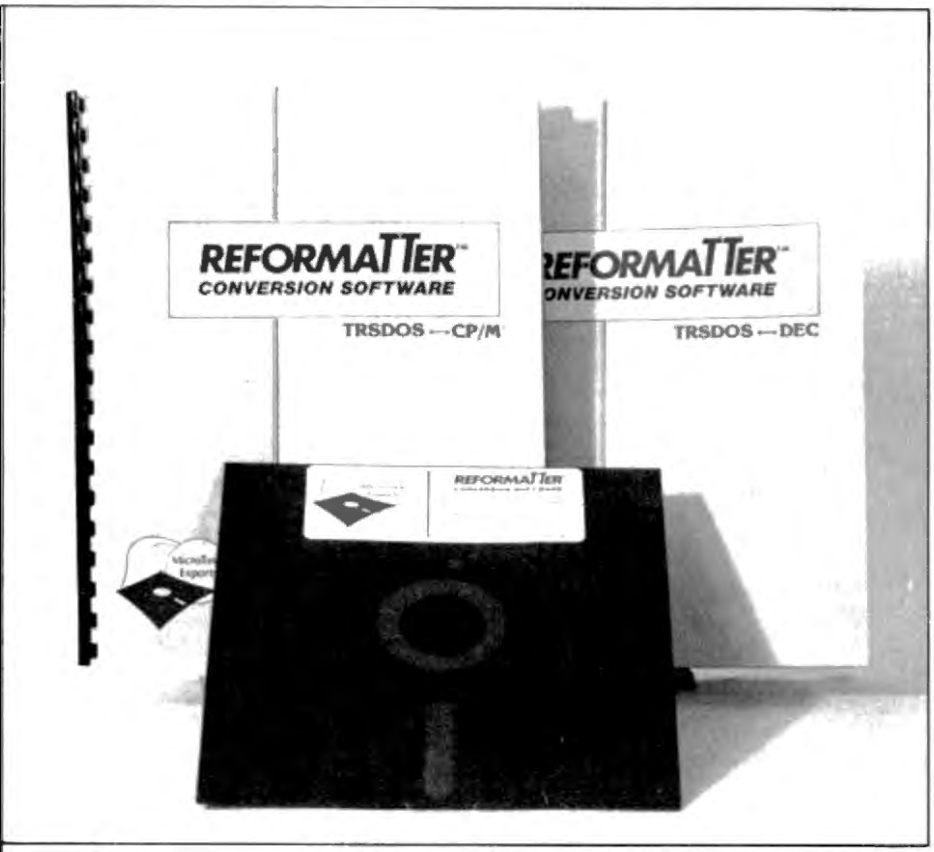
## Port Expanders

A series of software-controllable port expanders from Advanced Systems Concepts Inc. allow single-port units to interface with multiple peripherals or computers.

The port expanders are compatible with computers, printers, plotters, terminals, and modems that have an RS-232C or Centronics-compatible interface. The Model QS11 allows a single port to talk to one of four output devices. The Model MO11 allows four computers, modems, or terminals to share a common unit, such as a letter-quality printer.

Each unit weighs two pounds and comes in a 13.5-by-7-by-2-inch enclosure. They use CMOS technology and consume negligible power. Each retails for $450 from Advanced Systems Concepts Inc., P.O. Box Q, Altadena, CA 91001, (213) 684-5461.

Reader Service ✓562




*Reformatter*

## Bytewriter

The Bytewriter is a letter-quality, daisy-wheel printer/typewriter. It features a built-in Centronics-compatible parallel interface. All you do is connect it to your computer.

The Bytewriter has 44 alphanumeric keys plus 12 function keys, a 12-character buffer memory, a switchable second keyboard with foreign grammar symbols, memory erase of the past 10 characters, a half-space key, and automatic repeat on any key. Character densities of 10, 12, or 15 cpi are switch selectable.

This product sells for $795. For more information, contact Bytewriter, 125 Northview Road, Ithaca, NY 14850, (607) 272-1132.

Reader Service ✓590


*ASCI Port Expander*

## DEC and CP/M Conversion Software

Model II users can now exchange data files and source programs with CP/M or DEC RT-11 systems using Reformatter conversion software from MicroTech Exports Inc.

TRSDOS to CP/M reads and writes single-density CP/M disks, and TRSDOS to DEC reads and writes single-density RT-11 disks. A third program, TRSDOS to IBM, is available exclusively from Tandy Corporation through Radio Shack stores.

Each version runs on a single-drive Model II and provides full facilities for file reorganization and directory maintenance on the target disk. In addition to bidirectional transfer of files between the two operating systems, the program provides complete file statistics for the target disk, and permits the user to delete or rename files, alter user numbers, and convert TRSDOS disks to the target disk.

Priced at $249 for each version, Reformatter software is available from MicroTech Exports Inc., 467 Hamilton Avenue, Palo Alto, CA 94301, (415) 324-9114.

Reader Service ✓561

## Macro Assembler and Editor For Color Computer

Macro-80c is a macro assembler, screen-oriented editor, and machine-language monitor for the Color Computer disk system.

The assembler features local labels, conditional assembly, page headers, and symbol-table cross-reference listings. The full standard 6809 instruction set is supported, along with many additional pseudo-ops such as SET, ASK (for keyboard input during assembly), and even PEEK and POKE. There is a stack available for writing structured macros, and complete documentation on the assembler's support of such techniques as embedding machine language into Basic programs or writing programs that checksum themselves.

The text editor may be used on any type of text file, but is designed specifically for quick and easy editing of Assembly-language source files. It requires no line numbers, so you can use the arrow keys to position the cursor anywhere in the file. There are single-letter commands for insert, delete, change, move block, copy block, search, global change, and so on. An automatic repeat feature saves repetitive keystrokes.

A machine-language monitor is included, which allows display and modification of memory using a formatted hex display. Memory may be dumped to a printer or a text file. Breakpoints may be set, blocks of memory set or moved, and so forth.

Example programs on disk are included, as are 75 pages of documentation. Macro-80c requires a 16K or 32K Color Computer with disk drive; multiple drives are supported. Priced at $99.95, it is available from The Micro Works, P.O. Box 1110, Del Mar, CA 92014, (714) 942-2400.

Reader Service ✓555

## Finding Kindred Spirits

The Personal Computer Owners Directory can help you find people with the same interests or computer who are willing to trade information. The Models I and III and the Color Computer are listed, as are popular micros from Apple, Atari, Commodore, IBM, and others.

The directory lists 59 different interests, ranging from adventure and aerospace to utilities and word processing. Others are physics, commercial applications, education, data-base management, games, law, income tax, robotics, and many more.

For a free, permanent listing in the directory, send your name, address, computer type, interests, and whether you are willing to trade information to Personal Computer Owners, Dept. 1F, P.O. Box 426, Feeding Hills, MA 01030. To receive the current issue and the next issue with your listing, send $9.95 and the above information.

Reader Service ✓557

## Adventure Trilogy

Colorquest, a division of the Softlaw Corporation, has introduced a new line of fantasy games featuring full color, machine-language 3-D graphics, and sound.

The Adventure Trilogy takes you to the World Under the Cimeeon Moon to engage in ritual battle with tooamoath and narthokcs, then to Dazmor's Underworld of Doom to capture the Eye of Dazmor, and finally through the Ugrek Mountains to the Forsaken Gulch to restore the Eye to the Idol. Survivors are invited to try the sequel, Beyond the Cimeeon Moon, which involves a voyage to outer space.

The Adventure Trilogy and Beyond the Cimeeon Moon are available on disk for the Color Computer. Price is $29.95 each from Colorquest, P.O. Box 19096, Minneapolis, MN 55419, (612) 827-4703.

Reader Service ✓576


*Touch-Tone Modem*

## Touch-Tone Decoder Modem

The Touch-Tone Decoding Host Modem is a device for converting standard telephone Touch-Tones to ASCII equivalents. The modem has provision for a Bell 103 (300 baud) or Bell 202 (1,200 baud) data output response. An audio port for voice response is available as an option.

This product connects the direct-dial telephone network with any host computer capable of asynchronous communications via an RS-232 port.

The modem also features auto-answer circuitry. When called, it goes off-hook and awaits suitable input for two minutes. If no input is received, the modem automatically resets to on-hook, awaiting the next call.

When one of the 16 valid Touch-Tone pairs is received, it is converted to ASCII and sent in a serial stream to the host. The outbound responding data, from the RS-232 port, is used to modulate a frequency shift keyer (FSK). This permits transmission over the telephone network to the calling terminal.

Communication with the host RS-232 is at 300, 600, or 1,200 baud (depending on the model selected), via a standard DB-25 connector. Since the same baud rate is used for incoming

# SPECIAL-PURCHASE

## TRS-80 MOD III

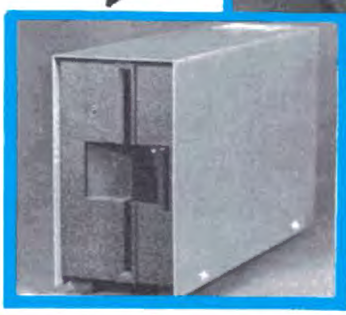
48K — 2 Drive — DISK III TRSDOS & Manuals

**$1775**

**6.3 to 38 MEG**

Winchester External Hard Disk III beginning at

**$1899**



**EXTERNAL FLOPPIES**

for MOD I or III **$275**

Fully compatible, 120 day warranty

| | |
|---|---|
| 80 TK or 2 sided | $395 |
| 80 TK & 2 sided | $515 |

**Optional HD III:**

| | |
|---|---|
| 9.5 meg | $2099 |
| 19 meg | 2799 |
| 2 x 6.3 meg | 2899 |
| 2 x 9.5 meg | 3199 |
| 2 x 19 meg | 4399 |

## CHECK THESE GREAT BUYS ON OTHER PRODUCTS:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Epson MX-80 w/graphtrax | $475.00 | C ITOH & Diablo Ribbons | 5.95 | Novation Cat Modem | 175.00 | Drive Extender Cable | 12.00 |
| Epson MX-80 ft | 575.00 | DC Hayes Smart Modem | 259.00 | Novation D Cat Modem | 199 00 | Head Cleaning Kit 5¼" | 14.95 |
| Epson MX-100 | 750.00 | DC Hayes Micro Model II | 339.00 | Percom Doubler II | 169.00 | Disk & Other Mysteries Book | 19 95 |
| MX-80 Ribbons | 12.50 | OKIDATA Microline 80 | 359.00 | Percom Data Separator | 29.95 | Basic & Other Mysteries Book | 27.50 |
| MX-100 Ribbons | 30.00 | OKIDATA Microline 82A | 529.00 | Paper 9½ x 11 (Fanfold) | 29.95 | NEC Ribbons (Min. 6) | 5.95 |
| Epson Graphtrax | 75.00 | OKIDATA Microline 83A | 799.00 | Paper 14⅞ x 11 (Green Bar) | 34.95 | Printer Cable | 25.00 |
| C.ITOH F-10 40 cps | 1,695.00 | OKIDATA Tractor | 75.00 | Verbatim 5¼" Diskettes | 26.90 | LDOS Operating Sys. | 129.00 |
| Tractor | 200.00 | Lexicon Modem | 125.00 | Maxell 5¼" Diskettes | 39.00 | DOSPLUS Operating Sys. | 149.00 |
| | | | | Nashua 8" Diskettes | 42.95 | | |

**NOW OPEN — VR DATA COMPUTER CENTER 616 Baltimore Pike, Springfield, Pa. (Delaware County)**

## DISK III

### SINGLE DRIVE $599

| | |
|---|---|
| Disk III Dual Drives | $864.00 |
| Disk III No Drives | 435.00 |
| TRSDOS & Manual | 21.90 |
| DOS PLUS & Manual | 149.00 |
| LDOS & Manual | 129.00 |

2 Sided or 80 Track Drives add $120. ea.
2 Sided & 80 Track Drives add $240. ea.

## MODEL III COMMUNICATIONS

**VR-RS232**
**$80**
Direct Replacement For R.S. RS232, Fully Tested & Burned-In, Easy Installation, 120 Day Warranty, Programmable Pinout, Prototype Area

**D-CON**
**$299**
Integral Modem, NEEDS NO RS232 — Direct Connect, Programmable Dialing, AUTO ANSWER/ ORIGINATE, Easy Installation

IF YOU DON'T SEE IT ADVERTISED CALL US AND ASK FOR IT.

**Call Toll-Free 800-345-8102**

VR data

777 Henderson Boulevard N-6
Folcroft, PA 19032 (215) 461-5300

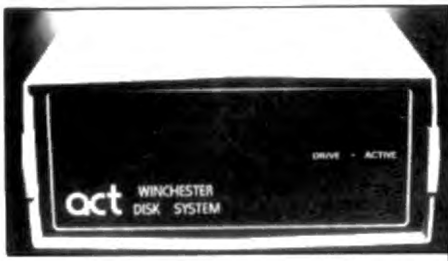
MasterCard VISA

ALL VR DATA PRODUCTS CARRY A 120-DAY WARRANTY INCLUDING PARTS & LABOR

Published prices reflect cash discount. All prices are subject to change without notice. TRS-80 and TRSDOS are trademarks of Tandy Corp. DISK III is a trademark of VR Data Corp.

8:30AM-7PM E.S.T. Mon.-Fri. Sat. 10AM-3PM
CABLE "VRDATA" TELEX 845 124

See List of Advertisers on Page 451




*ACT Winchester Drive*

and outgoing data, no special split baud rate software is required.

Priced at $500, the Decoding Host Modem is available from Micro-Peripheral Corp., Koll Business Center, 2643 151st Place N.E., Redmond, WA 98052, (206) 881-7544.

Reader Service ✓578

## Down Under Disks

The ACT Winchester Disk Drive, produced by the Australian Computer and Telecommunications division of Australian Protection Industries, New South Wales, offers five- and ten-megabyte storage capacity for Model II and III computers.

The hard-disk package includes a Winchester-type 5¼-inch drive with five megabytes of formatted mass storage and a microsequencing controller card with complete floppy-like interfacing. The controller end of the package has been designed to run a second 5½-inch hard disk, if desired.

Two Winchester hard-disk interfaces are also available: the TRS-80 Model II, designed to work with any Model II system and the ACT controller package, and the TRS-80 Model III for any 48K Model III system and the controller package. Both interfaces are fully software supported with a CP/M 2.2 operating system, and arrive fully assembled and tested for instant plug-in use.

Price and other information is available from the Office of the Australian Trade Commissioner, 636 Fifth Avenue, New York, NY 10111.

Reader Service ✓582

## Disk Listmaker

Disk Listmaker is an upgraded version of Manhattan Software's popular 16K tape program. It can store 500 names or items plus five-digit codes in 32K, or 1,000 names and codes in 48K.

In addition to fast machine-language alphabetical sorting, the program features selection and display or printout of names in any full code, or names in any group of codes. Any name or code may be edited, and any entry deleted.

Reported uses of the original program have included school pupil tracking, real-estate listing information, record and book cataloging, and many others.

The Disk Listmaker is available for Models I and III for $20 from Manhattan Software, P.O. Box 1063, Woodland Hills, CA 91365, (213) 704-8495.

Reader Service ✓588

## Consolidation System For VisiCalc Users

VIZ.A.CON for the Models I and III adds a third dimension to VisiCalc. You can combine multiple pages of VisiCalc data from a model for hierarchical consolidations (for example, department, division, company, and so on) or for summations over periods of time (for example, month, quarter, or year-to-date).

Typical uses are to combine weekly payroll data into monthly, quarterly, and yearly reports and to combine department budget data into division, region, and company-level reports.

A special system layout form helps you organize your consolidation systems. With the filled-in information, you can create a complete network during a question-and-answer session that allows you to define an unlimited number of interrelated consolidation processes. These definitions can then be stored for later use.

VIZ.A.CON gives you the ability to set up special formulas (such as ratios or percentages) that can be recalculated after a consolidation process. You can customize titles, row and column headings, footnotes, and so on for each report created. Reports can be automatically saved in standard ASCII form that can be used with a word processor.

Priced at $89.95, the program is available from Abacus Associates, Suite 240, 6565 W. Loop South, Bellaire, TX 77401, (713) 666-8146.

Reader Service ✓564

## Advice For Small Businesses

*Understanding and Buying a Small Business Computer* is a handbook of realistic, practical advice for small-business owners and managers interested in possible computerization. Written by consultant Susan Blumenthal, the $8.95 volume ranges from "Introducing an Average Computer" to determining whether your business needs a computer in the first place.

In addition, the book provides general information on microcomputers, minicomputers, and their peripherals' performance, operating costs, and maintenance and repair requirements. Other chapters help you decipher computer sales pitches, avoid employee hostility, and evaluate computer consultants.

It is available from Howard W. Sams & Co. Inc., 4300 W. 62nd St., P.O. Box 7092, Indianapolis, IN 46206, (317) 298-5400.

Reader Service ✓552

## Computer Care Kit

The RSI Computer Care Kit contains three useful products for maintaining a word processor and computer. The kit includes one eight-ounce bottle each of anti-static spray; print-wheel cleaner for removing ink and carbon build-up on Wang, Diablo, and Qume print wheels; and one spray bottle of screen and terminal cleaner. Also supplied are five foam-covered cleaning swabs, five lint-free cloths, and a print-wheel cleaning case.

The kit is available for $9.95 from Repair Service Inc., 4738 N. Ardmore, Milwaukee, WI 53211, (414) 963-0603.

Reader Service ✓573

## Name That Song

Name That Song is a game in which the player must guess the first word of


*Computer Care Kit*

the title of a well-known song. The player chooses a hard clue (fewer notes) or easy clue (more notes) for each song, and then hears the notes that allow him to identify it. The player must contend with a timer as well. Hard clues are worth 15 points and easy clues are worth 10 points, up or down. Two hundred points win the game amid great fanfare and dancing notes.

The game comes on cassette in two versions: Name That Song I—Children's Songs, and Name That Song II—Popular Hits. Each contains 72 familiar songs appropriate to the version. They are for a 16K Color Computer with Extended Color Basic and sell for $10 each.

For more information, contact Computer Island, 227 Hampton Green, Staten Island, NY 10312.

Reader Service ✓568

## TRS-80 Communications Program

Term/Host is a self-contained communications program for cassette or disk-based Model I or III systems with 48K of memory. The same version runs on all configurations.

The program handles all types of files in binary or ASCII modes. It supports complete remote operation of the user's system. As a host you may run machine-language or Basic programs, or up/download files from any remote terminal or system.

This menu-driven program comes with a 27-page user's guide for $34.95. For more information contact Freedom Financial Enterprises of Washington Inc., 3601 Carriage Drive, Raleigh, NC 27612, (919) 787-8082.

Reader Service ✓567

## Block Graphic Character Generator

MX-Graphic is a machine-language custom printer driver for the Models I and III with either an Epson MX-70 or MX-80 with Graftrax-Plus.

MX-Graphic generates standard TRS-80 graphic characters using the printer's bit-image mode and prints them in printer power-up configuration at between 1 and 2 cps. Normal text printing is not affected and graphics and text can be mixed as desired.

The block-character generator can be switched in and out by software to allow other printer configurations and use of the bit-image mode. NEWDOS-80 JKL and Model III screen-dump functions are supported.

The program is available for all standard tape and disk configurations. In addition, versions are provided for Model Is with a Memory Sidecar manufactured by Displayed Video. Memory overhead is less than 800 bytes and the program is self-protecting in Basic operation.

With documentation included, MX-Graphic sells for $19.95 (U.S.) or $24.95 (Canadian) plus $2 postage and shipping. For more information, contact CMD Micro Computer Services Ltd., 10546-106 St., Edmonton, Alberta, Canada T5H 2X6, (403) 423-3919.

Reader Service ✓570

## *Computer Carnival*

*Computer Carnival* by Richard Ramella contains 60 Level II Basic programs, written primarily for children but useful to beginners of all ages. It includes both games and educational programs, arranged in order of difficulty. Most of the programs are short, making them easy to enter. Some can be entered by an experienced keyboarder and then used by children too young to read.

Each chapter has an illustration by the author, a short description of the program and how to use it, and a program listing. The games range from favorites like keno and bingo to original games and simulations, as well as graphic games. The educational games include math, letter guessing, and word games; a program teaching states and capitals; and even a Morse-code tutor.

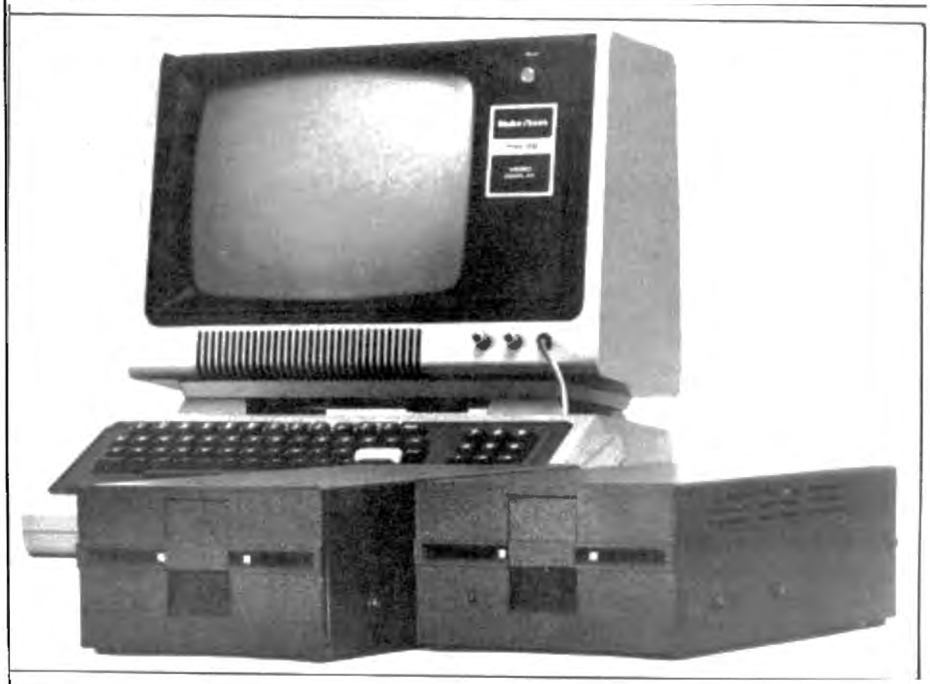
*Computer Carnival* costs $16.97 at computer stores or directly from the publisher, Wayne Green Books, Peterborough, NH 03458. A package containing the book and the programs on cassette is available for $24.97 and the cassette alone for $9.97, although the book is required for documentation.

Reader Service ✓556

## Expanded Disk Storage For Models I and III

A series of floppy-disk drives for both internal and external mounting, offering up to four megabytes of unformatted storage for the Model III and two megabytes for the Model I, is now available from Interface Inc.

For internal storage on the Model III, there are four drives available. These include a single-sided drive with 40 tracks and 250K of unformatted storage for $215, a single-sided drive with 80 tracks and 500K for $335, a double-sided drive with 40 tracks and 500K for $335, and a double-sided drive with 80 tracks and one megabyte of unformatted storage for $435. A



*Interface Disk Drives*

maximum of two drives can be installed internally.

There are also four drive types available for external storage with the Models I and III, including color-coordinated cabinet, power supply, and cable. These drives are single sided with 40 tracks and 250K for $315, single sided with 80 tracks and 500K for $435, double sided with 40 tracks and 500K for $435, and double sided with 80 tracks and one megabyte of storage for $535.

Cabinet colors include computer blue, standard off-white, lime green, dark green, bright orange, brilliant yellow, black, or chrome.

All drives are TRS-80 compatibility tested before shipment and have a 90-day warranty. For more information, contact Interface Inc., 7630 Alabama Ave., Unit 3, Canoga Park, CA 91304, (213) 341-7914.

Reader Service ✓565

## Single-Entry Ledger System

The Single-Entry Ledger System provides a menu-driven, easy-to-use general-ledger package for a cash-basis accounting system. This product maintains its own data files and provides a variety of reports. The programs, written in Extended Basic, are available for the Model III and the Color Computer.

The system's data files can contain any number of accounts and transactions, limited only by the size of the storage medium. Files may be added, deleted, or edited at any time. Reports include accounts in numerical order, comparison of year-to-date values with previous year, transactions by account number, and many more.

Priced at $95, the Ledger System is available from Universal Data Research Inc., 2457 Wehrle Drive, Buffalo, NY 14221, (716) 631-3011.

Reader Service ✓551

## Dual-Mode Joystick

A new plug-in joystick for Color Computer games offers both traditional spring-center return and a faster, more versatile free-floating mode.

A linear potentiometer translates the movement of the gimbal to electrical impulses, providing more accurate cursor control and faster motion. External switches allow players to shift back and forth between modes as often as they like, while cursor centering tabs on top of the unit permit fine tuning for precise response.

The joystick is available for $64.95 from Kraft Systems Co., 450 W. California Ave., Vista, CA 92083, (714) 724-7146.

Reader Service ✓586

## Faster Disk Thesaurus

Refware Thesaurus 2.0 is a faster and expanded synonym-finder that replaces the first Refware Thesaurus published in 1981.

Compared to the earlier program, Thesaurus 2.0's ZBasic 2.2 machine language and new index feature allow it to respond more than 10 times faster—on the Model III, only seven seconds from the time a user asks about a given word until the program supplies a group of synonyms or related words. Vocabulary has been expanded to 24,800 entries, providing nouns, adjectives, verbs, and adverbs for all types of writing.

Thesaurus 2.0 requires two disk drives and 48K of memory, and is available in three formats: a four-disk set for Model III, 40-track, double density; a five-disk set for Model I, 35-track, double density; and a nine-disk set for Model I, 35-track, single density. The price of each is $89.95 plus $3 for shipping.

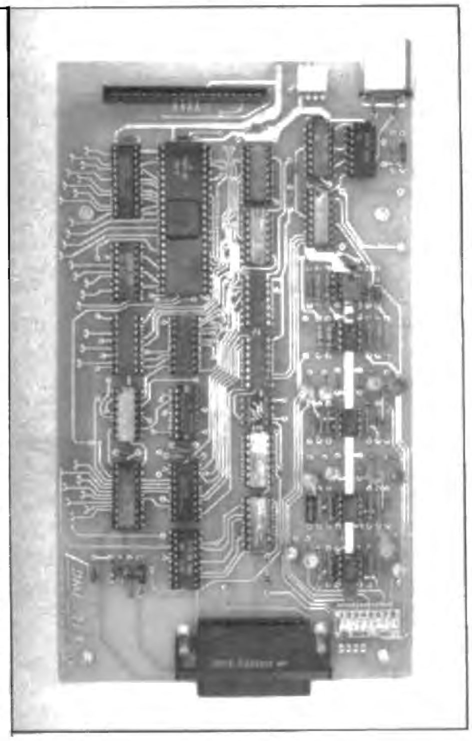
Through December 31, 1982, registered purchasers of Thesaurus 1.0 can exchange their disks for the new program for a $30 service fee and $3 shipping. For more information, contact Refware, David C. Whitney Associates Inc., P.O. Box 451, Chappaqua, NY 10514, (914) 238-8896.

Reader Service ✓579

## Camp Management System

EZ-Camp is an integrated set of programs designed exclusively for camp management. The system was developed and field tested in conjunction with several camps using the Model II, and requires no programming knowledge or programmer support.

Besides maintaining complete camper information, the system can produce mailing lists, medical records, birthday or bunk lists, and many other reports required in camp management. Users can select standard reports, create their own, or use the system's word processor for personalized letters, application forms, newsletters, and the like. In addition to standard financial applications, EZ-Camp also has capabilities aimed specifically at camps—like canteen supervision and cash advances for staff.



*Computex Communications Board*

A demonstration disk can be obtained for $50. For more information, contact EZR Micro Inc., 11 Conwell Drive, Maple Glen, PA 19002.

Reader Service ✓560

## Model III Internal Communications Board

The M3CB1 communications board provides an RS-232C serial interface and a complete direct-connect modem all on one card slightly larger than the Radio Shack RS-232 interface. Totally software and hardware compatible with the Model III computer, the M3CB1 installs in less than 30 minutes in the same location as the Radio Shack RS-232 board.

The RS-232 supports asynchronous serial transmissions and conforms to the industry-standard EIA RS-232C interface. Baud rates from 50 to 19,200 baud are software selectable.

The 300-baud, full-duplex, direct-connect modem uses FSK modulation techniques and supports both originate and answer modes of operation.

The assembly and installation manual provides step-by-step assembly instructions written for the first-time

# DOES STRING COMPRESSION HAVE YOU TIED UP IN KNOTS?


## LET TRASHMAN CLEAN UP THE MESS!

TRASHMAN is a machine language utility for the TRS-80 Models I and III. It was written by Glenn Tesler, the author of FASTER, and can reduce BASIC's string compression time by 95% (see table below).

### *WHAT'S STRING COMPRESSION?*

When a BASIC program changes a string (words, names, descriptions), it moves it to a new place in memory, and leaves a hole in the old place. Eventually, all available memory gets used up and BASIC has to push the strings together to free up some space. This takes time. Lots of time. The computer stops running for seconds or minutes, and you may even think it's "crashed". The keyboard won't work, and until all the strings have been collected, you just have to sit and wait. Then things run for a while, until string compression is needed again. And again.

If you're using your computer for business, that wastes your money. If you're using it personally, it wastes your time.

### *WHAT'S THE SOLUTION?*

As soon as you start using TRASHMAN, those delays almost disappear. It uses less than 600 bytes of memory, plus 2 bytes for each active string. It works with other machine language programs and with all major operating systems. It's easy to use, comes with complete instructions, and can be copied to your own disks.

### *WHAT'S THE CATCH?*

If a BASIC program uses only a few strings, very little time is wasted in string compression, and TRASHMAN won't be helpful. But, if hundreds of strings, including large string arrays, are used, TRASHMAN is just what you need.

| # | SECONDS DELAY | | PERCENT |
|---|---|---|---|
| STRINGS | NORMAL | TRASHMAN | IMPROVEMENT |
| 250 | 11.8 | 0.7 | 94 |
| 500 | 45.8 | 1.6 | 96.5 |
| 1000 | 179.6 | 3.5 | 98 |
| 2000 | 713.2 | 7.8 | 98.9 |

Ask your software dealer for TRASHMAN, or order directly on our toll-free number. The price is just $39.95 (plus sales tax in California).

(All timings done on TRS-80 Model I. Model III 15% faster, but pct. improvements identical. Listing of timing program available on request.)

## AMAZING PROGRAM SPEEDS UP BASIC

Your time is valuable, so why waste it on slow-running BASIC programs? PROSOFT's "FASTER" will analyze those programs *while they run*, then show you a simple change (usually one new line) that can reduce run-times by up to 50%.

Accounting systems, financial models, engineering and scientific programs all run faster; so do games. Large, complex programs improve the most, and "FASTER" is easy to use.

THIS ISN'T A COMPILER! Your BASIC programs remain readable and can be changed later on. While your programs run, "FASTER" counts how often each "variable" is used, then shows you the correct sequence for these variables. Afterwards, the computer finds them sooner, so your programs run faster.

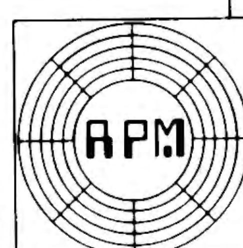
Does it really work? Yes! *Personal Computing* said so in their May, 1981 issue (p. 116); we've received many letters from customers who've gotten 20-50% improvements; and we will make you this offer:

Order "FASTER" now. Try it on your bread-and-butter programs. If you don't get an overall run-time reduction of at least 20%, return it within 30 days for a prompt and cheerful refund.

"FASTER runs on TRS-80 Models I and III, 16-48K, tape or disk

**$29.95**

## ERRATIC DISK DRIVES




**RPM** measures the rotational speed and variation of your disk drives, and reveals a common cause of unexplained errors. Simple one-key operation, runs under any DOS, interchangeable between Models I and III. Shows current and average speeds, plus fluctuation history. Recovers from severe errors. Documentation explains how to adjust drives. Use RPM monthly for best results. 32-48K Model I or III disk **$24.95**

## QUICK COMPRESS

Small (276 bytes), fast (processes 800 lines in under 3 seconds) utility removes blanks and remarks from your BASIC programs. Produces smaller, faster programs, and doesn't alter the original logic.

16-48K Model I or III, tape or disk. **$19.95**

**SPECIAL** FASTER + QUICK Compress: **$39.95**

**TO ORDER, CALL NOW, TOLL FREE**

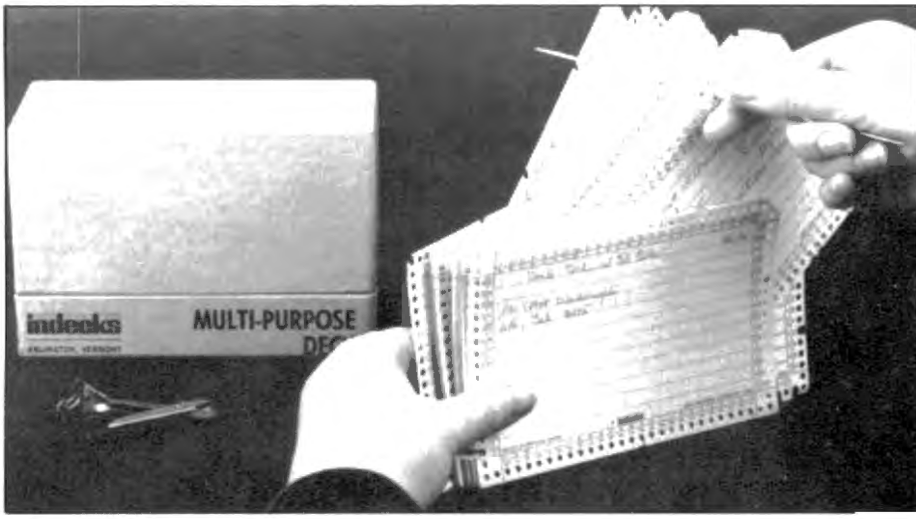
(800) 824-7888, Operator 422
CALIF: (800) 852-7777, Oper. 422
ALASKA/HAWAII: (800) 824-7919
FOR TECHNICAL INFORMATION CALL:
(213) 764-3131, or write to us.

**PROSOFT**™

Dept. G Box 560
North Hollywood, CA 91603

91

TERMS: We accept VISA, Mastercard, checks, Money Orders, C.O.D. and even cash. We pay shipping via surface UPS inside U.S.A. Please add $3.00 for Blue Label, 6% tax in California, and 15% outside North America (air shipment).




*Datasort*

kit builder and includes installation pictures for each major step, theory of design, user information, and much more.

The M3CB1 is available as just the printed circuit board and manual (you supply components and assembly) for $34.95, or as the complete kit for $129.95. The RS-232 is also available assembled and tested for $69.95, while the complete RS-232 and direct-connect modem comes partially assembled (you must insert one IC into a socket) for $169.95. For more information, contact Computex, 17321 El Camino Real, Houston, TX 77058, (713) 488-8022.

Reader Service ✓553

## Disk Menu and Directory

Most disk directories show only the filespecs and some seldom-needed other information. The Disk Menu and Directory by International Computer Products displays detailed descriptions of each program on the disk, as well as filespecs, date of creation, and more useful data.

The directory works without modification on Models I and III with TRSDOS, NEWDOS+, or other operating systems. Priced at $29.95 plus $1.50 shipping and handling, it is available from International Computer Products, 346 N. Western Ave., Los Angeles, CA 90004.

Reader Service ✓577

## Reading Skills Lab

Reading Skills Lab is a new product for the Model III that offers a variety of exercises for improving basic reading skills, as well as short stories for reading practice. The complete lab contains 63 programs on six disks and three teacher's manuals with complete documentation.

The set was designed by a certified reading specialist, and the stories were written by professional children's authors. The core vocabulary lists for each grade level were taken from a list of words introduced in the most widely used basal reading series in schools today.

Exercises include word recognition, sight vocabulary, definitions, vocabulary in context, recall of detail, inference, main idea, sequence of events, and many other skills. Most programs include an optional printer module that allows preparation of worksheets for desk use. Each grade level also has at least one word game using the core vocabulary list to spark interest and reinforce skills.

The complete Reading Skills Lab, with exercises for students in grades 1–9, sells for $1,000. The programs can also be ordered in smaller groupings: Lab A (grades 1–3), Lab B (grades 4–6), and Lab C (grades 7–9), priced at $350 each. They can be ordered from NTS Software, 680 N. Arrowhead Ave., Rialto, CA 92376.

Reader Service ✓571

## "Cardboard Computer"

Datasort, a "cardboard computer," allows you to store and retrieve information in any category combination you choose.

The system consists of a hand punch, a small rod (like a knitting needle), and file cards with numbered holes around all four edges. You develop your own category list indicating what you want the numbered holes to mean, then use your hand punch to clip away the tops of holes that apply to the information on each punchcard. To retrieve information, insert the rod at any category hole. When you lift the rod and shake the stack, cards notched for that category will fall off the rod.

To cross reference, all you need do is take the cards just retrieved and poke the rod through a hole for any other category.

The Datasort system is designed for medium-sized projects requiring up to 5,000 cards. Its cross-referencing capability can save up to 80 percent of the time usually spent duplicating and scanning, and 100 percent of the time spent refiling.

The system weighs less than three pounds and is priced at under $30. For more information, write or call Indecks Inc., Dept. 523, Arlington, VT 05250, (802) 362-3464.

Reader Service ✓589

## Loan Sales/Purchase Analysis Program

A new program for analyzing the sale or purchase of a loan is available on disk for the Model I or III with 48K.

The program will calculate either the return on investment or the value of loans of any amount, at any interest rate or due date. It allows both buyer and seller to make knowledgeable decisions when purchasing or liquidating loans.

The printout shows the loan amount, original interest rate, the number of months until due, the yield to the buyer (return on investment), the discount interest rate, the discount amount, and the value of the loan (price paid to seller). Loans in foreign currencies as well as U.S. dollars can be analyzed.

The program sells for $50 from Realty Software Co., 1116 'E' 8th St., Manhattan Beach, CA 90266, (213) 372-9419.

Reader Service ✓554

## Cables and Light Pen

Five new cables for the Color Computer are available from Spectrum Projects.

The Disk Interface/ROMpak Extender ($29.95) is a 40-pin ribbon cable that plugs into the ROMpak port and terminates with a female connector for disk interfaces and ROMpaks. Connectors are gold plated to eliminate corrosion. This three-foot cable eliminates disk I/O errors and makes arranging your components easier.

Three 10-foot extension cables let you place your joysticks, printer/modem, and tape recorder at convenient distances from the computer. They are $14.95 each. The last cable, a joystick Y adapter ($19.95), lets you hook in line a joystick and Spectrum's new low-cost light pen ($14.95).

These products can be ordered from Spectrum Projects, 93–15 86th Drive, Woodhaven, NY 11421, (212) 441-2807.

Reader Service ✓569

## Files and Labels

Management System II, a filing program featuring machine-language "lightning sort" of any column, is now available for Model I or III tape-based systems.

File presentation is a maximum of two horizontal lines (100-plus characters) with one to nine vertical columns. Column titles are easily created, and files may be added, updated, listed, deleted, or saved to tape or printer.

A labels command provides printer output of any or all columns for production of mailing labels, file labels, stock bin tags, or any similar home or office application. A merge feature consolidates files previously saved on tape, and a keyed selective-search function lists files based on the number of characters input.

All commands are executed by single-letter entry for faster and easier operation. The program sells for $26.95 from Micro-80 Inc., 2665 E. Busby Road, Oak Harbor, WA 98277.

Reader Service ✓583

## Animation Stand Move Calculator

Animcalc is a program designed to enable the users of hand-cranked animation stands to produce the same smooth, tapered moves generated on the new breed of computer-driven stands. Given the starting and ending settings of up to six different coordinates, and the frame number at the start and end of the move, Animcalc computes the coordinate settings for each frame within the move, including table rotation (clockwise or counterclockwise).

Each coordinate can be assigned its own taper style (no taper, taper start, taper end, or taper both ends), and different lengths of taper can be assigned to each coordinate. The results can be output to a line printer or the video screen for copying to the animation frame chart.

The program is available in CP/M or Basic for Models I and III. It is priced at $49.95 (disk or cassette). For more information, write Animcalc, 895 NE 90th Ave., Portland, OR 97220.

Reader Service ✓559

## First Down

NFL Pro Football allows play between any NFL teams in either manual or automatic mode. It uses actual NFL rosters and team statistics that may easily be updated as the season progresses. A predict mode produces amazingly accurate forecasts. Fast machine-code graphics add excitement to the game.

This game runs on a 48K Model I or III. Priced at $34.95, it is available from M-P-Software, Route 2 Box 533, Cumming, GA 30130, (404) 887-6814.

Reader Service ✓575

## Bulletin Board

SPS-BBS is a low-cost bulletin-board system for the Models I, II, and III. The system sells for $79.95 ($119.95 for the Model II) and includes a host utility, message system, packed disk files, six functions that can be defined by the system operator, chat mode, and many more.

For more information contact Small People Software Corp., P.O. Box 732, College Park, MD 20740, (301) 937-4339.

Reader Service ✓587

## Library Management System

The Library Support Option file-management system provides the capability to create and maintain libraries for logical organization of files and programs.

The system is compatible with the DOS, user, and application programs. Regular DOS commands can be used to list, print, kill, copy, load, or save files directly to or from libraries. It supports regular as well as library files, and includes faster file handling, one sector minimum file size, and text compression.

Priced at $79, the Library Support Option is available on disk for the Models I and III from XYZT Computer Dimensions Inc., 2 Penn Plaza, Suite 1500, New York, NY 10121, (212) 244-3100.

Reader Service ✓563

## Koupon Keeper

Koupon Keeper is a record-management system for the Models I and III that catalogs cash-off coupons in any order and displays them, at your request, by brand name, product type, amount, or expiration date. This product helps you make better, more effective use of store coupons than is possible using the traditional filing techniques. It also greatly reduces the amount of time necessary to maintain your coupon file.

Priced at $9.75, it is available from Kensoft, 2102–50th St., Kenosha, WI 53140, (414) 654-2722.

Reader Service ✓558

## Graph Paper for Your Word Processor

Word-Processor Graph Paper is a software product that produces screen formats of standard graph paper grids compatible with word processor storage formats.

By selecting one of over 30 stored formats, the user need only enter data points and graph titles. The graph then becomes part of the text file and is edited and printed as a normal page.

For more information, send word-processor or computer type to ATC Software, RT 2 Box 448, Estill Springs, TN 37330.

Reader Service ✓591

*New Products listings are based on information supplied in manufacturers' press releases.* 80 Micro *has not tested or reviewed these products and cannot guarantee any claims.*