January 1985
USA $4.00

80 micro

A CWC/I PUBLICATION

the magazine for TRS-80* users

Speed Demon

The 80 Micro Basic Compiler

Model 4 Scripsit Gets an Overhaul

NovaCalc: A Basic Spreadsheet Treat

Thirteen Patches To TRSDOS 1.3

How to Get More From Your DMP

Also:
The Next Step
BBS Express
Basic Takes
Project 80
2000 Plus

0 74820 08278 01

80micro 1/85 #60 B

...GIVE YOUR MODEL 4 NEW DOUBLE-DUTY AND TAKE IT EASY!

Introducing Profile 4 Plus

Our new Profile 4 Plus (26-1635, $249.95) makes advanced data base management available to Model 4 owners. It's based on the same powerful system already used by Model 12 and Model 16 users. Profile 4 Plus offers you plenty of options for data manipulation, so you can personalize your filing system.

Add Model 4 Word Processing

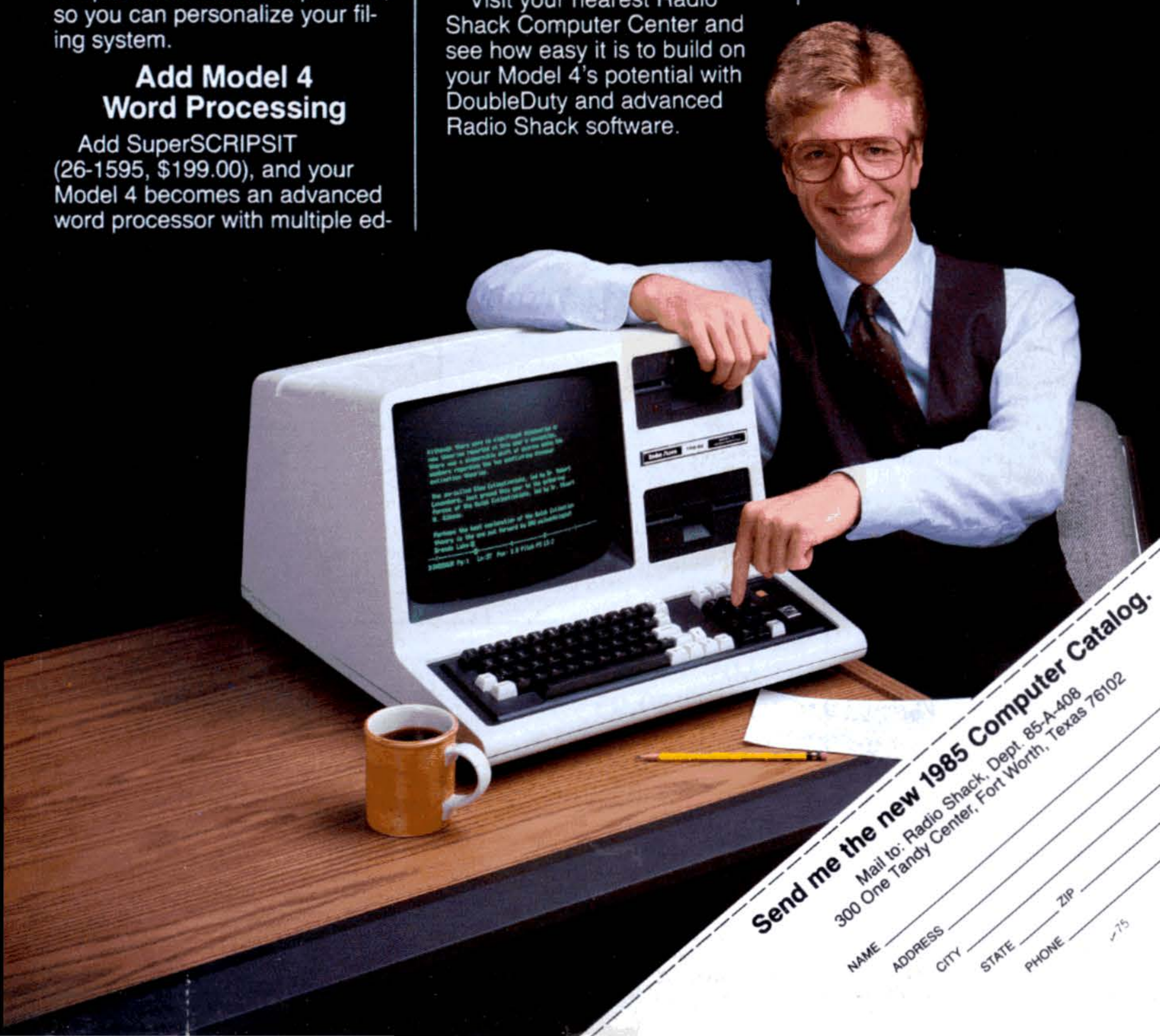
Add SuperSCRIPSIT (26-1595, $199.00), and your Model 4 becomes an advanced word processor with multiple editing features that let you delete, insert and move blocks of text easily. SuperSCRIPSIT supports proportional-spaced printing, multiple line spacing, underline, boldface, super and subscripts, multiple columns, 80 × 24 display and reverse video to aid in editing.

See All Three Today

Visit your nearest Radio Shack Computer Center and see how easy it is to build on your Model 4's potential with DoubleDuty and advanced Radio Shack software.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

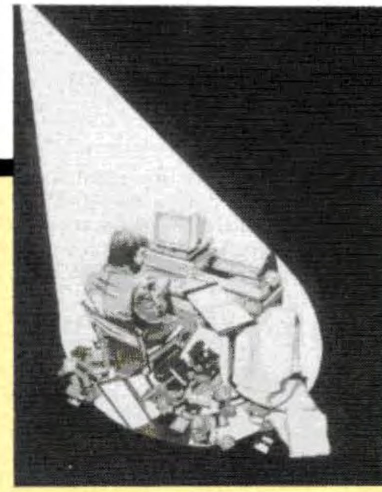
Double Duty requires 128K Model 4. All other programs require 64K Model 4. Prices apply at participating Radio Shack Computer Centers and at participating Radio Shack stores and dealers.



Send me the new 1985 Computer Catalog.
Mail to: Radio Shack, Dept. 85-A-408
300 One Tandy Center, Fort Worth, Texas 76102

NAME ______
ADDRESS ______
CITY ______
STATE ______ ZIP ______
PHONE ______

Owning a computer program that writes programs for you may seem like a science fiction dream. But that's exactly what The Producer does. You can now enjoy professional quality programs, custom written for your specific needs!



Here's a sampling of what Producer users are saying:

"This program has paid for itself over and over." People tell us this repeatedly. First time computer users have successfully been able to generate programs with The Producer that rival the professional quality of programs available through commercial sources.

"There are only two things you need to say about The Producer: it's easy to use and it's powerful!" The woman who told us this had used several computers before but had no programming experience. Now she is saving time, money and effort by creating the programs she needs for her office.

"Owning The PRODUCER is like having a professional programmer on call-24 hours a day. I particularly like the fact that if I design a program to do a specific job today, and later discover it is inadequate to my changing needs, I can regenerate the program with corrections in a matter of minutes." That's right. The PRODUCER gives you the ability to edit or expand your programs in any way you choose.

"The screen generator alone is worth the price! I created a professional data entry screen in minutes just after opening the package." With our screen generator, you are the artist and you are never locked into what someone else has designed for you.

"How did I ever live without it." This often heard quote comes from experienced programmers who are now using The Producer to generate programs in a fraction of the time it previously took to them to write programs.

"The Freeform Report Generator is an amazing feature. I never thought a product like this could give me the ability to get reports on my preprinted forms, but The Producer does it all." Yes, you can put the text anywhere on the page you want, do up to 100 interfield calculations, etc. And the Freeform is now a part of The PRODUCER package.

OTHER QUOTES FROM PRODUCER USERS

"The PRODUCER package I received was excellent. The finest software package I have ever purchased. Far beyond my expectations."
S.R. Foster, Pensacola, Florida

"I'm in love with The PRODUCER. It's one of my favorite programs."
R. Selsback, Burlingame, California

"The value of the deal, everything included, was the best I've seen to date."
G. Slusher, Martin, Kentucky

"Excellent! Above and beyond other software."
R. Hapgood, Henrietta, Texas

★

"The PRODUCER is the best all purpose program generator I have used. (We have tried almost all of them). The generated code is bug free, well commented and efficient."
A. Copella, Northbrook, Illinois

"This is by far my number one software and I will use it anywhere and everywhere I possibly can, both personal and in business."
R.A. Neuman, Okemos, Michigan

"The program I created with The Producer meets my needs exactly and I can change it easily as different conditions require. You can't do that with other software."
Neal Bloomenfader

"One of the best I've seen. We write about 20 volumes of material per year. Take it from a pro, it's good."
J. Crespi, Sherman Oaks, California

★

"Thank you for an excellent program. I agree that The PRODUCER will change the entire concept of program creation in the future. But for now, you stand as the best data base management system I can buy."
E. Sung, Vancouver, B.C.

"I think The PRODUCER will make the software hackers upgrade their products to this high level quality of the PRODUCER. I'm sure you realize that there is a lot of garbage on the market."
D.J. Smith, Lombard, Illinois

"The Producer is among the best systems I have used in ten years of professional data processing."
Dennis R. Cutshall, APO NY.

New from Producer Software
THE DATA SHUFFLER $34.95

All purpose Sort/Merge program

Many of you have been asking for it, so here it is. With **Data Shuffler** you can perform powerful sort and merge functions in conjunction with your data files. • Sorts up to 32767 Records • Sorts key size up to 256 bytes • Does up to 9 sort keys in ascending and/or descending order • Includes a complete sort parameter builder program, etc.

Data Shuffler is available now for TRS-80 Models I,III, and 4. Call us to place your order for immediate shipment.

The PRODUCER

The Professional Program Writer 76

TRS-80 Model I Version
TRS-80 Model III Version
TRS-80 Model IV Version

$199.95

(including Model 4-P)

1-800-433-5355 **Texas (817) 274-6998**

We accept money orders, cashier's checks, VISA, Master Card and American Express.

PUBLISHER
Peter Hutchinson

EDITOR-IN-CHIEF
Eric Maloney

MANAGING EDITOR (EDITORIAL)
Peter E. McKie

MANAGING EDITOR (PRODUCTION)
Deborah M. Sargent

REVIEW EDITOR
Ryan Davis-Wright

COPY EDITORS
Robert Mitchell (New Products)
Penelope Hamblin

ASSISTANT EDITOR
Catherine Geary

TECHNICAL EDITORS
Bradford N. Dixon
Marc-Anne Jarvela
Keith Johnson (Load 80)
Dave Rowell
Beverly Woodbury

EDITORIAL ADMINISTRATION
Carole Macioci

ASSOCIATE EDITORS
Hardin Brothers
David Engelhardt
John B. Harrell III
Terry Kepner
Thomas L. Quindry

ADVERTISING SALES

SALES MANAGER
William Smith

SALES REPRESENTATIVES
Richard J. Alden
Michael Wozmak
1-800-441-4403

WEST COAST OFFICE
1060 Marsh Road
Menlo Park, CA 94025
415-328-3470

SALES MANAGER
Giorgio Saluti

SALES REPRESENTATIVES
Allison Walsh
Karen Letendre

Cover by Joe Heiner

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 "How to Write for *80 Micro*." Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased.

*TRS-80, Scripsit, and TRSDOS are trademarks of Radio Shack, a division of Tandy Corp.

80 Micro (ISSN 0744-7868) is published monthly by CW Communications/Peterborough 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 $24.97 for one year, $38 for two years, and $53 for three years. In Canada and Mexico $27.97—one year only, U.S. funds drawn on a U.S. bank. Nationally distributed by International Circulation Distributors. 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 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 address changes to *80 Micro*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.

January 1985 **80micro**

page 42

page 60

page 82

On the Cover

Features

Departments

LOAD 80

Load 80 gathers together selected programs from this issue of *80 Micro* and puts them on a magnetic medium for your convenience. It is available on tape or disk, and runs on the Models I, III, and 4.

If you own a tape system, load the Load 80 tape as per the instructions provided. If you own a Model I or III disk system, boot the Load 80 disk and transfer the files to a TRSDOS system disk according to simple on-screen directions. If you own a Model 4, you must convert the programs from Model III TRSDOS to Model 4 disk using the Model 4 CONV command.

Not all programs will run on your system. Some Model III programs, for instance, will run on the Model 4 in the Model III mode, but not in the Model 4 mode. You should check the key box that accompanies the article to find out what system configuration individual programs require.

If you have any questions about the programs, call Keith Johnson at 603-924-9471. Yearly subscriptions to Load 80 are $199.97 for disk, or $99.97 for cassette. Individual loaders are available on disk for $21.97 or on cassette for $11.47, including postage. Direct subscription problems or orders for Load 80 to Lori Eaton, c/o *80 Micro*, 80 Pine St., Peterborough, NH 03458.

Directory

FastBas
Article: Running Like the Wind (p. 42)
System: Models I and III, 32K RAM.
Language: Basic
An updated version of *80 Micro*'s Basic compiler.
Cassette filespec: B, C, D, E
Disk filespec: FASTBAS/BAS, LIST2/BAS, LIST3/BAS, PONG/BAS.

ScripAid
Article: Model 4 Scripsit the Write Way (p. 60)
System: Model 4, 64K RAM, one disk drive.
Language: Assembly/Basic
An enhancement package for Model 4 Scripsit. Disk filespec: ENHANCE/OVL (system), REKEY/OVL (system), PRTDRIVE/BAS, HELPBLD/BAS. Requires editor/assembler.

NovaCalc
Article: NovaCalc (p. 82)
System: Models I and III, 48K RAM, one disk drive.
Language: Basic
A Basic spreadsheet program.
Disk filespec: NOVACALC/BAS.

Graphics
Article: Picture Perfect (p. 98)
System: Models I, III, and 4, 16K RAM.
Language: Basic
Create your own graphics characters on your Radio Shack dot-matrix printer.
Cassette filespec: J, K, L, M, N, O, P.
Disk filespec: GRAPAPER/BAS, CHARGEN1/BAS, CHARGEN2/BAS, STRINGS3/BAS, SUBGRAF4/BAS, SUBGRAF5/BAS, SUBGRAF6/BAS.

Random Files
Article: On the Record (p. 106)
System: Models I and III, 32K RAM.
Language: Basic
Three programs that create and read random-access disk files.
Cassette filespec: G, H, I.
Disk filespec: RANDISKU/BAS, RANDISK1/BAS, LOOKDISK/BAS.

Cubic
Article: Formula Solutions (p. 116)
System: Model I and III, 16K RAM cassette, 32K RAM disk; Models 4, 1000, 1200, 64K RAM, one disk drive.
Language: Basic
Programs for solving cubic, quadratic, and linear equations.
Cassette filespec: Q, R.
Disk filespec: CUBIC1/BAS, CUBIC3/BAS, CUBIC4/BAS.

Interrupt
Article: Project 80 (p. 120)
System: Models III and 4, 16K RAM
Language: Assembly
Demonstration program: How to use the 8259A interrupt controller with the Project 80 general-purpose I/O board.
Cassette filespec: INTRPT (source code). Requires Apparat editor/assembler.
Disk filespec: INTRUPT/SRC (source code). Requires Apparat editor/assembler.

BBS
Article: BBS Express (p. 132)
System: Model III, 48K RAM, two disk drives.
Language: Assembly/Basic
A routine for locating message numbers.
Cassette filespec: BBS1/(source code), S. Source code requires editor/assembler.
Disk filespec: BBS1/SRC (source code), BBS2/BAS. Source code requires editor/assembler.

Demo
Article: The Next Step (p. 140)
System: Model III, 32K RAM, one disk drive.
Language: Assembly
Demonstration program to create a beep generator.
Cassette filespec: DEMO1, DEMO2, DEMO3 (all source code).
Disk filespec: DEMO1/SRC, DEMO2/SRC, DEMO3/SRC (all source code). Requires Apparat Editor/assembler.

ART DIRECTOR
Beth Krommes
PRODUCTION SUPERVISOR
Dion Owens
PRODUCTION ASSISTANT
George Gardos
AD/GRAPHICS PRODUCTION
Gary Ciocci

PRODUCTION MANAGER
Joyce Pillarella
MANUFACTURING CONSULTANT
Susan Gross
TYPESETTING MANAGER
Dennis Christensen
FILM PRODUCTION MANAGER
Robert M. Villeneuve
DARKROOM MANAGER
Nathaniel Haynes

VICE PRESIDENT/GENERAL MANAGER
Debra Wetherbee
VICE PRESIDENT/FINANCE
Roger Murphy
ASSISTANT GENERAL MANAGER
Matt Smith
ASSISTANT TO VP/FINANCE
Dominique Smith
DIRECTOR OF MARKETING SERVICES
Pamela Esty
DIRECTOR OF CIRCULATION
William P. Howard
ASSISTANT CIRCULATION MANAGER
Frank Smith
DIRECT AND NEWSSTAND SALES MANAGER
Raino Wirein
1-800-343-0728
DIRECTOR OF CREDIT SALES AND COLLECTION
William M. Boyer
EXECUTIVE CREATIVE DIRECTOR
Christine Destrempes
PUBLIC RELATIONS
Michele Christian

FOUNDER
Wayne Green

80 Micro is a member of the CW Communications/Inc. group, the world's largest publisher of computer-related information. The group publishes 52 computer publications in 19 major countries. Nine million people read one or more of the group's publications each month. Members of the group include: Argentina's *Computerworld/Argentina*; Australia's *Australia Computerworld*, *Australian Micro Computer Magazine*, *Australian PC World and Directories*; Brazil's *DataNews* and *MicroMundo*; China's *China Computerworld*; Denmark's *Computerworld/Danmark* and *MicroVerden*; Finland's *Mikro*; France's *Le Monde Informatique*, *Golden* (Apple) and *OPC* (IBM); Germany's *Computerwoche*, *Microcomputerwelt*, *PC Welt*, *Software Markt*, *CW Edition/Seminar*, *Computer Business* and *Commodore Magazine*; Italy's *Computerworld Italia*; Japan's *Computerworld Japan* and *Perso ComWorld*; Mexico's *Computerworld/Mexico* and *CompuMundo*; Netherlands' *CW Benelux* and *Micro/Info*; Norway's *Computerworld Norge* and *MikroData*; Saudi Arabia's *Saudi Computerworld*; Singapore's *The Asian Computerworld*; Spain's *Computerworld/Espana* and *MicroSistemas*; Sweden's *ComputerSweden*, *MikroDatorn* and *Min Hemdator*; the UK's *Computer Management* and *Computer Business Europe*; the U.S.' *Computerworld*, *HOT CoCo*, *inCider*, *InfoWorld*, *MacWorld*, *Micro Marketworld*, *PC World*, *Run*, *73 Magazine* and *80 Micro*

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.
Problems with Load 80 Circulation: Address correspondence to Lori Eaton, 80 Pine St., Peterborough, NH 03458.
Problems with Advertisers: Send a description of the problem and your current address to: *80 Micro*, Rt. 101 & Elm Street, Peterborough, NH 03458, ATTN.: Rita B. Rivard, Customer Service Manager. If urgent, call 1-800-441-4403.
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 Raino Wirein, Retail Sales Manager, *80 Micro*, Pine St., Peterborough, NH 03458. (800) 343-0728.

Tandy Charts a New Course

I've recently heard some complaints from long-time TRS-80 owners that Tandy is selling out by stepping into the MS-DOS market. They're disgusted by what they see as a "me-too" philosophy, and nostalgically think back to the Model I days when Tandy was a pioneer in the industry.

I have to disagree. The Models 1000, 1200, and 2000 do not indicate a sell-out, but a recognition by Tandy that it cannot survive, let alone thrive, outside the mainstream.

It's easy to get sentimental about the Model I. But this sentimentality has clouded some people's perceptions of Tandy's role in the microcomputer market. They think that because Tandy was one of the first, it therefore should be at the forefront of new and creative technology.

History says otherwise. Through the years, Tandy's success has depended on effectively packaging proven technologies. It has shown a remarkable propensity for cashing in on whatever electronic consumer product happens to be hottest. It did so with stereos. It did so with CBs. And most recently, it is doing so with telephones.

In all cases, Tandy's strategy was to make an inexpensive product, sell it by the truckload through its Radio Shack retail chain, and provide convenient after-sale support. The company's method for selling computers is essentially the same.

You could argue that at some point in the distant past, Tandy could have aggressively established its Z80 computers and TRSDOS operating system as de facto industry standards, as Apple and IBM eventually did with their products. But that would have been asking the company to be something it was not.

Staying with its proprietary DOS would spell the end of Tandy as a microcomputer manufacturer. It would be like selling stereo systems that don't play 12-inch records. So let's not

mourn the decline of the Z80 computer, but instead be thankful that Tandy had the good sense to join the rest of the world.

1985—The Comeback Year?

1985 could very well be the most important year in Tandy's history, for it will reveal whether the company can reestablish itself as a vigorous participant in the microcomputer marketplace. The three key factors are how well the Model 1000 does, how well Model 4 sales hold up, and how well Tandy can impress itself on the consciousness of the buying public.

My gut feeling is that the 1000 is going to do well. This is the type of machine Tandy does best: inexpensive and expandable. The company should sell a good many just to its current user base. Model I/III and Color Computer owners in particular should see the Model 1000 as their chance to enter the MS-DOS market at a reasonable price. If Tandy can also reach businesses that want a cheap alternative to an IBM PC or want to add to their current installment of PCs, the 1000 could become the company's all-time best seller.

How quickly the 1000 market develops is another matter. This is where the Model 4 comes in. Tandy must continue to sell the system during the transition to MS-DOS. To do so, it must sell the machine creatively, by further reducing the price, bundling software, and adding enhancements. (For instance, the company might extend the 4's life if it could get Zilog's 16-bit Z800 in quantity.)

Lastly is marketing, which in Tandy's case means getting more visibility. The issue of Tandy's image has been beaten to death with a tire iron, and rightfully so. The Model 1000 is Tandy's chance to put all that criticism to sleep.

Tandy is finally realizing that selling TRS-80s is not the same as selling stuffed-animal radios and batteries. The computer marketplace is highly competitive, and exposure is critical. Unfortunately, Tandy's machines do not get the shelf space, and therefore the visibility, in retail stores that other systems do.

The beauty of the 1000 is that it can create its own visibility. While the machine won't be in non-Radio Shack retail stores, its software will. Furthermore, computer magazines will have to reckon with the 1000 when discussing MS-DOS machines and IBM clones. All Tandy needs to do is prime the pump with an intelligent advertising and promotion campaign.

Marketing the Model 4, on the other hand, is going to present problems. It has become *machina non grata* in the computer industry. Take, for instance, a recent cover article in *Personal Computing* comparing the Apple IIc and the IBM PCjr. The 4 is lumped with the Kaypro in a couple of throwaway lines, and dismissed as having "inferior graphics/color capabilities, very little entertainment or educational software and very little expandability." Considering the 4's price, power, and software base, it deserves more respect.

All in all, Tandy has reason to be guardedly optimistic. If it plays its card right, this could be the year the company makes its comeback. ■

These were printed by DOTWRITER on an Epson MX-80.

See What You Can Do With DOTWRITER 4.0!

Now available for the Model 4, too!

DOTWRITER lets you create spectacular, eye-catching letterheads, catalogs, invitations, or even books. It is just what you need to turn your dot-matrix printer into a versatile typesetting machine. And it's **available for the Model 4 (yes, in native mode)** as well as for the Models I and III.

What is DOTWRITER?

DOTWRITER uses the "bit-image graphics" of your printer to produce the kinds of stunning results shown inside the box. It is a full-function text printing program, so you can inter-mix different character sets, do centering, paragraphs, pagination, magnification, draw horizontal and vertical lines, reversals (white on black), and even print **right-justified proportional text.**

DOTWRITER includes the printing program, complete documentation, and fourteen useful sets of typefaces (60 to 90 characters in each set). We will also give you the 105-page Letterset Reference Catalog **free** with your order.

To use DOTWRITER, just write your text with any popular TRS-80 Word Processor, add the necessary formatting commands, and DOTWRITER will do the rest.

25 more disks are available separately. Each has 3–12 complete typefaces (60–95 characters in each set). These disks cost less than $25 each, and may be purchased at any time.

If you want to create your own logos, modify our typefaces, or even design entirely new typefaces, then you will also want to order the "Letterset Design System" (LDS). We offer LDS at half-price when you order it along with DOTWRITER. LDS operates in Model III mode on the Model 4.

Versions are available for Epson MX-80 with Graftrax, MX-100 with Graftrax-Plus, RX-80, FX-80, C. ITOH 8510/1550, Microline 84/92/93; Radio Shack's DMP series 200–2100, CGP-220 & Gemini 10X, 15X. Please specify printer and computer!

Our print samples were done on an Epson. Sizes vary on other printers. Some of the samples shown here are taken from the additional Letterset disks.

Two disk drives and at least 48K of memory are required. LDS is not available in native Model 4 mode.

Send for free print samples! We've only shown you a few of the 180 DOTWRITER fonts. If you want the best in graphics printing, we suggest you order DOTWRITER today, toll-free.

Please specify printer and computer when ordering.

DOTWRITER (Models I, III)	$79.95
DOTWRITER (Model 4)	99.95
Letterset Design System	39.95
Special: DOTWRITER and LDS	99.95
Additional Letterset disks (4–12 per disk)	17.95 and 24.95
Letterset Reference Catalog	10.00

ORDER NOW, TOLL-FREE
(800) 824-7888, oper. 422

PROSOFT®

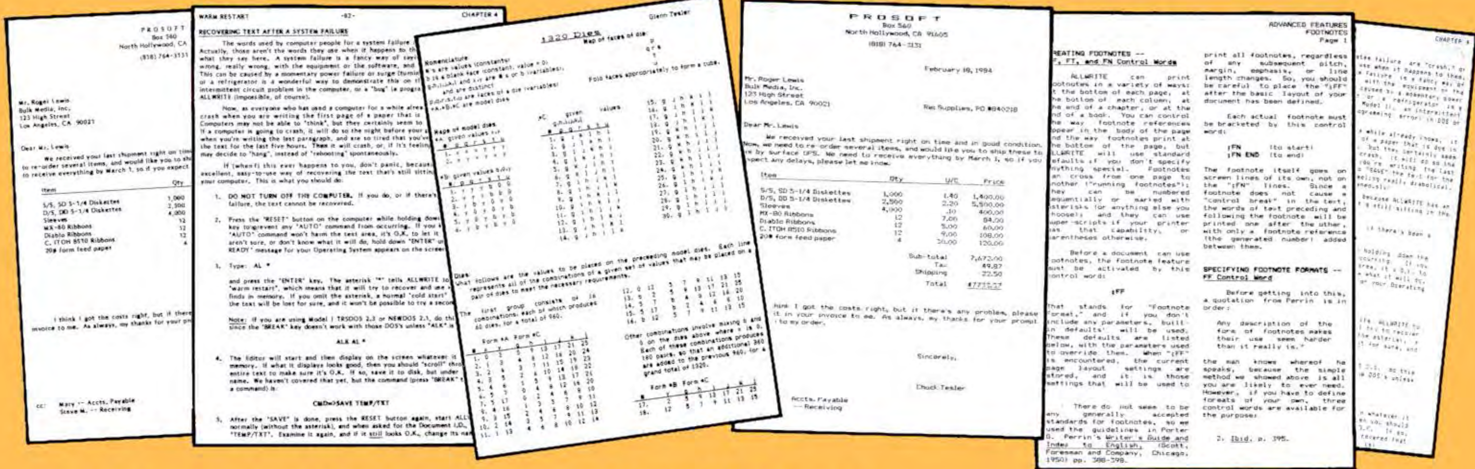
~91

Dept. C, Box 560, No. Hollywood, CA 91603
(818) 764-3131 Information and Same-Day Processing

TERMS: VISA, MC, checks, COD. Please add $3.00 shipping in U.S. or Canada. $15.00 overseas, sales tax in Ca. Most orders filled within one day

Allwrite!™

The Premier Word Processor for Your TRS-80 Model I, III, or 4



These were printed by ALLWRITE; shown 20% actual size.

We are proud to offer you the one Word Processor that will satisfy all your writing needs:* ALLWRITE. *It sets new standards for text editing and printing, and will give new life to your TRS-80. Let us tell you why...

In an attempt to push the public into expensive 16-bit computers, many manufacturers have been saying that the TRS-80 is obsolete. The truth is that the software, not the hardware, makes the difference. And the best word processor of all is now available *only* on the humble TRS-80, not on those expensive 16-bit machines!

ALLWRITE is based on the proven methods that made NEWSCRIPT the most popular independently produced TRS-80 word processor, but it also has the speed and new features our customers have asked us for. ALLWRITE will save you time and let you produce the highest-quality, most professional-looking letters, term papers, and reports available on a micro-computer.

Allwrite Can Save You Time!

Reads a 25,000 character file (10 printed pages) from disk in SIX SECONDS...does a global search-and-replace in FOUR SECONDS...outruns even the fastest popular micro-printer.

ALLWRITE'S Screen Handling Makes Word Processing Easier Than Ever

Change text width at any time; wide lines shift left and right as you type. ALLWRITE preserves double-blanks between sentences, uses the entire screen for text, and displays a complete Status Screen at the touch of a key. Scroll by line, partial screen, full screen, to top or end of file, or to any marked point. Move cursor by character, word, tab, line, or screen.

You can set and change on-screen tabs and store them on disk. The print-time tabbing features are incredibly versatile: they allow left, right, and centered tabs, and even line up your decimal points.

ALLWRITE shows you where you forgot to turn off underlining, boldface, italics, or double-width. Special on-screen Preview feature shows page breaks and page layouts...including underlining and boldface ...without annoying blinking or screen flicker. In "Summary" mode, ALLWRITE quickly flags formatting errors without wasting time printing all the text. These standard features make document preparation faster and easier than ever!

State-Of-The-Art File Handling

There is no upper-limit on document size with ALLWRITE, because it chains files *backwards* as well as forwards, even across diskettes. Switch from one chained file to another in less than six seconds by pressing two keys. Select portions of other files for inclusion at print time...great for stock paragraphs.

ALLWRITE salvages text from bad disks! If a sector goes bad, you won't lose the entire file, because it will skip bad sectors, **read the rest of the file,**

TAKES FULL ADVANTAGE OF YOUR MODEL 4.

The model 4 version of ALLWRITE uses **the entire 80-by-24 screen.** On a 64K machine, you can edit **over 34,000 characters** of text. On a 128K machine, you can edit **THREE FILES AT THE SAME TIME!** The second and third files can be over 32,600 characters each, for a total of **almost 100,000 characters** of text in memory.

and then show you where the lost text belongs. This advanced error recovery turns a disaster into a feeling of profound relief.

User-Definable Soft Keys Reduce Typing Time

You can store **22 phrases or commands at a time into "soft-keys,"** then press just two keys to retrieve them. This makes frequently-used phrases and formatting controls *a snap to use*. You can store these definitions on disk and build a library of hundreds of preprogrammed keys to fit every one of your applications.

Our specially-designed templates fit right on your keyboard to let you see your settings at all times. Each template is also a Reference ("Cue") Card, so it is always right in front of you when you need it, without using up valuable screen space.

ALLWRITE Is Easy To Learn

ALLWRITE's commands and control keys are easy to remember because they use the first letters of common English words: 'CE' stands for 'Center,' 'Search' and 'Replace' do just that, and so forth. The **on-line HELP menu** offers over fifty screens of topics.

NEWSCRIPT's documentation was acclaimed in every review, and ALLWRITE's 350-page book is even better. Portions of it are designed for beginners, with every feature clearly explained in step-by-step tutorial style. Since you won't always be a beginner, other parts of the book offer advanced topics. There is a cross-reference summary chapter, a 14-page comprehensive index, and a detailed Table of Contents. We've been developing computer programs and manuals for over 20 years, and understand the importance of good documentation.

To make installation easy, we include Tiny DOSPLUS for the Models I and III, and special, pre-tailored versions of **both** TRSDOS 6.2 **and** DOSPLUS IV for the Model 4, all at no extra charge. The Model I and III versions work equally well with all major DOS's.

PROSOFT'S On-Going Customer Support

Perhaps the best reason of all for having ALLWRITE is the continuing support we offer you: friendly, expert, direct support that is unsurpassed in the micro-computer industry. There is no time limit to our support: if you are our customer and you need help, just call or write. We give free updates for 90 days, and charge little or nothing for minor updates thereafter.

Customer Comments

"This is the best software package I have ever received... superb, easy to use, fast, and has more features than the business word-processor at the office." **(E.R.L.)**

"ALLWRITE is a professional system that sets a new standard in word processing. It's powerful and easy to learn and use."
80 MICRO, Nov., 1984

"Your company and products have to be one of the strongest factors I can think of for keeping me with the TRS-80!" **(J.R.H.)**

"NEWSCRIPT is the Cadillac of word processors. ALLWRITE is the Mercedes Benz!!" **(B.E.)**

"...a very readable manual." **(D.S.)**

BENEFITS OF OWNING ★ ★ ALLWRITE ★ ★

If Word Processing is important to you, PROSOFT's ALLWRITE is the best choice you can make. The clean, professional appearance it adds to your letters and reports will make an excellent impression on people. We will be happy to send you free print samples so that you can see for yourself how good ALLWRITE will make you look.

You probably know that quality word processors for CP/M and the IBM-PC sell for $300–500, and they don't have ALLWRITE's capabilities or speed ...or PROSOFT's proven, on-going support. Now, for a fraction of the cost of a new computer, you can have the most complete word processor of all. And you won't have the headaches of starting all over again with a new, different computer.

Note to college students: with its Footnote, Table of Contents and Index features, ALLWRITE is ideal for your reports and Term papers.

HUNDREDS OF USEFUL CAPABILITIES

ALLWRITE comes with just about every useful word processing feature ... standard. Here are some highlights: excellent right-justified proportional printing on most printers having that ability; powerful Form Letter and Mailing Label preparation; Instant counts of words, characters, lines, changes; block Move, Copy, Delete, Putfile, Getfile, and List; delete by character, word, line, sentence, paragraph, or block; insert and one-key insert; great RS-232 printer support; accepts all 256 ASCII codes from keyboard; intermix pitches on same line (printer-dependent); 1.5 line spacing, 6, 7, 8, 12 lines per inch (printer-dependent); does multiple-columns on all printers; perfect alignment of hanging indents; variables, logic statements, conditional printing; wildcard Directories; integrated with Electric Webster and DOTWRITER for Models I, III, and 4 (these are sold separately); "Legal" line numbering; paragraph, list, and figure numbering; supports most popular printers (all "printer drivers" included); compatible with high-memory drivers; fully explains all DOS and ALLWRITE error messages; wildcard search-replace; tabs, search-replace, other settings remembered across files; word reversal; up to nine levels of boldface; flexible page titles; footnotes at bottom of page or end of document; Table of Contents and Index generation; and PROSOFT's unmatched text formatting and printing capabilities.

Attention NEWSCRIPT Owners

You can trade in your copy of NEWSCRIPT at very substantial savings. Please call our technical line for details.

How To Order

You can order by phone or mail. For quickest delivery, call our Technical Support line. Please specify your TRS-80 model (I, III, or 4, 48K, at least two disk drives), and your printer(s). Our price includes normal shipping in the U.S. and Canada. The sooner you order, the sooner you will begin to benefit from the ALLWRITE! Word Processor.

~~$249.95~~ **$199.95**

ORDER NOW, TOLL-FREE

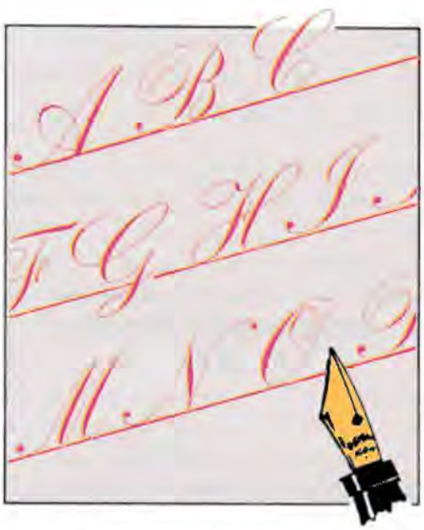
(800) 824-7888, oper. 422

PROSOFT®

Box 560, No. Hollywood, CA 91603

(818) 764-3131 for Technical Information and Same-Day Processing.

Terms: VISA, MasterCard, checks, C.O.D. California residents please add sales tax.
Most orders filled within one day.
Model 4 version available at Radio Shack: Cat. No. 90–0242 ✓30



NICE Perspective

I found R. Walter Steur's review of NICE (October 1984, p. 39) shallow and overly negative. Steur ignores the fact that NICE is one of the most flexible and versatile TRS-80 packages available.

NICE isn't perfect; its documentation, for example, leaves much to be desired. Yet, when considering its performance and capabilities, I think NICE deserves a four-star rating.

Michael R. Keller
Skowhegan, ME

Where's The Disk?

Contrary to what's stated in Table 1 of Hardin Brothers' "Making Your Selection: Choosing the Right Editor/Assembler" (September 1984, p. 60), Radio Shack's Series I Editor/Assembler includes only the Model III disk in its package. In place of the Model I disk that I wanted were instructions for exchanging the enclosed disk for a Model I disk. If it's the Model I disk that you're after, I advise that you open the package before buying the product.

G. F. Mueden
New York, NY

Communicating

I disagree with what M. J. Batham refers to as "drawbacks" in her review of Videotex Plus (September 1984, p. 168). First, hex codes are required only if you use the SETCOM feature on an automatic log-on. If you use the clear-Q keys to configure the program, you're presented with a chart of baud rates or UART configurations. Then, all you have to do is select the number conforming to the baud rate of the desired configuration.

Secondly, Batham states that you can't go back into DOS without logging off or redialing the host computer. I do this all the time; just exit using clear-X, and then reload Videotex Plus. You will, of course, lose information sent by the host computer while you're out of Videotex Plus.

Finally, control-P doesn't send the signal that prints the contents of the buffer; rather, it sends a control-P to the host computer. It's clear-P that prints what's contained in the buffer.

L. J. Kutten
St. Louis, MO

The inaccuracies in M. J. Batham's review of Videotex Plus can't be ignored. The program is capable of generating a true break signal; p. 23 of the User's Guide and the on-screen help menu both state that the F3 function key generates this.

You turn on the printer with clear- or control-R, while clear- and control-P is sent to the RS-232 board. In addition, Videotex Plus doesn't support XMODEM protocol, but instead supports CompuServe's B protocol.

Adam Rubin
Wappingers Falls, NY

In reviewing Videotex Plus, I tested the program extensively using the Model 4 to communicate with CompuServe and a local IBM-type BBS. Hex codes are required to use Videotex Plus's auto log-on feature. Pages 8–11 of the manual explain how to input hex codes, yet it's confusing to a user who doesn't know hex.

The only way to return to DOS is to exit the program which, in turn, makes you lose information from the host computer. Other telecommunications programs allow a return to DOS while capturing data in the buffer.

Control-P on my version (1.00.00) toggled the printer, allowing it to print the contents of the buffer. Page 25 of the manual explains the clear-P option to print the buffer contents. The use of both P and R are similar, with R acting as a print screen function, and P as a print buffer option.

Adam Rubin is correct in stating that the F3 key generates a break, yet the break key doesn't initiate a true break. When following the instructions on p. 31 of the manual, I had no difficulty with file transfer using the XMODEM protocol. In doing this, I also used the No option to turn off the XON/XOFF support.

M.J. Batham
Des Plaines, IL

80 ALERT

Occasionally, 80 Micro *receives letters from advertisers who have changed their status, or from readers who have had difficulties with our advertisers. Most of these problems are resolved to the satisfaction of all parties, but some problems appear to be insoluble.*

As a service to our readers and advertisers, 80 Alert posts the names of advertisers we are unable to reach, or who have changed their address or status. Anyone who has current information about a manufacturer or distributor, or who has an advertiser complaint, should write to 80 Alert, c/o 80 Micro, *80 Pine St., Peterborough, NH 03458.*

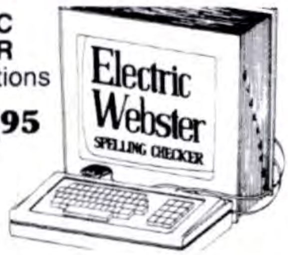
80 Micro has been informed that Micro Management Systems Inc., 2803 Thomasville Road E., Cairo, GA 31728, has stopped taking orders until further notice.

PRICES SLASHED!

Our Same Super Service at New Lower Prices

ELECTRIC WEBSTER
W/Corrections

105^{95}



HYPHENATION OR GRAMMAR **37^{95}**

LeScript **97^{95}**

NewScript™

77^{95}

Lazy Writer **93^{50}**

DOT WRITER
W/Letterset Design System

71^{95}

SOFTWARE

NEWDOS80/V.2	$ 97.50
COMREF for Newdos80	13.95
MICRO CLINIC Mod 1	24.50
MICRO CLINIC Mod 3	28.50
ARRANGER II	44.95
TRASHMAN	32.50
FASTER	22.95
R.P.M.	21.50
DSMBLR II	33.95
PRO-DUCE	33.95
SUPER UTILITY +3.2	62.95
ZORLOF II	49.95
MSCRIPT	55.95
EDAS IV	84.00
PRO-CREATE	84.00
OMNITERM	64.95
CYBERCHESS	24.95
TALLYMASTER	54.75
MAXI CRAS	74.95
MAXI MANAGER II	146.95
ENBase	132.50
SBSG Accounting Modules	
Model 1 or 3	165.00
Model II	240.00
MAS80 Accounting Syst.	299.95
EDIT-Basic Editor	33.50
PASSWORD 300 Modem	184.95
PASSWORD 1200 Modem	339.95
VOLKSMODEM w/cable	81.90

VERBATIM DATALIFE DISKS

5¼″ SS/DD	22.95/10
5¼″ DS/DD	29.95/10

WRITE FOR OUR FREE CATALOG

DOSPLUS 3.5

$\$50^{95}$

DOSPLUS 4A
with M-ZAL

115^{95}

6.X PLUS
Improve Your TRSDOS 6.2

37^{95}

MTERM
Special Price thru 1/31/85

$\$49^{95}$

MOD4 Speedup Kit
By Alpha Technology, Inc.

75^{00}

TEAC DRIVES

FD55A 40 Track	145.95
FD55B 40 Track Double	179.95
FD55F 80 Track Double	199.95

The above Teac Drives are half height bare drives and include a full year limited warranty.

- Free Use of Credit Cards
- Friendly, Honest, Reliable Service
- Toll Free Order Line
- 24-Hour Shipping for Items in Stock
- Large Selection-Call for Items Not Listed

We accept Visa, MasterCard, check, cash money orders, COD, and POs on approval. In the 48 continental States, please add $2.50 for UPS standard shipping. Alaska, Hawaii, FPO, U.S. Mail, and foreign orders are charged actual shipping & handling charges. COD orders are charged an extra $2.00 and require cash or certified check upon delivery.

When ordering by mail include your telephone number, credit card information, computer model, memory size, and number of drives. Colorado residents add appropriate sales tax. Prices are subject to change without notice.

TOLL-FREE ORDER LINE
1-800-468-4474

COLORADO/TECHNICAL
(303) 861-9250 ~390

AMI applied microsystems, inc.
612 Washington, Denver, CO 80203

Send any questions or problems dealing with any area of TRS-80 microcomputing to Feedback Loop, 80 Micro, *80 Pine St., Peterborough, NH 03458.*

Q: I've been helping a friend upgrade his Model I from 16K to 48K. He's had problems with the Model I's random-access memory and I was thinking of speeding up the RAM-refresh signals. Dennis Kitsz, in *The Custom TRS-80,* says "locate Z69, and cut the trace running from pin 5 to pin 12. Connect pin 12 to pin 13. This speeds the memory-select process (from MREQ and RD) just a tad."

However, an old *80 Micro* Exclusive Oracle column says "Find Z89, locate the circuit trace running from pin 5 (it runs back underneath Z69). Cut that trace. Jumper pins 10 and 12 of Z69 with a short piece of wire." I figure one of them must be right. I'd appreciate it if you could tell me which one.

Also, what is Dennis' full address? I have no street address, and my post office says it can't deliver a letter without one. *(Steven Maguire, Port Richey, FL)*

A: Not to worry; the two methods are electronically equivalent. The first is slightly more reliable. Both generate the memory-select signal a cycle ahead of design, giving the memory more time to react and deliver data.

Your post office people are naive. Dennis Kitsz' complete address *is* Roxbury, VT 05669. When the total population of a town is about 100, street addresses are superfluous.

Q: An LNW ad on the back of *80 Micro* says that the Team computer is compatible with Radio Shack's Models I, III, and 4. How compatible is compatible? I've had quite a few problems with my Model III due to lack of cooling and/or ventilation. Also, I'm at the point where I need the added capabilities of a machine like the LNW. So overall, how would you rate the Team?

I use Cobol and SuperScripsit with TRSDOS 1.3 and I want to move them to DOSPLUS 3.5 (or 4.0 if I get the LNW). How can I do this without knowing the passwords? I have patches from MicroSystems to make them work under DOSPLUS, but how do I move the programs?

Finally, when I write (in Cobol) to a DMP-2100 printer, the underline feature turns on and I have to turn off the printer to reset it. Can you or anyone else help me with this one? *(Rick C. Chandler, Brentwood, TN)*

A: Get a cooling fan for your Model III. Radio Shack sells muffin fans that'll do the job. If you want something a bit more attractive, look for a 4-inch box fan at a local electronics store. It shouldn't cost more than about $20.

Model I/III/4 software seems to be better than 90 percent compatible with the LNW. I can't give an opinion on the computer as I haven't seen it.

As for the passwords for SuperScripsit and Cobol, MicroSystems should be able to give them to you, or tell you how to get the job done. I believe their Transfer utility will move the program by ignoring the passwords and protection levels, but the programs will still have these attributes on the new disks. Check your manual for more information.

I don't know what to suggest for your printer problem except doing a dummy write, turning off the printer, resetting the underline, and continuing with the job. Try looking up the underline codes in the DMP-2100 manual and sending the turnoff code in a Write command.

Q: I have a 48K, dual-drive Model III on which I use Scripsit to write novels. I desperately need more RAM. Is there a way to increase the RAM in my machine? Do you know of a patch that adds underlining to Scripsit?

Second, are you familiar with Holmes Engineering's Sprinter Three? Does it work? What, if any, problems are associated with it? *(Howard Davis, Vail, CO)*

A: Unfortunately, you can increase the amount of available RAM, but Scripsit won't recognize it. If the buffer space is inadequate for the file size you're using, I suggest that you get a new word processor such as Newscript, which lets you chain individual buffers into one long printed manuscript. Acorn Software's Superscript program patches Scripsit to add features like underlining and boldface. You can order the package from Acorn at 634 N. Carolina Ave. S.E., Washington, DC 20003, 202-544-4259; $50.

Sprinter Three speeds up CPU operation. Cassette and disk I/O automatically slow down to normal CPU speed to maintain compatibility with other Model IIIs. You can increase CPU clock speed by 50 percent; increases of 100 and 150 percent are possible but you might have to replace some of your computer's integrated

Newclock-80 $69.95

The right time at the right price! Keep the time and date with quartz accuracy, even when your computer is off. The backup lithium battery (included) will last for over 2 years. Software on tape or disk, please specify. Use "TIMESET" once to set the clock. Use "SETCLK" to set your computer's internal clock (at power up) or use "TSTRING" so that the "TIME$" function reads the Newclock.
Connection: Model I: plugs into the keyboard or expansion interface. Model III: plugs into the 50-pin I/O bus. Compatible with all operating systems.

Printswitch $59.00

Do you have 2 printers? Get a Printswitch. Stop plugging and unplugging those printer cables. With the Printswitch, you can have 2 printers connected to your computer and you can select either one at the flick of a switch. Works with any printer, plotter, or device that uses the parallel printer port. Simply plug the 14 inch Printswitch cable into your computer, and plug your existing printer cables into the Printswitch. This is the nicest unit on the market. Superior quality board with gold plated edge connectors. For Models I,III, 4 and 4P.

Alpha Joystick $27.95

When it's time for fun, don't be without your Alpha Joystick. Do you know that most action games are Joystick compatible? Stop pounding on your keyboard and enjoy real arcade control. The joystick can also be used with BASIC programs; simply do J=INP(0) to read the joystick position (8 directions and fire button). Model I: plugs into keyboard or expansion interface. Model III, 4 and 4P: plugs into 50-pin I/O bus. The Alpha Joystick comes fully assembled and tested, ready to plug in and enjoy. (Specify Model I, or Model III,4).

Interfacer-80 $159.00

Low cost input and output device. The outputs consist of **8 relays** (rated 2 Amp @125V), easily controlled using "OUT" commands. For example, OUT 0,0 turns all the relays off. Eight LED's show the states of the relays. The **8 inputs** are optically isolated, so it's safe and easy to connect external devices (switches, sensors, thermostats, etc.). Simple "INP" commands read the inputs.
Connection: Mod I: 40 pin bus. Mod III, 4, 4P: requires 50-pin I/O bus converter ($39.95), plugs into 50-pin I/O bus. Comes complete with power supply, cable, and detailed manual.
(Up to 8 interfacers can be connected to your TRS-80 using our Y- cables).

Analog-80 $139.00

8 channel 8 bit Analog to Digital converter. Your TRS-80 can read voltages, temperatures, pressures, light levels, etc. • Input range: 0 to 5.1 Volts. • Resolution: 20mV. • Conversion time: 120 microseconds. In BASIC, you can take up to 100 readings per second. • Port address: selectable. Up to 8 Analog-80's can be connected to your TRS-80 for a total of 64 channels!
Connection: Model I: 40 pin I/O bus. Model III, 4, 4P: requires 50-pin bus adapter ($39.95). Comes complete with power supply, cable, and manual.

Special Cables

Disk drive extender cable (8")...C160:$9.95

Y-Cable for Mod I bus (40 pin): • **X2-40**...$29 • **X3-40**...$44 • **X4**...$59 • **X5**..$74
Y-Cable for Mod 3 & 4 bus (50-pin): • **X2-50**...$34 • **X3-50**...$49 • **X4-50**...$64
Disk drive cable (34 pin): • 2-drive...**C162:**$32 • 4-drive...**C163:**$45
Extension cable, 4 foot: • For printer and drive (34-pin)...**C165:**$22
• For Mod I bus (40-pin)...**C167:**$24 • For Mod 3 & 4 bus (50-pin)...**C169:**$28
Keyboard to E/I (40-pin, 8")...**C161:**$21 If this is confusing, send for our Cable Flyer.
Our cables are made with high quality gold plated connectors to ensure utmost reliability.

X5 X4 X3 X2

NO RISK

We know that ordering by mail can sometimes be a hassle, but with Alpha, you are fully protected.
1) Our 15 day money back guarantee protects you from disapointment.
2) All our products are tested on a TRS-80 and waranteed to 90 days (parts & labor)
3) We ship 90% of our orders within 24 hours.
4) We are committed to help you

Green Screen $12.50

Do your eyes a favor, put on a green screen. Tens of thousands are in use for a good reason: they work. Contrast is enhanced and eye fatigue is greatly reduced. Our green screen is curved; it fits right on the face of the tube. (Fits Models I,II,III,4,12,16)

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421 ✓17 **(718) 296-5916**

Toll Free Order Line
800-221-0916
Orders Only. NY & Info call (718) 296-5916.
Hours: 9-5 Eastern Standard Time

Add $3.00 Per Order For Shipping And Handling. We Accept Visa, Mastercard, Checks, M.O. C.O.D. Add Sales Tax. Overseas, FPO APO: Add 10%. Dealer Discounts Available. COD add $3.00 extra.

circuits (memory and associated chips) with higher quality chips.

Q: I have some comments about Model 4 software. I recently purchased CP/M+ from Radio Shack, and my biggest complaint is that a number of files are missing.

My Model 4 has double-sided drives. The CP/M+ manual provides instructions for reconfiguring the Basic input/output system (BIOS) and Basic disk operating system (BDOS). The only problem is that the CP/M+ package lacks the necessary BIOS or BDOS files. When I requested help, Radio Shack told me that it wasn't going to support CP/M+ and that I'd have to go to Digital Research to get the necessary files.

I also requested help with installing Wordstar 3.0. CP/M+ is configured to be a DEC VT52 terminal, but the installation program doesn't include a configuration file for this terminal. I finally bypassed the problem by installing Wordstar as a Heath H19/H89 terminal, which is DEC VT52-compatible.

I configured my system to use the click filter, which produces a noise each time you press a key. The click filter conflicts with SuperScripsit. I asked the computer service rep at the local Radio Shack if there were any patches to get around the problem. After two weeks, I was told that the two programs conflict and that I'd have to turn off the filter when using SuperScripsit. I already knew this. *(Gerald Crosby, Columbus, OH)*

A: Thanks for writing. Does anyone know of a patch that takes care of the SuperScripsit problem?

Q: Regarding Christopher Jensen's question about using Enhanced VisiCalc on the Model I (April 1984, p. 14): Logical Systems Inc., maker of LDOS, has a patch that lets you run Enhanced VisiCalc on both the Models I and III. It works under LDOS or SmallDOS. The patch is public domain and comes on a disk of patches called the Fix Disk. The disk is $10; you can reach Logical Systems at P.O. Box 23956, Milwaukee, WI 53223, 414-355-5454. *(Robert M. Connors, APO NY)*

A: If it isn't too much of an expense, readers should get LDOS when they want to use Model III software on a Model I. They'll also have to buy a double-density modification (and most Model III software requires the additional drive space available with double density) but that's still cheaper than a new Model 4.

Q: I recently bought a secondhand Model III and took it home in my car. When I tried to call a directory for the drive 1 disk, I got an error message (data lost during disk I/O). I had Radio Shack align both drives, which cost me $60, but the problem returned after a week. Radio Shack fixed the drives again, at no charge.

The computer worked fine for two months. Then I took it to work in my car, and it did it again. I have a sneaky feeling that the ride in my car is shaking the computer and moving the drives out of alignment. Is this possible? Is there a special disk I can use to protect the drives when transporting the computer; if not, can I realign the drives myself? *(Dorothy Belk, Little Rock, AR)*

A: The problem is definitely vibration, but your drives shouldn't be that sensitive. I suspect that the alignment screw is a bit loose in its fittings. If you must move the computer around, get the Disk Drive Analyzer from J & M Systems (137 Utah NE, Albuquerque, NM 87108, 505-265-1501; $79 for single-sided drives) so you can test your drives. This program and disk are the next best thing to having a professional technician with an oscilloscope look at your drives.

Q: I've been having problems saving longer programs on my Radio Shack Series I Editor/Assembler. Could it be the computer or tape recorder? Is there a limit to the size of a file the editor/assembler can save? The manual doesn't mention any such limitations. *(Rodney Maglente, Chicago, IL)*

A: Series I is limited to a maximum file size of 211 lines (note that the limit is lines, not memory). If you're having problems with files smaller than that, I suspect that the tape recorder head is drifting as you're writing or loading a file, but that should give you some type of error message when the file fails to load properly. If you aren't getting the error message, I'm stumped. Does anyone else have ideas or comments?

Q: Here's an additional suggestion for Don Schenbarger (March 1984, p. 15) regarding the "Diskette?" prompt. If the cable from the disk drives' controller board to the CPU board doesn't make good contact, the prompt can appear, so the problem could be hardware related.

I'd like to know the relationship between Model I Scripsit and my version, 3.2. Craig Lindley's patch program for Model I Scripsit (October 1982, p. 276) doesn't work with my version. Does Scripsit 3.2 occupy different memory locations? Is there a disassembled listing of version 3.2 available so that I can install some of the patch functions?

Also, would you recommend a good disassembler program for the Z80A as used in the Models III and 4? I'm interested in one that converts machine hexadecimal code into symbolic or mnemonic code. *(Stephen A. Torkko, North Bend, OR)*

A: Patches for programs on the Model I invariably don't work on Model III versions of the same program. The problem is that disk and printer I/O routines are almost totally different. If you want patches that make Scripsit much easier to use, try Acorn's Superscript. For more information, see my answer to Howard Davis' letter above.

I don't know of any disassemblies of Scripsit 3.2 you could buy. Can anyone else help? I'm not familiar enough with the disassemblers currently on the market to make a suggestion.

Q: I have a late-issue Model I with an RS Expansion Interface, for which I bought double-sided drives (Tandon TM-100-2). To date, I've been unable to access the second side. My DOSPLUS 3.5 and LNW 5/8 doubler perform flawlessly as long as I've configured only one side. I run into problems when I try to generate a double-sided master of DOSPLUS. I've tried several copies of DOSPLUS and even had the SYSGEN utility replaced, suspecting it was at fault. I feel that I have some kind of hardware problem, and that the computer isn't physically accessing the second side. Any suggestions? *(Chuck Fullgraf, Aiken, SC)*

A: To the best of my knowledge, DOSPLUS 3.5 works with double-sided drives. Before you try SYSGENing the system disk, use the CONFIG command to access the second drive as a double-sided drive. Once DOSPLUS knows that the second disk has two sides, you should be able to verify and use both sides of any disk in that drive. Don't forget that you have to use double-sided disks; single-sided disks won't format as double-sided unless you've altered them properly.

As soon as you finish with the CONFIG command, format a double-sided disk. If the second side fails to verify, check the configuration of your system and make sure that the drisk in drive 1 reports as being double-sided. If it is, and you still can't format both sides of a disk, you definitely have a hardware problem.

Call LNW to make sure your doubler supports double-sided drives. Then check with MicroSystems Software and see if you're doing anything wrong with the CONFIG command, and ask for advice. Finally, take your drives to a technician and have them checked out; they might have a jumper that needs to be set, or some other such silliness.

Once you get DOSPLUS working on both sides of your second drive, you can then SYSGEN a master DOSPLUS disk for use in drive zero.

Q: I bought a Model 4 from Displayed Video, but I didn't receive a TRSDOS 6.0 registration card. How do I get one? Can I expect TRS-80 equipment from dealers like Displayed Video to be equal in quality to all-Radio Shack equipment?

Also, is there a way to solve the problem of losing characters when words wrap around in SuperScripsit? Do any software companies offer modifications to SuperScripsit? The features I have in mind are indexing, footnoting, and so on. Since software firms seem to be advertising less in magazines like *80 Micro*, where can I learn about new programs for the Model 4?

Finally, why can't I directly transfer files between a Model 4 and a Model 100 via the cassette interfaces? Is it absolutely necessary to have an RS-232C interface? *(Hugh Ruppersburg, Athens, GA)*

A: Only Tandy gives out registration cards, but you're in luck. Tandy just released TRSDOS 6.2. If you go to a local Radio Shack store and ask for the updated DOS, you should get a new registration card with it. And they shouldn't charge you for the update.

Buying from third-party companies that provide their own memory and drive upgrade is a *caveat emptor* situation. Usually you get equipment equal in performance to standard Tandy devices, but that's not a guarantee. Sometimes you'll get equipment superior to Tandy's. Rarely will you get inferior equipment, but it's possible. In 99 percent of the cases, as long as you stay within the normal operating specifications of a standard Model 4 (that is, don't try to add speed-up kits, super-high-density drives, etc.) you won't be able to tell the difference between standard and non-standard Model 4's.

I haven't heard of any solutions for SuperScripsit's losing characters during wraparound, and I don't know of any companies selling patches or enhancements for the Model 4 version. To find out about new software, check the New Products section of this magazine. If you're looking for a word processor with automatic indexing and table-of-contents capability, give Prosoft's Allwrite serious consideration (for a review, see p. 35 of the November 1984 *80 Micro*).

While the Models 4 and 100 use the same techniques and baud rates to save data to tape, there's one big difference: The Model 100 saves a title before the data, and the Model 4 doesn't. The Model 100 rejects Model 4 tapes because they aren't identified by a title, and the Model 4 rejects Model 100 tapes because the computer requires a line number before the title. You need a machine-language routine that creates a title when saving programs from the Model 4, or one that ignores the disparities in the files when you're loading from one machine to the other.

Q: I've had the same problem as Robert Goodman (April 1984, p. 16) with the automatic 60-line count in Basic. If he begins each print series with a PRINT CHR$(28), the auto line count will reset to zero. *(Galen F. Tustison, San Marcos, CA)*

A: In other words, before starting a print job, line up the paper to the top of the page, send a CHR$(28) to the printer, then start printing. Now the six-line skip will occur at the page perforations, making for a clean printout. Thanks for the advice.

Q: Does anyone know where I could beg, borrow, or steal a Micro Design MDX-2 expansion interface board? Micro Design has discontinued it. I'll accept a slightly used or damaged board, but I'd prefer a board in its case, with the complete set of manuals. *(Steve Winokur, Horsham, PA)*

A: All right, folks: Does anyone out there know where Steve can get an MDX-2 board?

Q: I'm looking for Forth for my Tandy 2000. Laboratory Microsystems Inc. of Marina del Rey, CA, has a package, but it's based on the 8086 microprocessor. I've read that the 2000's 80186 has the 8086 instruction set in it, and therefore Forth should execute. Is this true? Also, does the 2000 have a math coprocessor included, or can I use the 8087-3? Better yet, do you know of a version of Forth written specifically for the Tandy 2000? Is Microsoft planning to publish one for its MS-DOS system? *(Ron Watkins, Tucson, AZ)*

A: Yes, 8086 Forth will run on the Tandy 2000, but you must make sure you have a version designed for that computer. Just as you can't run Z80 SuperScripsit on every Z80 computer, you can't run 8086 Forth on every 80186 computer. The program must be customized to use the computer's keyboard, video, and disk drives.

Before investing in a commercial program, try contacting Forth Inc., 2309 Pacific Coast Highway, Hermosa Beach, CA 90254. They own the Forth trademark and are responsible for distributing FIG-Forth, the public-domain version available for almost all computers. ■

Terry Kepner is a writer and programmer, and an Associate Editor of 80 Micro. *He's been writing about microcomputers since 1979.*

You've Got TOTAL ACCESS

(specializing in TRS80 *)

TO YOUR COMPUTER HARDWARE & SOFTWARE NEEDS. CALL ROSE TODAY!

QUALITY DISK DRIVES

These drives are complete with power supply, cover and external drive connector. For TRS-80 Model I, III, 4, IBM PC and others. All drives are Double Density and step at 6ms or less. SS means single head, DS is double head. Specify white or silver color cover. Add $5 per drive shipping unless otherwise specified. All drives have a one year warranty on parts and labor.

COMPLETE FULL SIZE 5.25" DISK DRIVES

40tk Single Side Tandon TM100-1	$ 169
40tk Dual Side Tandon TM100-2	209
80tk Dual Side Tandon TM101-4	279

COMPLETE HALF-HIGH 5.25" DISK DRIVES

1ea 40tk SS TEAC FD-55A in dual case	$ 189
2ea 40tk SS TEAC FD-55A in dual case	324
1ea 40tk DS TEAC FD-55B in dual case	243
2ea 40tk DS TEAC FD-55B in dual case	424
1ea 80tk DS TEAC FD-55F in dual case	297
2ea 80tk DS TEAC FD-55F in dual case	522

COMPLETE 8" DISK DRIVES

Add $10 shipping per 2-drive case

2ea SS TM848-1E's in dual case with fan	$ 699
2ea DS TM848-2E's in dual case with fan	799

BARE 5.25" & 8" DISK DRIVES

Add $4 shipping per drive

40tk SS, Full Size, Tandon TM100-1	$ 129
40tk DS, Full Size, Tandon TM100-2	169
80tk DS, Full Size, Tandon TM101-4	239
40tk SS, Half-High, TEAC FD55-A	129
40tk DS, Half-High, TEAC FD55-B	169
80tk DS, Half-High, TEAC FD55-F	239
8" SS, Thinline, Tandon TM848-1E	290
8" DS, Thinline, Tandon TM848-2E	360

TRS-80 MODEL III/4 DISK DRIVE KITS

Add $8 shipping per kit

Internal drive kit complete with disk controller, power supply, mounting brackets, cables and all hardware plus step-by-step instructions. This kit contains everything you need (except the Disk Operating System, drive and a screwdriver) to convert your cassette Model III or 4 to fast reliable disk operation. Don't confuse this quality kit with the high-priced ripoffs. Thousands of satisfied customers cannot be wrong. You can join them for only .... $ 199

OTHER DRIVE GOODIES

Add $2 shipping

TRSDOS 1.3 Disk & Manual for Model III	$ 23
TRSDOS 2.3 Disk & Manual for Model I	23
TRSDOS 6.x Disk & Manual for Model 4	29
LDOS for the Model I or III	69
NEWDOS 80 v2.0 for the Model I or III	93
2-drive cable for Model I/III/4	24
2-drive cable for IBM PC	40
4-drive cable for Model I	34
Extender cable, 7" long	13
5.25" power supply & encl., white or silver	59
8" power supply, fan & enclosure, beige	159

TRS-80 MODEL I DOUBLE DENSITY CONTROLLERS

Add $3 shipping

Aerocomp "DDC" Really the best by test	$ 99
Aerocomp "DDC" with LDOS	149
Aerocomp "DDC" with NEWDOS 80-v2.0	189
LNDoubler 5"/8" if you need 8" drives	169

MEDIA & SUPPLIES

5" Diskettes SSDD, Lifetime Guarantee. Pk of 10	$16
5" Diskettes DSDD, Lifetime Guarantee. Pk of 10	21
5" Flipsort, holds 75 Diskettes	19
8" Diskettes SSDD, Lifetime Guarantee. Pk of 10	29
8" Diskettes DSDD, Lifetime Guarantee. Pk of 10	34
8" Flipsort, holds 50 Diskettes	29
5.25" or 8" Head Cleaning Kit	9
Letter Size 20 lb. Tractor Paper, 2900 sheets	25

TRS-80 SPECIAL EQUIPMENT

12" Green Phosphor Monitor	79
12" Amber Phosphor Monitor	84
16K 200 nsec RAM Guaranteed 1 year (8 chips)	13
64K 200 nsec RAM Guaranteed 1 year (8 chips)	64
64K RAM plus Genuine PAL for Model 4 upgrade	74

ROSE GETS RIGHT! NOW---ROSE'S MOD 4 CP/M

$ 69 Complete with Manual

Rose has developed this streamlined version of CP/M 2.2 that lets you run most of your favorite CP/M programs. They are in stock, ready to ship to you at this unprecedented low price.

SPECIAL SOFTWARE DEALS

Add $4 shipping

WordStar 3.3 (Specify MM or R/S format)	$ 250
MailMerg, SpellStar, Starindex	220
WordStar Professional (Above 4 Progams)	450
DataStar Data Entry & Retrieval	175
ReportStar Report Generator	150
InfoStar Advanced DBMS (Above 2 Programs)	300
dBASE II Complete With Disk Tutorial	385
Super Utility Plus 3.2 by Kim Watt	59
CP/M 2.2 for Model 4 by Monte Zuma	199
Pickles & Trout CP/M 2.2m for the Model 2-12-16. Floppy Version	179
Same thing but the Radio Shack Hard Disk	219

CP/M® & 80 COLUMN FOR YOUR MODEL III

No need to buy a new computer when you can use the Holmes VID-80 CP/M, 64K RAM and 80 column modification. This kit is easy to install and requires no soldering. You end up with a complete 64K CP/M computer with an 80 column screen that is still able to run all your existing Model III software. Now you will be able to use most of the CP/M programs that normal people do, such as dBASE II and WordStar. The regular cost of this kit is $524.00. Now Rose will fix you up for only **$399**

SUPER SALE

Same as above except complete with WordStar 3.3 installed and ready to run. Only.............................. **$499**

PRINTERS & ACCESSORIES

ANADEX DP-9725, 240cps plus COLOR	1395
PROWRITER 120cps, 10" Wide Friction/Tractor	339
PROWRITER 2, 120cps, 15" Wide Frict/Trac	595
STARWRITER F10-40cps Daisy Wheel	995
LPRINTMASTER F10-55cps Fast Daisy Wheel	1199
Rutishauser Bidirectional Tractor	199
Rutishauser Sheet Feeder, A Best One	695
A Good, Cheap Sheet Feeder for the F10's	475

Please add $5 handling charge to orders under $50

ORDER NOW! TOLL-FREE 800-527-3582

Orders Only Please

Call in your order or write to us at the address below. Texas residents call us at 214-337-4346 and deduct $2.00 from your order but you should remind me 'cause sometimes I forget. If you need technical information or service please call the Texas number as the Toll Free lines are just for orders only. Prices are subject to change without notice and are mail order only. I accept AMERICAN EXPRESS, MASTERCARD and VISA and I will not charge your card until I ship your goodies. You can send a check or a money order. I also accept COD orders but they require cash or a cashier's check upon delivery. If shipping charges are not shown please call for the correct amount. Add $5.00 handling charge if your order is less than $50.00. Shipping charges quoted in this ad are for the lower 48 states only. Orders to Canadian address add $20.00 Documentation Fee if order is $500.00 or more. Texans add State Sales tax. No tax collected on shipments outside of Texas. Be sure you know what you are buying. SOFTWARE IS SOLD ON A REPLACEMENT BASIS ONLY - NO REFUNDS. If it is defective call us for instructions. Please order from me now---I need the money and I will not jack you around. All merchandise carries the original manufacturers' warranty and all repairs or adjustments will be made by the manufacturer or his designated representative.

NEXT DAY SHIPMENT of Goods in Stock.

TOTAL ACCESS
P.O. Box 790276
Dallas, Texas 75379
214-337-4346

✓81

Hard Facts About Tandy's New Software

Tandyland

edited by Bradford N. Dixon

Photo 1. The Tandy 1200.

Radio Shack's fall line-up of new software had something for everyone, with offerings for the Models III, 4, and 2000. Tandy 2000 owners will notice a new terminal package called Softerm-2000 on the shelves as well as two programs from the popular PFS series, PFS: Graph and PFS:Write. The introduction of new business software for the 2000 shows Tandy's continued commitment to its first MS-DOS machine despite the debut of its IBM-compatibles.

For the Model 4 owner, fall brought new business programs, including Financial Management, Profile 4, Radio Shack's Money Decision series, and an educational package called Introductory Science.

Model III gamers weren't forgotten, either; Computer Diplomacy, Frogger, and Zaxxon are now available for the Model III. Radio Shack also sells a Model 4/4P version of Zaxxon.

Meanwhile, Tandy's entry into the IBM-compatible world means some big changes in Radio Shack's software market. Unlike the good old days, when Radio Shack was the primary software source for its proprietary TRSDOS operating system, the Shack now has plenty of competition for Model 1000/1200 software sales.

When *80 Micro* asked if the introduction of the new machines might put Tandy out of the software business, Tandy 1000 buyer David Frager responded, "Tandy will continue to produce software for all its machines as the need for software arises. If we see a need that isn't being filled by a third party MS-DOS software company, we will definitely examine if we should provide the product."

Frager also noted that Radio Shack's Express Order software program won't offer much MS-DOS software for the Tandy 1000 because users can get MS-DOS titles at other computer stores. Nonetheless, Radio Shack Computer Centers will carry selected third-party titles that they deem to be high-quality products.

Among third-party MS-DOS products Radio Shack is stocking are Ashton-Tate's Framework and dBase III, all of the PFS titles from Software Publishing, and MicroPro International's WordStar Professional.

Model I/III/4 owners who move over to Tandy's new hardware offerings, the Models 1000 and 1200, may be in for a case of sticker shock when they start checking out the price of MS-DOS software. For example, Lotus' son of 1-2-3, the integrated package Symphony, sports a price tag of $695.

Print and television advertising campaigns have made Symphony a visible product. But sales and conversions from 1-2-3 haven't lived up to expectations, and price may be partly to blame. Turning to Symphony's chief competitor, Ashton-Tate's Framework, won't save you money—it retails for the same price.

For those who don't mind spending as much on a program as a lot of people spend on a computer, Symphony and Framework should run as-is on the Models 1000 and 1200.

Almost as soon as Tandy introduced the Model 1200 (Photo 1), its "mirror-image" IBM PC XT work-alike, rumors about where the computers were being manufactured started to fly. According to the most prevalent rumor, Tandon Computer Inc. was the manufacturer.

When *80 Micro* asked Ed Juge, Tandy's director of computer marketing, about the story, he said that company policy dictates that the company's sources remain confidential. Juge noted that he, too, had heard the rumor, but would neither confirm nor deny it.

Radio Shack Computer Centers are polishing their image with a slick new flyer aimed at potential computer buyers for business, education, and home use. It's a vast improvement over the old flyers that were stuffed into Sunday newspapers.

Called "Radio Shack Computer Center Answers," the brochure (Photo 2) poses questions from a hypothetical computer buyer; the answers, of course, highlight a Tandy machine that exactly meets the questioner's needs. The flyer looks like a newsletter printed on glossy stock.

This new ad campaign may well bring attention to the quality image Tandy wants to communicate to buyers. It's yet another example of the Shack's long-overdue effort to solve its image problem.

Photo 2. Tandy's new brochure, "Radio Shack Computer Center Answers."

Bulletin: "There is a new version of TRSDOS 6 which contains several improvements and also includes some new features not previously available in this operating system. Upgrading to this new operating system is required."

These words are taken from a Tandy customer service bulletin dated March 15, 1984; similar notifications are mailed out for upgrades of other operating systems or software. The final paragraph of these customer service bulletins instructs the software owner to pick up an upgrade at the local Radio Shack Computer Center. All well and good, but what's a customer to do when he gets to the computer center and there's no upgrade waiting for him? It's bad enough when the new software isn't in the local store; sometimes a customer may be told it's out of stock at the Fort Worth warehouse as well.

Fear not, there is a way to get the first available upgrade even though the warehouse is empty. Tell the manager of your computer center to order the upgrade with your name and address on it. Then, the next time the warehouse receives a shipment of the upgrade, your order will be flagged and sent.

It seems that computer center managers can order upgrades in two ways: for their own inventory or by customer name and address. Inventory ordering has a lower priority than special orders and when shipments come into the warehouse, they aren't earmarked for the stores needing upgrades. If the stock is there, it's sent out, but if it isn't, then the store manager has to try again later. Only personal orders are flagged.

The logic behind the arrival of cellular remote telephones in some Radio Shack Computer Centers became a little clearer last fall with the announcement that Radio Shack had worked out a customer referral arrangement with mobile telephone companies in five major markets. The companies involved are NYNEX Mobile Communications, PacTel Mobile Access, NewVector Communications, Contel Cellular, Gencom, and Bell Atlantic Mobile Systems Inc.

Tandy's press release didn't say exactly how much Computer Center space would be devoted to telephones, but considering the number of computers available from the Shack, things may get a little crowded in those stores that support the new marketing venture.

New Threads

"Phone booth journalism" was the phrase reporters used to describe how Radio Shack's Model 100 changed their reporting habits at last year's Republican National Convention.

According to Tandy, the lap-top portable revolutionized political reporting in the 1984 election. Press planes were full of reporters using Model 100s to write stories; when the planes landed, reporters raced for phone booths to transmit their stories to host computers back at the newsroom.

Reporters from the *Wall Street Journal*, the *New York Times*, the *Washington Post*, and the Associated Press were among those using Tandy lap-tops at the convention. Tim Gallivan, AP's news technology editor, notes that AP owns about 200 Model 100s. Gallivan says that, to his knowledge, AP was the first news organization to adopt the notebook-style computer for remote reporting; the wire service had the computers even before Tandy distributed them nationally.

With one foot in the grave, Lobo Systems of Santa Barbara, CA, introduced a $795 Model 4P clone that runs LDOS and CP/M software. The Mini Max, as it's called, is designed to go head to head with Radio Shack's transportable. Lobo already has a Model III work-alike in its Max-80, which retails for $945.

Lobo's president, Robert Bledsoe, quoted in the Oct. 15, 1984 *InfoWorld*, said he didn't see any problem in staying with a CP/M machine or in competing against Radio Shack. Bledsoe hopes sales will offset some of Lobo's $1 million in liabilities. "There's enough demand for CP/M-based machines," he said. "If we can sell 100 to 200 machines a month, that will meet our needs. We aren't looking for high volume."

Lobo may be the last of only a handful of one-time Radio Shack clone manufacturers. According to the *InfoWorld* piece, LNW Computers of Tustin, CA, the only other surviving clone maker, was reportedly out of business as of late September 1984.

Another ruling from the courts in the complex area of software copyrights: Federal Judge W. Arthur Garrity of Massachusetts ruled on Aug. 31, 1984, that copyrighted programs published in a computer-user magazine may not be put on disk, duplicated, and sold by a third party.

Nibble Magazine, for Apple users, had sued Amtype Corp. for copyright infringement for selling the magazine's programs on disk. *Nibble*'s publisher, Micro-Sparc Inc., already provides such a service by selling program disks for $20–$30.

In its defense, Amtype said it offered a "typing service" to readers of *Nibble* and other similar publications. Amtype charged between $7–$10 for putting all the programs from a single *Nibble* issue on a floppy disk.

Judge Garrity was to rule on possible damages at a later date.

MicroTrends

The folks at the Hewlett-Packard Personal Computer Group in Cuper-

tino, CA, released some statistics last fall that put them in third place in the personal computer market. The former third-place company was Tandy Corp.

Hewlett-Packard made a point of the fact that it couldn't confirm or deny the validity of the statistics that put the firm right behind IBM and Apple. However, HP credits its apparent rise to third place to the success of the HP 150 Touchscreen Personal Computer.

As for market share, IBM holds first place in the $1,000–$6,000 desktop/portable market with 33 percent of sales, Apple comes in second with 20.7 percent, and HP is a distant third with 6.1 percent. Tandy Corp. now holds fourth place, according to the information supplied by HP, with 5.2 percent of the desktop/portable market (see Table 1).

Manufacturer	Percent Market Share
IBM	33.0
Apple	20.7
Hewlett-Packard	6.1
Tandy	5.2
DEC	3.0
NEC	3.0
Others	29.1

Table 1. Hewlett-Packard's ranking of the top firms in the $1,000–$6,000 desktop/portable market.

What's ahead for the business office? Frost and Sullivan, a New York market research firm, recently released a study predicting that personal computers will shape the course of office automation over the next five years. That isn't hard to believe, but the dollars involved may come as a surprise.

The report indicates that word processing applications sales will increase by 23 percent to $10.7 billion by 1988. Unit sales for word processing equipment are expected to increase by 30 percent yearly. That's because of the inroads personal computers have made into the managerial and professional ranks: According to the wordy report, "The multifunctional capability of both word and data processing of the desktop computer makes it well suited to the needs and tasks performed by those in these job levels."

Frost and Sullivan considered as word processors dedicated units and even electric typewriters, but predicted that personal computers will become the major component of the word processing market for the next five years.

Speaking of growing computer use, some new figures show a startling leap in the number of computers available to public school students. According to Quality Education Data, a Denver research marketing firm, the number of students per microcomputer has dropped by half since 1983. During the 1983–84 school year, the national average was 170 students per micro; by the beginning of the 1984–85 school year, that average had improved to fewer than 85 students per computer. The top school district is Florida's Broward County, where the ratio is 23.4 students per computer.

QED's report, called "Top Fifty Districts," also showed some gains by IBM in the battle for the educational dollar. Big Blue's share jumped from 4.6 percent of all micros in schools in 1983–84 to 6.05 percent for the current school year.

Apple computers, however, still dominate the education market, with a 47.9 percent share among large school districts. Radio Shack holds second place with 22.4 percent and Commodore is in third, according to the report, with 10.4 percent. IBM remained in fourth place despite its increased infiltration (see Table 2).

Hot Items

Software Arts of Wellesley, MA, the company that developed the classic spreadsheet program VisiCalc, has been working for some time to come up with another blockbuster program. Now it may have one: a PC desktop management program called Spotlight. Belonging to the new genre of microcomputer accessory software, Spotlight acts as a calculator, note pad, appointment calendar, telephone book, and index file deck. In fact, the program sounds very much like the Tandy DeskMate program that comes bundled with the Tandy 1000.

Desktop management programs have gained attention since Apple introduced one with their Macintosh a year ago, and the price for such programs remains fairly reasonable. Spotlight retails for $150, a moderate price for professional software, but not exactly chopped liver. If the program is comparable to the Tandy DeskMate, then the rock-bottom price of the Tandy 1000 is even more of a bargain.

Yes, but can you back it up? Elite Software Systems of Albany, NY, is staking some serious money on its claim that its new encryption program, Encomp, defies piracy. The company is so confident of its MS-DOS-based product, it's offering a $10,000 reward to any individual who can break the code. Elite will award an additional $2,000 to the dealer who sold the program.

IBM's relentless quest for more memory in less space took another step forward recently as the computer giant began producing volume quantities of a 256K chip.

The 50.5-square-millimeter chip occupies only twice as much area as a 64K chip and makes it possible to assemble more than 4 million characters on a 7- by 9-inch circuit card. The significance of all this becomes clear when you consider that in terms of real storage, you could put the contents of a 1,400-page book on a single chip. ■

	Number of Units		
Manufacturer	Fall 1984	Fall 1983	1984 Market Percentage
Apple	35,256	15,879	47.9
Radio Shack	16,495	10,098	22.4
Commodore	7,700	3,494	10.4
IBM	4,449	1,676	6.0
Atari	3,200	3,106	4.3
Texas Instruments	3,180	952	4.3
Franklin	2,151	1,102	2.9
Others	1,139	525	1.5

Table 2. 1984 computer market shares in the top 50 U.S. school districts.

Verbatim. 3M IBM®

TOLL FREE 1-800-321-3552
IN OHIO & OTHER INQUIRIES 216-481-4993

star NEC SANYO

OKIDATA EPSON Datalife TAVA PC KAYPRO II BASF

New Low Prices

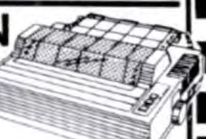
IF ANYTHING BEATS OUR SELECTION, IT'S OUR PRICES!

EPSON

RX80
$269

- 100 cps
- Tractor paper feed
- 128 Typestyles
- 7 graphic modes
- Proportional space
- 10" carriage

EPSON



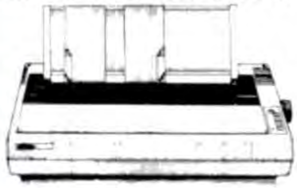
FX100
$659

- 160 cps
- Tractor paper feed
- 128 Typestyles
- 7 graphic modes
- Proportional space
- 15" carrriage

EPSON FX80

160 CPS, 10" carriage . . . $449

EPSON LQ1500



includes parallel interface, 200 CPS, bi-directional, sheet feeders available. $1199

IBM PCX

IO MEGABYTE HARD DRIVE

NOW ONLY $2995

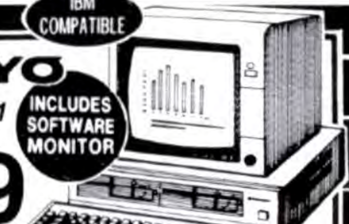
Includes MDS controller, 10 megabyte hard drive and memory, CPU, IBM keyboard, 256K RAM, mono/-parallel board, monitor and single 320K floppy drive. Easily expandable.

IBM COMPATIBLE

SANYO

The MBC 555-1

INCLUDES SOFTWARE MONITOR



$1199

System Package INCLUDES . . .

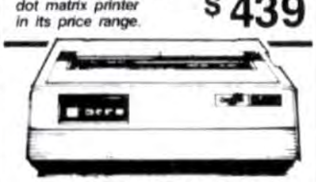
	Sugg. Retail Price		Sugg. Retail Price
MSDOS	$60	INFOSTAR	$495
SPELLSTAR	$250	WORDSTAR	$495
MAILMERGE	$250		
CALSTAR	$250		

$2999 VALUE

Features include 8088 Microprocessor, 128K RAM expandable to 256 RAM, parallel printer port, graphics capability, game paddle port, 8087 math co-processor port. Also available with double-sided disk drives — $1359.00.

OKIDATA 92

The most advanced dot matrix printer in its price range. $439



OKIDATA 93

- 160 cps
- Tractor paper feed
- Proportional space
- 15" carriage

$679

COMREX II

13 CPS Printer $439

COMREX III	$689
STAR GEMINI 15	$449
STAR DELTA 10	$479
OKIDATA 82	$349
OKIDATA 93	$679
OKIDATA 84	$799
NEC 8023	$399
NEC 2030	$899
NEC 3530	$1599
BROTHER HR 15	$459
BROTHER HR 25	$699
BROTHER HR 35	$959
PANSONIC 1090	$299
PANSONIC 1091	$349

WE'RE GIVING MAIL ORDER A GOOD NAME!

MICRO DATA SUPPLIES

22295 Euclid Ave. Euclid, Ohio 44117

IF ANYTHING BEATS OUR SELECTION IT'S OUR PRICES!

A SUBSIDIARY OF THE COMPUTER WAREHOUSE, INC. OF OHIO

TOLL FREE 1-800-321-3552

IN OHIO & OTHER INQUIRIES 216-481-4993

ORGANIZE AND PROTECT

ROLLTOP 100
Disk File Model #RT100

Twice the capacity (100 - 5¼" Disks) of the leading "flip top" file. It includes 10 diskette di viders and anti-skid fee An outstanding design . . . $36.00

Locking Model #RT100L $46.00

FREE BOX OF DISKETTES
WITH PURCHASE OF PAC 100
Limited Time Only!

BUSINESS PAC 100

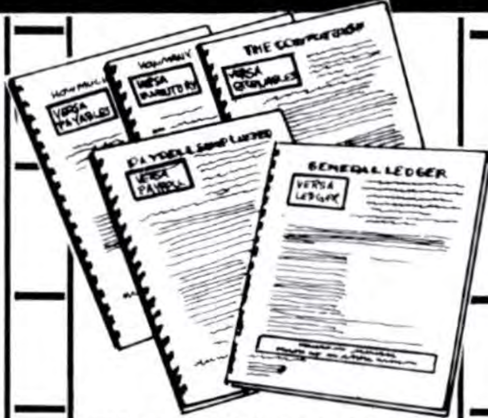
READY-TO-RUN BUSINESS PROGRAMS

$99.95

INCLUDES

128 Page Users Manual • Inventory Control • Payroll • Bookkeeping System • Stock Calculations • Checkbook Maintenance • Accounts Receivable • Accounts Payable.

PLUS 90 MORE!



VERSABUSINESS SERIES

The Most Powerful Business Software Ever!
Model I, III & IV

Each Versabusiness module can be purchased and used independently or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES $84.95
A complete menu-driven accounts receivalbe, invoicing and monthly statement generating system. Keeps track of all information relating to who owes you or your company money and can provide automatic billing for past due accounts.

VERSAPAYABLES $84.95
Designed to keep track of current and aged payables, keeping you in touch with all the information regarding how much money your company owes and to whom.

VERSAPAYROLL $84.95
Powerful and sophisticated yet easy to use! Complete employee records are maintained and all necessarry payroll calculations are peformed automatically.

VERSAINVENTORY $84.95
Just what you need to maintain complete control of your inventory system. Gives you instant access to data on any item. Keeps track of all information related to what items are in stock, etc.

VERSALEDGER II $129.95
This complete accounting system grows with your business. Can function as a simple personal checkbook register, a small business bookkeeping system or a large corporate genreal ledger system!

ELECTRIC WEBSTER

w/correcting feature $119.00
Hyphenation feature . . . . $ 39.00
Grammatical feature . . . . $ 39.00

COMPLETE SYSTEM

All four programs $189.00

"Cadillac" of Spelling Checkers
80 Micro 9/82.
Available on Model I, III & IV

ELECTRIC PENCIL Version II

Disk Version $79.95
Tape Version $69.95

Le Script $104.95

Model I, Model III, & IV LNW-80, PMC-80 MAX-80.

NewScript

7.1 . . . . $99.95
With Labels $114.95

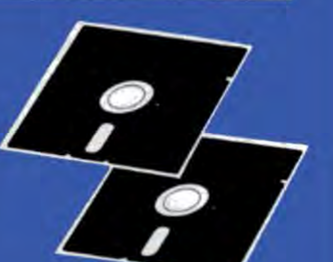
MISC. Avery Tabulables

5000 - 3½ x 15/16" $15.95
5¼" File Box for 75 Diskettes $19.95
LABLEMEAKER Kit w/400 1 up labels . $29.95
Casslabel Kit w/200 Casette Labels and 200 1 up Labels $34.95

Verbatim

DATALIFE Premium Diskettes at Affordable Prices

5¼ MD525-01 1S/D Den (box of 10)	$19.95
Special Savings! Soft Sectored Diskettes	
5¼ 2S/D Den (MD550-01)	$31.95
5¼ 2S/4 Den (MD557-01)	$49.90
8 1S/D Den (FD34-8000)	$43.95



New Low Prices

MICRO DATA SUPPLIES

22295 Euclid Ave.
Euclid, Ohio 44117

A SUBSIDIARY OF THE COMPUTER WAREHOUSE, INC. OF OHIO

Lazy Writing

After many unsuccessful attempts at trying to get my Model 4 to use the spooler with Lazy Writer, I wrote to AlphaBit Communications, Inc. The solution, though not included in the manual, is simple. Copy the SYSDRV/CMD file, found on the Lazy Writer disk, for 6.X.X to P1/CMD. When the spooler is operational, it's automatically used with Lazy Writer. I use this JCL file to activate the spooler:

```
1 SET *LM CLICK/FLT
2 FILTER *KI *LM
3 SPOOL *PR TO TEXTFILE:1 (MEM=5,
DISK=5)
4 L X X1
5 //STOP
```

Lance K. Mertz
Snohomish Publishing Co.
114 Ave. C
Snohomish, WA 98290

Better Positioning

I merge this one-line utility with my programs to get each PRINT@ correct on first trial:

```
65000 X=PEEK(16416)+256*(3ANDPEEK
(16417)):PRINT@1016,X;" "@X,CHR$(14);:
Z=PEEK(14400):Z=2*Z:Z=Z/2:
PRINTCHR$(15);:PRINT CHR$(-27*(Z=8)
-26*(Z=16)-24*(Z=32)-25*(Z=64));:Y$=
INKEY$:IFY$>=" "ANDY$<>"["
ANDX<>1023THENPRINTY$;:GOTO
65000ELSE65000
```

You'll get a nondestructive cursor that you move with the arrow keys, while your program displays the position in the lower right corner. You can even use this as a screen editor since it accepts alphanumeric input.

Thomas Eggarter
Chatadata Inc.
Ramirez Mitchell 358
5700 San Luis, Argentina

International Style

Daisy-wheel printer owners who type in foreign languages will be pleased to know that Radio Shack released a daisy wheel, Scandia, that includes both Spanish and Danish characters. You access them via the SuperScripsit System Code utility.

Carlos Ortiz
119–37 Metropolitan Ave. #30
Kew Gardens, NY 11415

CLOADing

I'm able to run "Catacomb Conquest" (August 1984, p. 78) on my cassette-based Model III using "NODOS 80" (1983 Anniversary Issue, p. 96). Hopefully, "NODOS 80" will let me run other disk-based programs from *80 Micro*.

Norman Jennings
1605 Walton Street
Cornwall, Ontario
Canada K6H 1W3

Due to hardware interface problems, the bar code reader program originally scheduled for this issue has been delayed; we will publish the program as soon as we resolve these problems.

Also, because of Tandy's involvement with the Radio Shack showcases, Ask Tandy will not appear this month; the column will resume in February.

—Eds.

For Geminis Only

Glenn Parkinson's FORMGEM ("Character Sketching," March 1984, p. 156) works perfectly on the Gemini-10X printer with only two changes. First, change the CHR$ (136) in line 1390 to CHR$(8), and then change the CHR$(132) in line 1430 to CHR$(4).

Mark Allen Reed
Box 368
West Lebanon, NH 03784

I recently purchased Model 4 SuperScripsit only to find that none of the printer drivers furnished with the program works with my Gemini-10X.

The drivers send a CHR$(15) to the printer, resulting in condensed-mode printing. You can prevent this code from reaching the printer with forms filter. Type in these instructions at TRSDOS Ready:

```
SET *FF FORMS/FLT
FILTER *PR *FF
FORMS (XLATE = X'0F00')
SYSGEN (YES)
```

The first two lines set the forms filter, while the next line converts all CHR$(15)s to CHR$(0)s. The last line writes the configuration to disk, and the printer driver now works properly.

Jerome D. Klutts
Route 2, Speedway St.
Campbell, MO 63933

Reader Aid

After unsuccessfully trying to run Model III graphics programs on the Model 4P, I realized that the Model 4P doesn't contain a Model III ROM. Because of this difference, the Model 4P prints incorrect graphics. Any suggestions?

Jim Gonsalves Sr.
2257 Manhattan Place
Santa Clara, CA 95051

We had the same difficulty and found that POKE 16892,02 corrects this problem on the Model 4P.

—Eds.

Can anyone suggest a professional astrological program to run on a 48K Model III with twin double-density drives? The program should be able to print natal and harmonic charts.

Roger G. Dawson
22 Lancaster Road
Preston, Lancashire
England, PR1 1DA

I'm looking for the automatic sequential dialer program that was first seen in the movie *War Games*. I've been told that its name is AMODEM-90 and that it's a public-domain program.

Richard H. Epson
1440 Sumner Ave.
North Charleston, SC 29406

I give up—how do I get to the last level in Deathmaze 5000? Also, I'm trying to locate a copy of a game program called Space Age 21.

George D. Madison
P.O. Box 649
Winchester, OR 97495

Error Trap

In line 8 of "Double Trouble" (Reader Exchange, October 1984, p. 29) BACKUP SYS:0 should read BACKUP SYS0:0.

Gil Seiler
126 Boas Drive
Santa Rosa, CA 95405

Incorporate this change in Alain Cirkovic's Easydata program ("Little Wonder," December 1984, p. 72) to get a directory in TRSDOS: Change Dir to D: in line 380.

—Eds.

DEBUG

Listed below are modifications to X-Mart ("Attention Shoppers," August 1984, p. 75) to eliminate the syntax error that occurs in line 290 when you try to exit the program. Delete line 1210 and make the following changes:

```
290 D1=LEFT$(C3,X-2):D2=MID$(C3,X,LEN(C3)-1):IFC3= "QUIT" 1210 ELSEIF QH<>0THEN340

1210 PRINT@576, "SURE? ";:GOSUB 1300:IFN=0THENPRINTA:GOTO 280 ELSEF=1
```

Also, to eliminate two additional bugs, change lines 710 and 740 to read:

```
710 IFC2="CAN"THENIFJ=58THEN IFE(38)=2THENPRINTDZ:FORT=1 TO800: NEXT:E(38)=0:GOTO80ELSE= BELSEA=DELSEIFC2= "POC"THEN IFE(21)=1 THENIFE(40)=2THENE(40)= 0:I(40)=J:PRINTDF:FORT=1TO800: NEXT:GOTO80ELSEA=BELSEA=D8

740 IFC2="WAL"THENIFE(5)=2THENE (5)=1:E(46)=1:I(5)=J:I(46)=J:A="YOU NOW HAVE AN"+B(5)+"AND A"+ B(46):Q=Q+2ELSEA=B
```

Finally, several readers have had problems after they have been ar-

rested for jaywalking. After falling asleep they can move around, but can't manipulate objects. This isn't a bug! After moving around approximately 20 times, the player once again wakes up in jail. Because you're not actually playing during the dream, you can't manipulate items. Pay attention to the message that appears during the dream—it helps you escape from jail.

Joshua Barinstein
124647 Victory Blvd.
N. Hollywood, CA 91606

"The Direct Approach" (September 1984, p. 96) does work on the Model 4P, contrary to what's stated in the Key Box. Also, the following changes prevent garbage collection: Line 10010 should contain an OUT 132,142 immediately following the POKE, while line 10200 should contain an OUT 132,135 immediately following its POKE.

Seth Monger
4694 Coos Bay Wagon Road
Roseburg, OR 97470

Please note the following corrections for "Bugs from Outer Space" (March–August 1984). Line 100 of Text6B should read:

```
00100 ;****PART6B
```

Change line 350 of TEXT6D to read:

```
00350 ERROR1 JP 7637H ;8637H for Mod3
TRSDOS
```

The buffer for adding scores to the scoreboard isn't properly closed. The Program Listing corrects this problem.

To assemble Text6E, load the completed game using System or Load, and then load Text6E. Save the complete game as described in the August 1984 issue (p. 88).

Roger Smith
505 Ridge Lake Road
Crestview, FL 32536

Send correspondence to Reader Exchange, c/o 80 Micro, *80 Pine St., Peterborough, NH 03458.*

Program Listing. Correction to "Bugs from Outer Space."

```
                   00100 ;****PART6E
                   00110 ;CORRECTS BUGS IN SCOREBOARD
749A               00120        ORG   749AH   ;849AH FOR MOD. III TRSDOS
749A CD1578        00130        CALL  OOPS           ;PATCH
7815               00140        ORG   7815H   ;8815H FOR MOD. III TRSDOS
7815 D5            00150 OOPS   PUSH  DE             ;SAVE DE
7816 3E20          00160        LD    A,32           ;A=SPACE
7818 0606          00170        LD    B,6            ;6 CHARACTERS
781A 12            00180 OOP10  LD    (DE),A         ;STORE SPACE
781B 13            00190        INC   DE
781C 10FC          00200        DJNZ  OOP10          ;LOOP
781E D1            00210        POP   DE             ;RESTORE
781F CD0362        00220        CALL  6203H   ;7203H FOR MOD. III TRSDOS
7822 C9            00230        RET
0000               00240        END
00000 Total Errors
OOP10    781A
OOPS     7815
```

End

NEW! *Lower Prices!!*

wabash

six-year warranteed

DISKETTES!

$11.90/box (10)

5¼" single -side, single-density; double-density add $2/box. Add $3 per order shipping. *In Illinois add 7% sales tax.*

Library Case-$1.50 (with disk purchase)

Immediate shipment on VISA, MasterCard or Money order; add 14 days for personal checks.

CALL TOLL FREE
(800) 222-1248
In Illinois Call (312) 882-8315

DEALERS! SCHOOLS! USER GROUPS!
Call for our volume discount prices! ✓367

DIGITAL IMAGES
1185 Tower, Schaumburg IL 60195

LNW SERVICE

QUALITY REPAIRS ON ALL LNW FACTORY ASSEMBLED PRODUCTS INCLUDING:

- TEAM AND ALL LNW80 MODEL COMPUTERS
- SYSTEM EXPANSION II
- LNDOUBLER AND LNDOUBLER 5/8
- LNW DISK DRIVES

COMPUTER UPGRADES:

- CP/M 2.2
- HARDWARE 80 × 24
- MODEL 4 UPGRADE
- TEAM UPGRADE

OTHER PRODUCTS AVAILABLE:

- HARD TO FIND LNW PARTS
- LNW SOFTWARE

Editor Note:
This space open for specials!

WILSON TECHNICAL SERVICE

8:00 AM-5:00PM(PST)	AFTER 5:00 PM(PST)
VOICE 714-531-8136	DATA 714-531-8136
	300/1200 BAUD

✓123

Attach Your Printer to SuperSCRIPSIT

Easily and automatically attach your printer to SuperSCRIPSIT with an ALPS software driver program. ***Over 100 different printers supported.*** No need to learn special codes. Call or write for latest Product Review Sheet describing features supported on each printer (underline, bold, proportional, etc.) Two years experience. Customer support.

$49 or $59 each, depending on printer

Use Memory as Superfast Disk Mod 4(III) — $39

Speed up programs by using your extra memory as a drive. Use MEMDISK as your *System Drive*, or load files into memory for ***instant access***. Easy to use with 1 simple command. No program changes required to access MEMDISK as if it were a disk drive. An entire MEMDISK can quickly and easily be saved to (or loaded from) floppy disk. Uses 80K of 128K memory (64K on 4P). (LDOS only.)

TIME-SAVERS FOR TANDY 2000

(and IBM PC)
IMPROVE PERFORMANCE & FLEXIBILITY OF YOUR PC APPLICATIONS

Use Memory as Superfast Disk — $49

RAMDISK - Use spare memory as a disk to speed up applications. (You specify RAMDISK size.) *Appears to programs as a disk, but gives instant access.* Easy to install and use. Superfast!

Directory / File / Backup Tools — $59

1. ***Display alphabetized 5-column directory listings.*** View entire directory at a glance on 1 screen.
2. ***Find a file*** - search thru all directories for a file name and display all occurrences.
3. ***Directory Tree*** - display directory tree structure below any name in easily readable form.
4. ***Backup only modified file*** - speed up and automate backup operation. Stores in same format as original for direct access. Great for hard disks!

User Tool Assortment — $59

1. *Define a command as one key.* Keyboard aid - less typing and faster data entry by defining any command string as 1 key, then press key to perform.
2. *Count lines on files.* Displays count.
3. *Dump files in Hex and character* (with scrolling).
4. *Change ownership & attributes of files.* You can make files hidden, read-only, or reset to original.

Attach a Serial Printer — $49

With 2 easy commands, you can attach a serial printer to the RS 232 port. Complete instructions.

More ALPS Tool are available. Call or write for catalog. All products are guaranteed to perform as advertised. We provide full customer support.

ALPS ✓74
23 Angus Road
Warren, New Jersey 07060
201-647-7230

- **DISPLAYS CORRECT SPELLINGS:** If you don't know the correct spelling, EW will look it up for you, and display the dictionary.
- **VERIFIES CORRECTIONS:** If you think you know the correct spelling of a word, EW will check it for you before making the corrections.
- **HYPHENATES AUTOMATICALLY:** (Optional). Inserts discretionary hyphens throughout text.
- **GRAMMAR & STYLE CHECKER:** (Optional). Identifies 22 types of common errors. Makes suggested corrections with the stroke of a key. Runs within EW.
- **50,000 WORD DICTIONARY:** Uses only 2½ bytes per word; add as many words as you wish.
- **FAST CORRECTING:** In as little as 30 seconds, Electric Webster can return you to your Word Processing program, with your text fully corrected and on your screen.
- **INTEGRATES:** into 12 different word processing programs: WordStar, Spellbinder, Palantir, NewScript, Lazy Writer, SuperScripsit, Scripsit, Electric Pencil, CopyArt, Superscript, Zorlof, and LeScript (specify). Begins proofing at the stroke of a key; returns you to word processing automatically.

"The Cadillac"

of spelling checkers
80 Microcomputing, 9/82

VOTED #1: For the second straight year, Electric Webster was voted the #1 spelling checker in the *80 Micro* Reader's Choice Awards. (1/83, 1/84)

"The most helpful program I've found is Electric Webster. After looking at nine proofreading programs, I've settled on Webster . . ." *Creative Computing, 11/83*

This dictionary is not published by the original publishers of Webster's Dictionary or their successors.

ACCLAIMED:
"Electric Webster is the best. Just read any review in any magazine and I don't believe that you will find even one disagreement to that statement." *CINTUG, Cincinnati's Users Group Mag., 4/83*

"Now there's a program called Electric Webster that would let me write to Presidents and Kings and never feel embarrassed. Miss Mulberry would give Electric Webster an A+, and so will you." *Computer User, 1/84*

Performance "Excellent"; Documentation "Good"; Ease of Use "Excellent"; Error Handling "Excellent". *Info World, 8/82*

LOW PRICES:

TRS-80 Electric Webster	$ 89.95
w/Correcting Feature	$149.95
Hyphenation	$ 49.95
Grammar & Style Checker	$ 49.95
CP/M, PC/DOS or MODEL 2000	
Electric Webster (with Correcting Feature)	$209.95
Hyphenation Option	Included*
Grammar & Style Option	Included*

*Limited Time Only

CORNUCOPIA SOFTWARE ✓45

(415) 524-8098
Post Office Box 6111 Albany, California 94706

T/Maker: Functional CP/M Integration

by Eric Grevstad

edited by Ryan Davis-Wright

Illustration by Katherine Mahoney

The age of integrated software didn't begin with the IBM PC and Lotus 1-2-3. Since 1980, programmer Peter Roizen has been merging CP/M-80 applications into T/Maker, fitting the software concept of today with the hardware and operating system of yesterday. T/Maker includes word processing, spreadsheet, and data base management capabilities in one package.

The latest T/Maker (version 4.01) is no threat to new PC packages like Lotus' Symphony or Ashton-Tate's Framework, but the CP/M world has nothing else like it. If you have the patience to learn its syntax, T/Maker delivers genuinely flexible integration. It's an impressive piece of programming, as long as you don't compare it to dedicated word processors or spreadsheets.

A Clean Slate

To give T/Maker a high compliment, it's as close to a blank sheet of paper as a Model 4 program can get. When you boot up the system, it presents you with a blank screen

T/Maker has a switchable menu to execute its English-language commands and supplies on-line help for its sometimes cryptic keyboard functions. While the program may seem difficult to use at first, T/Maker tries to make things easy for you.

Don't be afraid when you open the box and five disks fall out. You'll need the first disk each time you start the program (and whenever you want help), but you'll do most of the work with the second (editing, spreadsheet, data base, and printing) and third (list processing, graphics, and spelling checker with 15,000-word dictionary) disks. The fourth disk contains a 35,000-word auxiliary dictionary, and the fifth is a demonstration/tutorial.

Since T/Maker usually reserves drive B for data, you'll be swapping program disks in and out of drive A. Invoke a disk-3 command with disk 2 in the computer, and a "Command Failed: Nonexistent" message prompts you to try again.

You can change the data drive or other defaults with a program called TModify, which also lets you reconfigure the T/Maker keyboard rather than use its standard pattern, awkward for things like cursor movement.

Some of this has already been done for the Model 4/4P version. The arrow keys function and an addendum to the manual tells you how to emulate other CP/M terminals (the TRS-80's break key works the same as others' escape keys, and control-L replaces the missing tab key). While T/Maker comes configured for 128K RAM and CP/M Plus, it also includes set-up instructions for Montezuma Micro's 64K CP/M 2.2.

Commands And Conversations

While other programs prompt you with question marks, T/Maker's interface is a chatty "What Next?" You answer it with English commands: "Create" starts a new file and prompts you for a file name, and "Edit" brings you into the file. You can save time by entering commands together (Create Newfile Edit, or Align Save Print It), though the instructions grow less grammatical as they grow more detailed; Print Nonstop From 10 To 30 It, for example, prints pages 10–30 of the current file without format queries.

Many of T/Maker's vital commands only work downward. A command to align or print text, for example, affects only the portion of the current (RAM-based) file below the top line shown on-screen. In most cases, you'll either move to the top of the file beforehand or use T/Maker's framing command, which is similar to a spreadsheet's ability to freeze a column of labels while viewing other data.

T/Maker's most important command, though, doesn't directly affect anything. Edit (E) removes the "What Next?" prompt and activates T/Maker's editor, the heart of the T/Maker system. It's a free-form program that lets you put anything you want anywhere on the screen, but it's a slight exaggeration to call it a word processor.

You create every T/Maker file, whether it's a document, spreadsheet, or data base, through its editor.

As mentioned, the editing screen is initially blank. Pressing break and then the ? key calls up a status line that indicates the current file's name, length, free space, current cursor position, and carriage-return mode. (To explain the last, break-R rotates among manual returns, word wrap after 63 characters, and spreadsheet-style horizontal scrolling, while break-I toggles an insert mode. A ruler below the status line helps you set margins and tabs.)

T/Maker Integrated Software

★ ★ ★

T/Maker Company
2115 Landings Drive
Mountain View, CA 94043
415-962-0195
Models II, 4/4P (requires CP/M), Tandy 2000
128K RAM
Two disk drives
$450

Easy to use?	★ ⯪ ☆ ☆ ☆
Good docs?	★ ★ ★ ★ ⯪
Bug free?	★ ★ ★ ★ ☆
Does the job?	★ ★ ★ ☆ ☆

Running under Model 4 CP/M, the editor suffers from poor typing response and keybounce (if I were using it to write this, it'd be a review of Tt/Mmaker) and it lacks some useful functions, such as cursor movement that wraps around lines or a way to delete words as well as characters. There are no true block-move operations, but you can copy lines one at a time into a 40-line buffer for movement (first in, first out) elsewhere.

On the positive side, the editor's embedded print commands are convenient: ".single" and ".double" control line spacing, and ".newpages" starts a new page if fewer than five lines are left on the current one. Footnotes are handled automatically, and search and search-and-replace functions (break-' and break-" respectively) are fast and accurate.

Words Under Wedges

Oddest of all is T/Maker's approach to margin setting and justification. Instead of embedding commands or keystroke sequences, you set line width by typing in pairs of wedges (<< >>) delimiting text boundries. The wedges shape the text below until T/Maker encounters the next set of wedges. Minor changes, such as a right wedge of −>> versus >>, yield justified, ragged, or centered copy for block-style or indented paragraphs.

T/Maker doesn't format text until you leave the editor and use the Align command, which reads the wedges and adjusts the text accordingly. It also runs paragraphs together unless there's a blank line or a single period between them, turns columnar tables into prose if you forget the "do not align" wedges (>> <<), and puts two spaces after question marks, exclamation points, and periods (including the one in Mr. Jones).

While T/Maker's word processing features make you long for Scripsit, its spelling checker is quite good, in an odd sort of T/Maker way. Entering the Proof command (with the first line of your file at the top of the screen) produces a handy table tallying the file's total number of words, unique and unfamiliar words, and the average number of characters per word and words per sentence.

When you return to the editor, you'll not only find the unrecognized words tagged in your file but a double list, with two copies of each offending word per line, appended to it. Fix the second sample of each word and it'll be changed throughout your file when you issue the Correct command, or added to the dictionary with Amend. You can even ask for help, changing the second sample to something ambiguous like "rhy?hm" and asking T/Maker to spell it.

Monster Math

I also like T/Maker's skill with lists and tables. Once you learn the lingo, it sorts and shuffles columns of data easily and quickly. If the first item is at the top of the screen, the command Sort A N 12 32 puts the numbers in columns 12–32 in ascending order. You could also sort a list of characters, or sort in descending order.

A similar command, Tally, is more explicitly mathematical. It takes lists (files with plus signs in the first column of each line), breaks them into categories by column number and, if you want, sums the numeric values associated with them. For example, I could tally the second and third columns of a list of dinner guests, their sex and their ages, and learn there were three males and four females, with combined ages of 82 and 103, respectively.

The further you get into mathematical work, though, the more T/Maker's word-based structure hinders you. It works best with vertical lists. With plus and minus signs running down the left margin and an equals sign at the bottom, the Compute command can solve the list as if it were an algebra problem.

Beyond this, you're into T/Maker's spreadsheet. Like its other program features, the spreadsheet is great at merging data into business letters but terribly difficult to master. If you can imagine a word processor with no Delete Word function, imagine a spreadsheet with no cells, parentheses, or automatic recalculation for "what-if" changes.

T/Maker's way of defining column widths (up to 50 computable columns) isn't necessarily bad. The top line of a spreadsheet is an example line (labelled "ex") and contains maximum width numbers such as 99,999.99. Any numbers below them, even if not perfectly lined up, will be realigned to match the examples (with or without commas, and with the same number of decimal places).

The rest is like VisiCalc in reverse Polish notation. All mathematical functions (from + and / to exp and abs) appear not in row and column intersections, but externally, running down the left side or across rows atop or between spreadsheet entries. This system works for simple sums, but calculating a specific cell without anything like a cell formula is murderous.

In addition, T/Maker doesn't use parentheses. Its inflexible method of operation is to do all additions, subtractions, multiplications, and divisions in that order. To overcome this, you must sprinkle the spreadsheet with instructions: ac1 (first step, compute for all subsequent rows), jc2 (second, just compute for the next row), rc3 (third, rarely compute only on subsequent rows derived from other rows).

After studying the manual's examples and typing them in, I'll admit that T/Maker's spreadsheet can do some sophisticated things, such as built-in notations for functions like growth rate and rounding interim values. But doing them requires learning a whole new approach (even stranger than dBase II language) that applies to no other spreadsheet on the market. The system almost requires enough pre-planning with pencil and paper to defeat the whole purpose of spreadsheet work.

Keeping Records

Clearly, the spreadsheet is T/Maker's worst feature. I'm a lot happier with its data base manager which, while requiring similar pains in setting up, performs swiftly and capably once the job is done.

The data base, as you'd expect, is a text file created with the editor. It begins with sections (<form>... <end> and <record>...<end>) describing how a record will appear on the screen and how T/Maker will file it. A record can have up to 120 fields of 80 characters apiece, with field lengths defined by curly braces (left-shift/@ and right-shift/@ on the Model 4 keyboard).

Once you define a data base, the Update command works much like the regular editor, letting you review or alter records one at a time (the search function for individual items is unchanged, the "next word" cursor movement command becomes "next record"). As with lists, you can rearrange a data base alphabetically or numerically, in ascending or descending order.

For more precise work, the Select command finds groups of records easily. Altering all the records in a data base is as easy as reentering the file, changing the form and record definitions, and adding another definition for use with the Set command (such as Status = Prime When Income > $50,000).

T/Maker is no R:Base or Profile, but its data base is a nice variation on the text-oriented theme. When using Update, for example, the break-? status line changes from an editing ruler to an indication of how many more records of the current length can fit in a file.

Mixed Impressions

There are other good things about T/Maker. While it doesn't have windows, the List command lets you look at a second file without losing the current one in RAM. It's a breeze to chain, merge, and insert different files of different types for printing (one print command strips away ugly example and instruction lines from the spreadsheet).

The command menu is handy for beginners, though users will soon skip it in favor of stringing together multiword commands. The help screens are excellent: well-organized, well-written, and valuable even to seasoned users who haven't used a particular function in a while.

And the documentation is first-rate. There is a superbly organized 400-page manual and two reference booklets, as helpful and readable as anything I've seen in some time. It's relevant to remember that T/Maker has been around since 1980, gathering a group of hardcore fans. It's not only thoroughly debugged, but thoroughly documented.

I also used T/Maker on the Tandy 2000, and liked it better on that machine than on the Model 4. It was considerably faster, not so much in terms of commands like Align (for which T/Maker goes to disk) but in terms of general feel and responsiveness. The 2000 T/Maker has a full invaluable implementation of function keys: PgDn sure beats break/control-P for "move to end of line."

Is T/Maker worth $450? It's certainly worth the increase from the prior version, T/Maker III (still available at $275, but with no spelling checker, no data base, and no on-line help). It does look good when you compare its cost to several separate programs, which aren't able to merge or create such versatile files. And none can compare to its blank-paper approach only now being seen in MS-DOS products like Aura and Enable. On the other hand, the standalone programs beat T/Maker's performance in some vital areas.

T/Maker is an extremely versatile, somewhat awkward text editor with fast and intelligent search functions and limited mathematical capabilities. If Framework is the sleek new Porsche of integrated software, T/Maker is an old Plymouth Road Runner with a big V-8 and a bad clutch. While it may be inefficient and hard to maneuver around town, it has great power and performance on the straightaway.

I wouldn't choose T/Maker's editor over a real word processor and I wouldn't choose its spreadsheet at gunpoint, but it's the closest thing to real integration that CP/M will ever see. It is certainly worth considering. If you only want to learn one set of commands, it's worth using. ■

Photo. The CGP-220 printer.

Color Me Beautiful: Radio Shack's CGP-220 Color Printer

by Bradford N. Dixon
80 Micro Technical Editor

Radio Shack's CGP-220 ink jet printer is compact, quiet, and colorful. But the CGP'S most remarkable feature is its low price: For $699, home computerists and small-business users can get high-quality color printouts of screen graphics.

The CGP-220 is compact (15¾ by 11 5/8 by 4¼ inches) and lightweight, weighing only a little over 12 lbs. It has a spare and uncluttered design, with the operating switches on the unit's right front side. The Ready, Alert, and Power indicators are also located there. The Reset, Paper Feed and On/Off line switches are touch sensitive, lying flat on the front.

The slots for the ink jet's color and black ink packets are hidden from view by a door on the front. The ink packets come inside plastic cartridges so you never handle the ink. On the left, underneath the paper cover, is the ink pump. Before you can do any printing, you have to prime the ink jets with a few short pushes on the green pump lever.

ATTENTION
SUBSCRIBERS

We occasionally make our mailing list available to other companies or organizations with products or services which we feel might be of interest to you. If you prefer that your name be deleted from such a list, please fill out the coupon below or affix a copy of your mailing label and mail it to:

CW Communications/Peterborough
80 Micro
PO Box 981
Farmingdale, NY 11737

Please delete my name from mailing lists sent to other companies or organizations.

Name ____________________

Address ____________________

City __________ State ________ Zip ________

80 MICRO

Figure 1. The CGP-220 took about three minutes to make this screen dump graphic.

The power cord connector is on the back, with parallel and serial interface connectors, as well as a 600-/2,400-baud switch.

Features and Performance

After you're acquainted with the machine and have inserted one of the three sheets of paper included with it, you can start printing. While the print speed is an unimpressive 37 characters per second (for the full ASCII and special-character sets) this is of little importance, since you'll probably use the machine almost exclusively for graphics. In the graphics mode, the CGP-220 moves along at 2,630 dots per second and features bidirectional scanning. The printout in Fig. 1. took about three minutes to produce.

The ink jet prints in seven colors (black, red, green, blue, yellow, magenta, and violet) by mixing the yellow, magenta, and cyan base colors from the color packet cartridge. Even the most demanding designs are clear and sharp, and none exhibit color bleeding (see Fig. 2).

The quality of the paper used to produce the screen printouts makes a lot of difference in its final appearance. The CGP-220 will print on cut sheets or on roll paper, but not on form-feed paper. Radio Shack sells roll paper (catalog number 26-1333, three rolls for $9.95) and single sheets (catalog number 26-1341, 250 sheets for $6.95) for the ink jet printer. While I tried other types of stock, including high-quality bond paper, photocopy machine paper, form-feed paper (without the pin holes), and special paper from a Diablo ink jet printer, the Radio Shack paper produced the best results.

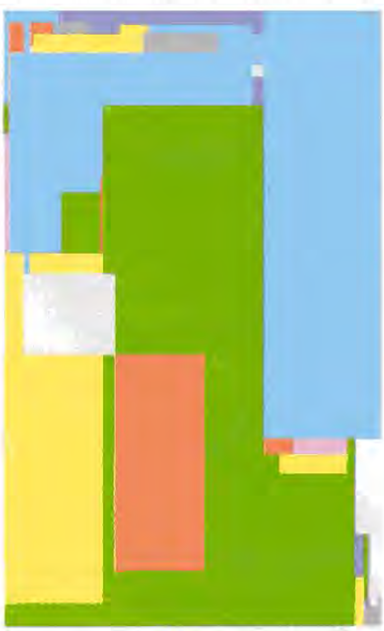


Figure 2. All the designs were printed clearly, with no color bleeding.

Using the CGP-220

The easiest way to use the ink jet printer is in the text mode, the default mode at power-up. You select the color of print desired and whether you want normal or elongated characters as the print style. You can issue print commands in all print modes from Basic without the cumbersome use of DIP switches inside the printer.

While print quality in the text mode isn't as good as Radio Shack's other dot-matrix printers, it is readable.

The CGP-220 was made to create graphics and in that capacity it performs superbly. When you use it with Tandy's Model 2000, which has one of the best screens available for color graphics, you lose virtually none of the resolution in transferring a picture from the screen to paper. Although there are differences in color because of code incompatibilities in the computer and the printer, you can avoid these with careful programming.

The MS-DOS disk that comes with the Model 2000 contains a screen dump program for the color printer (CGPDMP.BIN) which you call from Basic when you want a screen dump. On Radio Shack's Color Computer, another screen dump program is available that transfers intricate designs from the CoCo's screen onto paper.

CGP-220 users don't have to use a Model 2000 or a CoCo to get a lot out of the printer. Programmers using the Models I, III, or 4 can also take advantage of color graphics by programming the printer directly from Basic in the CGP-220's bit-image mode or the color-scan mode.

The bit-image mode lets you address each of the 640 horizontal dots per line and seven vertical dots to produce colorful designs and business graphics without a color computer. Of course, the biggest drawback to using one of Radio Shack's non-color computers is that you can't see what you're creating until it's printed. Another disadvantage of using a Model III or 4 with the CGP-220 is that you can't use the screen dump command to produce hardcopy because the character set in the printer does not recognize TRS-80 block graphics. If you press the shift/down-arrow/asterisk key combination on a Model III, any block graphics on the screen will appear as periods on the CGP-220.

The color-scan mode on the ink jet printer is similar to the bit-image mode, but allows for greater manipu-

CGP-220 Color Printer

★★★★

Tandy/Radio Shack
One Tandy Center
Fort Worth, TX 76102
$600

Easy to use?	★★★★☆
Good docs?	★★★½☆
Bug free?	★★★★☆
Does the job?	★★★★☆

lation of color. This mode uses large amounts of memory, however, and programming in it is difficult from Basic.

The Documentation

The printer's manual makes using the machine effortless. It's well-written and easy to understand, with examples to illustrate the CGPs capabilities.

The manual's lengthy table of contents directs you to areas that describe advanced features without searching through the 52-page book. Separate chapters in the manual describe the process of color printing, setting up the printer, descriptions of the various control codes, and lessons on the three kinds of printing available with the CGP-220.

Radio Shack also provides extensive appendices with additional programming tips, care and maintenance suggestions, troubleshooting aids, and schematic diagrams of the machine. Any user can get the most out of the manual whether a novice or a seasoned programmer.

While the text quality isn't outstanding, it's adequate as dot-matrix printers go.

Conclusions

With a machine like this, you'll want to try out different color combinations and graphics. The CGP-220's ink cartridges let you print up to 4 million characters before you have to replace them. Replacement ink cartridges cost $9.95 for the black ink pack and $14.95 for the tri-color. I've used the CGP-220 for many complex screen shots for three months and still have plenty of ink left.

While the text quality isn't outstanding, it's adequate as dot-matrix printers go. In terms of the graphics this machine produces, it's tough to beat for the price. Radio Shack has the CGP-220 listed with their Model 100 and Model 2000 as part of their Advanced Technology Series of products. ■

Programming with The Producer

by Hardin Brothers

The Producer
Professional Program Writer

★★★★★

Software of the Future
P.O. Box 1245
Arlington, TX 76004
Models I, III, and 4, 48K RAM
Two disk drives
Printer optional but recommended
$199.95

Easy to use?	★★★★½
Good docs?	★★★★★
Bug free?	★★★★½
Does the job?	★★★★★

The Producer won't land you a job in Hollywood, but it will let you create your own Basic data-handling programs with a minimum of bother. Even if you know nothing about programming, you can develop your own powerful data base managers. And if you're only moderately skilled in Basic, The Producer will create working program modules that you can easily add to your own software.

The longer I used The Producer, the more impressed I was. After only one month, it had found a place among my most-often-used software.

Using The Producer

The Producer comes in a hefty package that includes the documentation (several hundred pages) in a three-ring binder with reference dividers; a sample evaluation disk; the master disks, which include automatic back-up routines; and a three-hour cassette tutorial, which helps you through any rough places in the documentation.

The easiest way to explain how The Producer works is to describe how you would use it to create a data-handling program. When you boot up the Producer's master disk, an Auto program reserves some high-memory space and runs the main program. When the menu comes up, you turn on the printer, press the A and 1 keys, and print a planning form.

Using the form, you pick the program's name, state its main purpose, list the fields you want included in each record, and make any calculations that your program will apply to those fields.

Then return to the main menu, format a disk, and start to define your program. The Producer comes with a stripped-down version of DOSPLUS 3.4 that you must use to create the program, although you can transfer the completed program to any DOS.

The Producer includes a more-than-adequate editor to create an input screen for your program. You can mix graphics and text, and add large-font letters, borders, message areas, and field areas.

When you're satisfied with the screen's appearance, The Producer translates it into lines of Basic and then goes to a screen definition mode. It asks you to define the location and length of each of your fields, as well as the characters allowed for input for each. You must also define an area for The Producer to use for messages. There is an optional custom prompt area in which you can display messages that will help the program's operator respond to each field question.

After you draw and define the screen, you enter a mode called Edit Basic Data. Here you can create custom prompts for each of the fields defined earlier. These prompts will appear either when the program's user begins to enter data in each field or when he presses the clear key; the choice is yours.

Also, while in the edit mode, you define the calculations to be used on the numeric data in various fields. You may decide to either save the results of the calculations along with the rest of the information in each record or display the results for the user's benefit.

The First Draft

Finally, you're ready to create the first draft of the finished program. One of the selections from the main

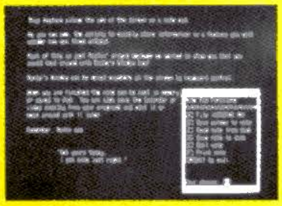
Continued on p. 160

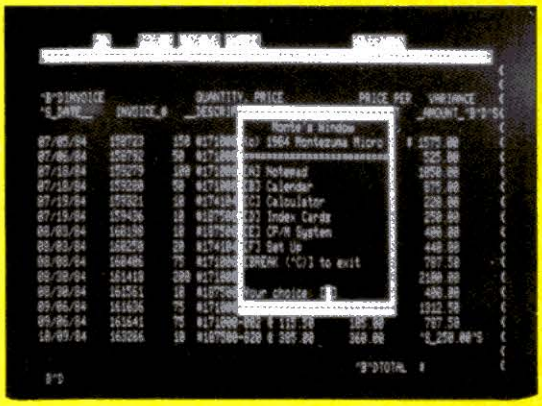
MONTEZUMA MICRO

PRESENTS

MONTE'S WINDOW™

NOTEPAD





CALENDAR

WINDOWS ON YOUR MODEL 4!

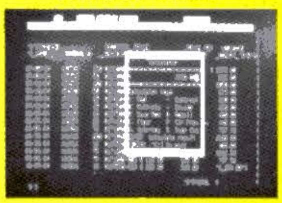
TAKES NO USER RAM!

411

VISA MasterCard AMERICAN EXPRESS

REQUIREMENTS
Montezuma Micro CP/M® 2.2 version 2.21+
128K RAM
Model 4 or 4P
8-bit Fever

CALCULATOR



Pop Up Menus!

$49

Easy to Use!

INDEX CARD FILE

A touch of the keyboard opens a window in your screen for – a Note Pad, an Appointment Calendar, a Calculator, even a Mini Data Base. All yours for just $49! Need RAM? Monte's Christmas gift to you – 64K and the window, both for **$99!**

Once Upon A Time,

Monte Zuma, our Founder, President and King, has always had trouble keeping his desk organized. The Sidekick™ from Borland International would solve the problem, but alas, it was not available for CP/M®. So Monte asked his favorite nephew, the legendary LaMont E. Zuma (distant cousin to Rondo Talbot, a direct descendant of Monte Zuma hisself) to work on the problem as best he could during recess at the home. LaMont, a true legend in his own time, really outdid himself this time. A touch of both shift keys halts your application program in its tracks and up pops **Monte's Window™** *ready to use. What could be simpler? Put an end to the fumbling and pawing around the pile of papers on your desk. You will find* **Monte's Window™** *indispensable. When you are finished, break back to your application program and it resumes without error.* **Monte's Window™** *is truly a breakthrough. See for yourself – Look through* **Monte's Window™** *on* **your** *Model 4. How did you ever get along without it? See the page opposite for order information.* **Monte's Window™** *is available right now.*

CP/M...The Software Key That Unlocks Your Model 4

CP/M is the standard 8-bit Z-80 operating system and many thousands of programs have been written to run under this system. With Montezuma Micro's CP/M you can run these programs on your Model 4/4P. Think about all those nationally known programs you've wanted to use. Programs like WordStar; dBASE II; SuperCalc; MultiPlan etc. With our version of CP/M 2.2 all those public domain programs on bulletin boards across the USA are available for free downloading. CP/M is the missing link that joins all this software to your Model 4/4P. Montezuma Micro's CP/M comes ready to use and requires no hardware modifications. This product has been awarded the best and highest ratings in the reviews and we are continuously improving it with you in mind. With our CP/M you get more than just a DOS. You get the other half of your Model 4/4P.

AVAILABLE NOW FOR IMMEDIATE SHIPMENT – Less Hard Disk Drive Support............ **$169**
Optional Hard Disk Drive Support .... $ 30

(Radio Shack 5M, 12M, 15M, 35M • Aerocomp/Percom 5M, 10M, 15M, 30M • Bi-Tech 5M, 10M, 11M, 15M, 20M, 30M, 40M)

FEATURES

- Full range of floppy drive support. Dual-head and/or 80 tracks.
- Optional hard disk support allows positioning and selective assignment of logical drives. Easy backup routine.
- Memory drive allows the use of the other 64K RAM bank on 128K machines.
- Modem 7, a powerful public domain communications program furnished at no charge, allows for file transfer and remote database access such as CompuServ and The Source.
- CONFIG is our flexible utility that allows complete control of all operating parameters from menus. Format, read and write more than 30 different manufacturer's disk formats with more being added rapidly.
- Disk Utility Program allows fast format, backups and verifying of ours and other manufacturer's disks.
- These CP/M utilities are included: ASM; DDT; DUMP; ED; LOAD; MOVECPM; PIP; STAT; SUBMIT; SYSGEN; and XSUB.

WHY BUY OURS?

	MONTEZUMA MICRO	RADIO SHACK
Transient Program Area (TPA)	55K	52K
Bytes free in MBASIC	30,776	18,488
Bytes free of formatted disk	196K	160K
64K Memory drive	YES	NO
Double-Side/80tk drive support	YES	NO
Format, read/write other CP/M formats	YES	NO
Communication program included	YES	NO
Share HD with TRSDOS/LDOS	YES	NO
Assign multiple drives to HD	YES	NO
Boots from Hard Disk (4P)	YES	NO
Popular terminal emulation	YES	NO
User defined function keys	9	3
Timely product support	YES	NO

128K MEMORY UPGRADE

Our upgrade kit includes 64K RAM, a geniune PAL chip and instructions for installation. This kit will upgrade your 64K Model 4 to 128K and allow the use of our MEMLINK program and the TRSDOS 6.x MEMDISK. Guaranteed 1 year.

A BARGAIN AT ONLY $74 Model 4
$64 Model 4P – No PAL

This popular software is available for the Model 4/4P using our CP/M.

WordStar Fast memory-mapped version 3.3 ..... $250
MailMerge Multi-purpose file merging program ... 125
SpellStar 20,000 word proof-reader on a stick 125
StarIndex Creates indices and Tables of Content... 85
DataStar Data entry and retrieval is yours 175
ReportStar Report generator and file manipulator.... 150
InfoStar The above two programs................. 300
dBASE II with Disk Tutorial........................ 385
CBASIC version 2.8................................. 85
TURBO PASCAL by Borland. This is the one........ 45

ORDER INFORMATION

Call now and your order will be shipped immediately. We accept American Express, MasterCard and Visa plus we ship COD (cash or cashier's check only). Credit cards are not charged until your order is shipped. Add $4 shipping per item on orders within the 48 states. Suitability of the software is the responsibility of the purchaser as there are **NO REFUNDS.** Defective items will be replaced upon their return, postpaid.

ORDER NOW . . . TOLL FREE

800-527-0347
800-442-1310

The Toll Free lines are for orders only.
Specifications subject to change without notice.

MONTEZUMA MICRO

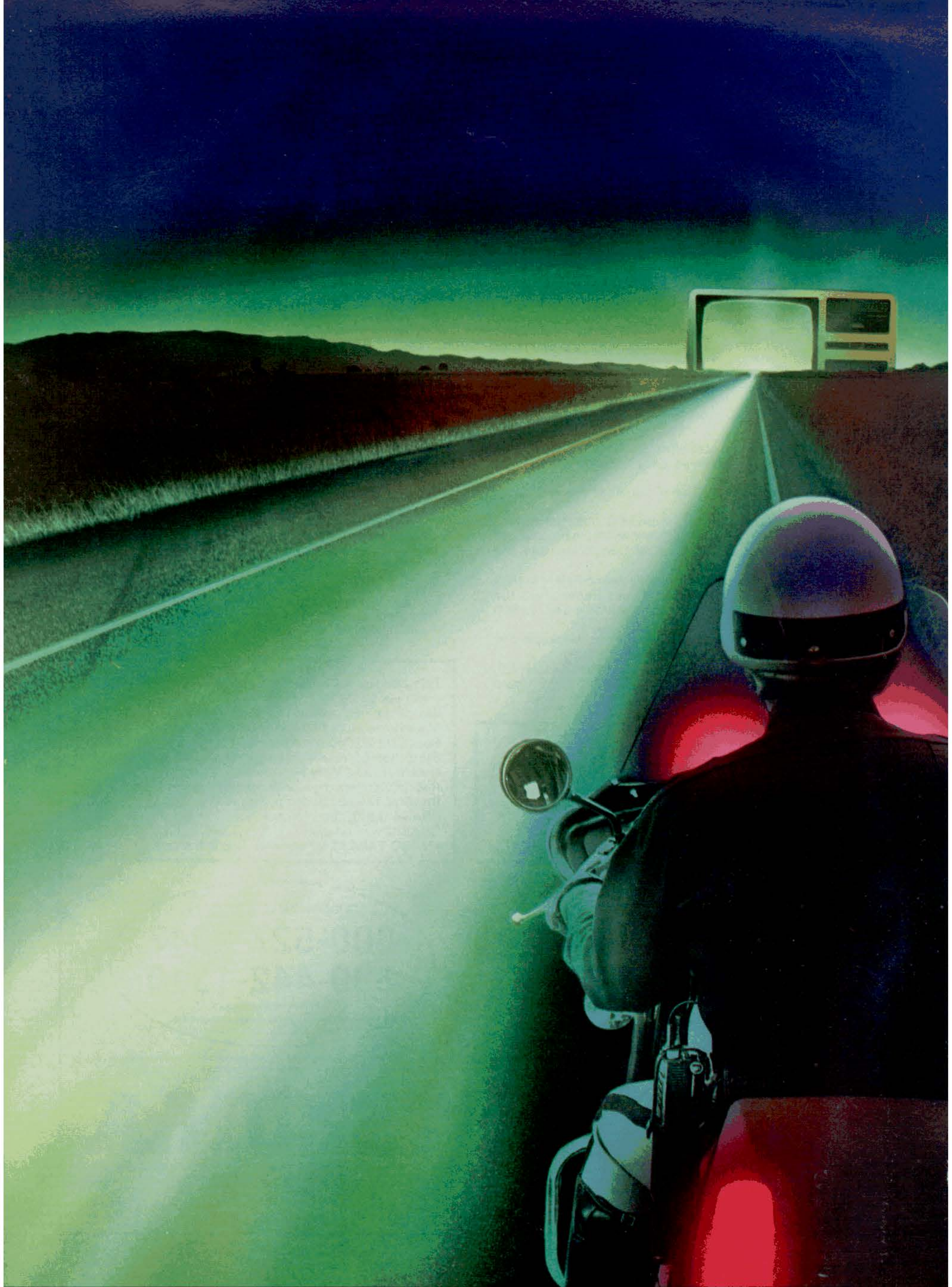
214-339-5104
Redbird Airport, Hangar #18
P.O. Box 32027
Dallas, Tx 75232

411

"WE KEEP YOU RUNNING"

VISA MasterCard AMERICAN EXPRESS

See List of Advertisers on Page 145



Running Like the Wind

by Thomas L. Quindry

***80's* own Basic compiler is back and better than ever—it compiles more Basic commands and it's easier to use than our original version, and it now runs under any Model I/III DOS.**

After hours of painstaking programming and debugging, you've come up with the ultimate arcade game—The Flying Slim Whitman Records—where you try to blast away the disks as they speed past you; "Una Paloma Blanca" garners a 500-point bonus. But when you run the program, the records move more slowly than they did up the *Billboard* charts.

Your problem lies with the way Basic executes program instructions: The Basic interpreter breaks down program statements into machine-language commands before executing them. You can circumvent this time-consuming process by running your program through a Basic compiler, a program that converts Basic statements to machine code. Since the compiler does some of the work of the Basic interpreter, the resultant program runs faster.

I wrote an enhanced version of Dr. Dimitri P. Bertsekas's Basic compiler ("A Basic Compiler in Basic," October 1982, p. 122). My version, which I call FastBas, compiles more Basic commands, is easier to use, and runs under any DOS on the Models I and III. (For a comparison of FastBas with commercial compilers, see the sidebar.)

When FastBas converts Basic programs to machine language subroutines, programs run faster. Programs that use primarily integer arithmetic and graphics run 50–100 times faster, while programs written mostly in single-precision variables execute 3–20 times faster.

The programs you're compiling must stringently follow Basic programming rules for correct compilation. FastBas doesn't accept all Basic commands, but all compilable commands run in Basic. I suggest running and debugging your program in Basic, then compiling it.

Compiler Basics

To understand how Basic compilers work and how they increase execution speed, you must first understand how your TRS-80 handles programming functions.

The TRS-80 ROM contains a machine-language program that interprets Basic programs line by line. If you run a line 1,000 times during the course of a program, the machine-language program interprets the statements in that line 1,000 times.

The Basic interpreter calls specific machine-language subroutines in ROM for each statement. Because of the extensive error-checking that takes place during this process, the program runs slow compared to machine-language programs.

A compiler, on the other hand, looks at each Basic statement once. It interprets the statement, then writes a machine-language subroutine for it and stores that subroutine in memory. The compiler then interprets the next statement and puts the machine-language subroutine in memory after the previous subroutine. Once the compiler changes all Basic statements to machine-language subroutines, you can save and run the compiled program.

How much faster compiled programs run varies with the nature of the program. Compiled programs still contain many inefficiencies in coding relative to bona fide machine-language programs.

Program Changes

I rewrote as few lines of the original program as possible and didn't renum-

ber it so that readers who have keyed in the original can modify their programs.

Lines 0-500 in Program Listing 1 contain a sample Basic program for you to compile. Note that the last line compiled must be an End statement.

Many of FastBas's subroutines appear before the main program to speed up compilation. The main program starts at line 1000.

Most of the statements I added are string functions. FastBas can compile the Basic commands to input a string value, and then find a string's VARPTR, ASC, and VAL. The program also interprets string concatenation (addition), and redefines strings from other string values. It also accepts Dimension (DIM) and Clear statements (see Tables 1-4 for a complete list of compilable statements and functions).

Program Listing 2 uses the string Input function and the ASCII (ASC) function to let you answer yes/no questions. The ASCII function finds the ASCII value of the first character of the input string and acts on it using If statements.

This program also exemplifies the use of the Dimension and Clear functions. FastBas accepts only one Dimension statement in the first line of the program you're compiling. You can, however, precede Dimension with a Clear statement, as in line 10.

For best results, put Clear and Dimension on a line separate from other program statements. Also, you can't GOTO the line containing the Dimension statement from within a compiled program.

Listing 2 also shows you how to use the Dimension statement in both Basic and compiled programs. The Dimension statement sets up the variable storage table and tells you how many of each type of variable exist. In this sample, DIM A$(40) tells the compiler that the program uses one string, A$, and that the maximum string length is 40 characters. The equivalent statement in Basic (Run) sets up a string array with a dimension of 40.

Although Basic doesn't need a 40-string array, setting one up doesn't affect program execution, and you can test the Basic version before compiling it.

Program Listing 3 demonstrates how to use VARPTR (variable pointer) with PEEK and POKE commands to simulate uncompilable Length (LEN) and LEFT$ functions. By using substitute programming techniques, you can simulate other uncompilable functions, such as RIGHT$ and MID$.

The VARPTR function identifies the memory location of each string used. As in Basic, the VARPTR function for strings points to a 3-byte table that gives the length of the string in the first byte and the location of the string in the next 2 bytes.

No specific string variable table exists for FastBas, so VARPTR creates a table for the requested string. FastBas always stores the table below the variable buffer in memory.

VARPTR moves the correct string length and location values to the 3-byte table. Only one VARPTR location is active at a time. Once the program in Listing 3 obtains the VARPTR, it stores the values at the VARPTR location in other variables before seeking another string location.

The Clear command clears all variables to zero. It's not needed at the beginning of the program you're compiling, since FastBas automatically zeros the variable storage area.

If you don't want variables automatically zeroed, make the USR call to your compiled program 13 bytes higher than specified in the addresses given after compilation. This lets you compile several machine-language subroutines that the USR function calls from your Basic program. Each subroutine can then use the previously defined variables.

You can put the Clear command anywhere in the program you're compiling. FastBas ignores any digits after the Clear statement up to the colon or end of the program line (whichever comes first), but you can include them if they're necessary to run the uncompiled program. For instance, "Clear 200:" in the compiled program gives the same results as "Clear:" in the Basic version.

Compiling Basic Variables

FastBas accepts three variable types—integers, single-precision, and string variables—and you can't do much variable mixing. When using integer variables, the only way to include single-precision variables in the same statement is to use the integer (INT) function. FastBas allows only addition and subtraction with integer variables.

You must first define the values acted on within the functions PEEK, POKE, VARPTR, ASC, VAL, and so on as integer variables or integer numbers without using arithmetic operations. The values must be integer variables or integer numbers between −32767 and 32767 without arithmetic operations.

FastBas can perform close to a full set of mathematical operations with single-precision variables. This includes the four basic math functions plus exponents, logarithms, random numbers, trigonometric functions, and so on.

The hierarchy of FastBas's mathematical operations is not as sophisticated as in Basic. The order of a Basic mathematical evaluation is exponentiation, followed by multiplication and division, then addition and subtraction.

FastBas evaluates operations from right to left, using the order of parentheses. This simplified system speeds compilation, but requires careful use of

General Statements	Description
Clear	Integer value is optional
CLS	Clears screen
DIM	Same syntax as Basic except for strings (see text)
END	Mandatory statement to return to Basic from USR subroutine
GOSUB (line number)	
GOTO (line number)	
REM or '	
RETURN	

Table 1. General statements that can be compiled.

The Key Box

Models I and III
32K RAM Disk Basic
Cassette Basic

Basic Compiler Comparison

by Hardin Brothers

Compiler	Fast Bas	Accel ¾	Bascom	Vivace!	WIBASIC	ZBasic 3.0
Available for Models	I/III All DOSes.	I/III All DOSes.	III/4; only supported on TRSDOS but appears to work with other DOSes.	III/4 TRSDOS only.	I/III All DOSes.	I/III/4; all DOSes plus Apple, IBM-PC, Model 2000, Lobo & CP/M80.
Source File Location	Lines 1–500 of compiler program.	In memory with Accel and Disk Basic.	On disk as an ASCII file.	On disk as a tokenized file.	In memory with WIBASIC compiler and interpreter.	In memory with ZBasic compiler & editor or on disk in tokenized form.
Compile to	Memory only.	Memory only.	Disk /REL file, then linked into a CMD file.	Disk file.	Memory or disk.	Memory or disk.
Compatible with Disk Basic	Moderate. Supports integer and single-precision only. Several commands not supported.	Complete. Compiles only those commands it can optimize. Leaves others for Basic interpreter.	Very high, with a few minor differences.	Complete, though the Model III version had sporadic problems with file commands.	Uses its own version of integer Basic that is different from TRS-80 Basic, especially error-trapping and file-handling.	Uses its own version of Basic that is similar to Disk Basic, plus extra commands including structured loops (Repeat/Until & While/Wend) and graphics commands.
Location of runtime? library	All necessary code compiled into each command.	Separate disk file.	Separate disk file.	Separate disk file.	Compiled into CMD program.	Compiled into CMD program.
Chaining with common variables	No program chaining possible.	Chaining possible but no variables can be passed.	Chaining with any or all variables passed.	Model 4 only.	Possible, but requires tricky programming.	Can merge source programs, can't chain.
Speed of compiled program †	Moderately fast.	Moderately fast.	Very fast.	Relatively slow. Model 4 version was slower than original Basic program in benchmark test.	Very fast.	Fastest of all compilers in benchmark tests.
Ease of use	Moderately easy to use.	Easiest to use and understand.	Several options in both compiler and linker can cause confusion at first. Compiled program can be easily linked to Fortran and MACRO-80 programs.	Easy to use except I found the program sometimes had difficulty reading its own protected. disk.	Compiling is easy but programs can be difficult to develop because of differences between Basic and WIBASIC.	Easy to use and understand. Most differences between Basic and ZBasic are enhancements, and the syntax feels "normal."
Other Comments	Best choice for developing USR routines for your Basic programs.	Best feature is its ease of use. Requires both a runtime library and Disk Basic to run CMD program.	All math functions can be double-precision. Best choice for very large and complex programs that require chaining.	Supplied on protected disk. Slow execution speed of compiled programs. I found erratic bugs in both the compiler and completed programs.	Compiled programs are very fast. Requires learning a completely new version of Basic.	Speed of compiled programs and power of extensions to Basic make this a top choice for programs that do not require chaining. Source code compatibility with other computers may be a major asset for some.
Supplier & Cost	This issue of *80 Micro* & Load 80	Allen Gelder Software $99.95	Radio Shack $195	WittSoft, Inc. $49.95	WittSoft, Inc. $49.95	Simutek Computer Products $89.95

† Benchmark used for compilers except Fast Bas was a combination of all programs in "Marking Time" (May 1984, p. 100). Programs were run compiled and uncompiled under various DOSes and versions of Disk Basic.

parentheses for proper evaluation. The use of parentheses doesn't slow compiled programs.

The compiler accepts 26 integer variables, denoted A%–Z%, and 286 simple single-precision variables, denoted by A–Z, or by a letter followed by a single decimal digit. Possible simple single-precision variables are A, A0–A9, B, B0–B9,...Z, Z0–Z9.

Integer Statements	Description
+ and −	The only math operations allowed with integer variables. Example: A% = A% + B% − 7
For...To...Next	Requires integer variable or integer number. Follow Next with the specified integer. Example: FOR N% = A%TOB%:...:NEXTN% FOR N% = A% TO 10:...:NEXTN%
If...Then (Line number) Else	Any combination of =,<,> conditions are valid. Replace Then with GOTO or Then GOTO. Only specify line numbers after Then or GOTO. Any number of If...Then...Else statements can follow each other. Example: IFA%>=B%GOTO400 IFA%>=45GOTO400 ELSEIFA%<30GOTO400
Input	Don't include strings in quotation marks. Example: Input A%
INT	Changes single-precision expression to integer expression. Computed value of single-precision expression must be between −32767 and 32767. Example: A% = INT(R*256 − 65536)
Let	Optional. You can only define integer variables to an expression that includes single-precision when using INT. Example: LET A% = B% A% = B%
PEEK	Requires integer variable or integer number. Mathematical operations are not acceptable. Example: A% = PEEK(A%) A% = PEEK(−450)
Point	Same as PEEK.Integers must be between 0 and 127, and 0 and 47 respectively. Example: A% = POINT(C%,34) A% = POINT (23,34)
POKE	Same as PEEK. Second Integer must be between 0 and 255. Example: POKEA%,B% POKE −3445,B% POKE −3445,45
Print	You can mix integers, single precision, and strings, as well as messages between quotes. Example: PRINT"The answer is ";A% PRINTA%;B;" ";S$
PRINT@	Location is integer variable or integer number between 0 and 1023. Otherwise same as Print. Example: PRINT@C%,"The answer is ";A% PRINT@320,A%;B;" ";S$
Reset	Same as PEEK and Point. Example: RESET(C%,34) RESET(23,34)
Set	Same as PEEK. Example: SET(C%,34) SET(23,34)

Table 2. Integer statements that can be compiled. Integer statements except Let require definition to an integer variable and can't include math operations. They must be between −32767 and 32767.

You can represent one- and two-dimensional arrays using the letters A–Z. Two-dimensional arrays must be square. For example, array B with dimensions B(34,34) is acceptable, but B(34,35) is not.

Denote string variables using A$–Z$. The maximum length for each string is the same. The program uses the same maximum length for string literals in Print statements.

If FastBas doesn't find a Dimension statement in the first line of your program, it uses default values for all variables (see line 1010 of Listing 1). Regardless of the dimension, FastBas has a minimum of integer and single-precision values from A%–Z% and A, A0–Z, Z0, respectively.

The Dimension statement doesn't have to include all types of variables. FastBas zeros variables not included, in the case of arrays, or takes minimum values. The minimum simple single-precision variable value is a letter from A–Z.

The compiler sets strings A$–Z$ to a character length of 40 unless otherwise specified in the Dimension statement. I recommend that the string length remain at a minimum of 40 so you can use print statements in your program. You can specify a minimum of one string, A$, with the Dimension statement.

You can dimension more than one variable of each type in one Dimension statement. This is so the equivalent Basic program doesn't cause an error.

Only the last variable specification of each type sets the maximum dimension. For instance, Dimension C(20,20), B(2,2) is permissible, although in the compiled program, the maximum two-dimensional array is B(2,2).

The program hasn't saved enough space to store the C array. Changing the Dimension statement so it compiles to Dimension B(2,2),C(20,20) saves space for both arrays. This is important, since FastBas doesn't check for illegal arrays.

FastBas doesn't error-check variables, either. This can produce problems or work to your advantage. In the case of the defaulted variable values, the program has reserved memory for one-dimensional and two-dimensional arrays.

FastBas stores all arrays sequentially and computes the location of each array value. You'll probably dimension the maximum array location to 20, but you can define larger arrays.

You can include an array with a max-

HOWE SOFTWARE PRODUCT GUIDE

SYSTEM DIAGNOSTIC

Is your computer working correctly? **Are you sure?** Find out for certain with System Diagnostic, which has complete diagnostic tests for every component of your TRS-80™ Model 1, 3 or 4.

- **ROM:** checksum tests.
- **RAM:** three tests including every location and data value.
- **Video Display:** character generator, video RAM, video signal.
- **Keyboard:** every key contact tested.
- **Line Printer:** character tests with adjustable platen length.
- **Cassette Recorder:** read, write, verify data.
- **Disk Drives:** disk controller, drive select, track seek, read sectors, formatting, read/write/verify data with or without erasing, disk drive timer, disk head cleaner. Single or double density, 1-99 tracks.
- **RS-232-C Interface:** connector fault, data transmission, framing, data loop, baud rate generator.

Individual or continuous testing mode, reports optionally written to line printer. Separate versions for Models 1, 3, and 4. (Cassette and ROM tests not included in Model 4 version.)

System Diagnostic $99.95

SMALL BUSINESS MANAGEMENT SYSTEM

A complete system for using your computer to manage a small business.

- **Order Entry:** enter orders, print invoices, change past invoices. Use preprinted forms or plain paper.
- **Bookkeeping:** complete general ledger by user-defined categories. Print monthly or year-to-date summaries and ledgers.
- **Accounts Receivable:** show balances due from unpaid invoices at any time.
- **Accounts Payable:** show amounts you owe for unpaid bills in any category.
- **Inventory:** complete summary of products sold by part number and prices charged. Limited to 1,000 8-character part numbers (2,000 for MS-DOS).
- **Installation package** tailors the programs to your business.

Requires a minimum of two floppy disk drives or hard disk, 80-column printer. Available for: TRS-80™ Models I, III, 4, 2, 12, 16 (TRSDOS), or 2000, IBM-PC, and most CP/M computers.

Need more information? Ask for our detailed description of this program with sample forms and reports.

All versions $250.00

TYPITALL — *The* most powerful *TRS-80 Word Processor*

TYPITALL is a word processor which is upward compatible with SCRIPSIT™ for the TRS-80™ Models 1, 3 and 4. If you know SCRIPSIT™, you can start using TYPITALL immediately. Even if you have never used a word processor before, you will find TYPITALL easy to learn, and full of so many powerful features that you'll wonder why other word processors don't have them. Following is a list of some of TYPITALL's features. (If you need to know more, ask for our **free** catalogue.)

- Send **any** control or graphic/special character to the printer.
- Print the formatted text **on the screen** before going to the printer.
- Send the formatted text to a **disk file** for later printing.
- **Merge** data from a disk file during printing. Names, addresses, or any other text can be inserted during printing.
- Print during editing **(spooling).**
- Assign **any sequence of keystrokes** to a **single** control key.
- Call up to 16 **HELP screens** at any time.
- **Move cursor** forwards or backwards by character, word, line, or page.
- **Reenter** the program with **all text intact** if you accidentally exit without saving the text.
- **Spelling checker** with 29,500 word dictionary available.
- Verify 3,500 word document in **less than two minutes.**
- True Model 4 (**80 x 24 display, TRSDOS 6)** and Model I/III versions.

"If you like SCRIPSIT™, you'll probably love TYPITALL." **80 Micro.**

TYPITALL Demo Version $49.95
TYPITALL Complete Version $129.95
TYPITALL with Spelling Checker $179.95

HOME BUDGET and CHECKBOOK ANALYST

Combines the functions of a checkbook program with budget analysis, comparisons, summaries and projections.

First, it is a checking program, which can be used to maintain your checking account. Enter deposits, print check registers, compute your current balance.

More important, the program analyzes both your income and expenses and keeps track of your budget. While it comes with a classification for home expenses, these can easily be modified, so that program can be used for a business.

Income is classified in up to 32 categories. Expenses are broken into up to 12 main categories, each of which can be further divided into 12 subcategories. Checks can be divided into split transactions, and cash expenses can be included. (Only one checking account allowed.)

The program computes monthly and year-to-date income and expense summaries, and projections for the rest of the year based on data through a known month. Monthly expenses can be compared to a preestablished budget.

Model 1/3/4 disk version ... $59.95
Model 1/3/4 cassette version ... $29.95
Model 2/12/16 version $69.95
MS-DOS or CP/M version ... $69.95

SMALL BUSINESS ACCOUNTING

Based on Dome Bookkeeping Record #612, this program keeps track of income, expenses, and payroll for a small business.

Complete ledgers are maintained for both income and expenses. The program computes monthly, through last month, and year to date summaries.

The payroll section (not included in cassette version) contains provision for up to 99 employees with up to six categories of deductions. The program can print both checks and summaries of payroll for one employee or all employees.

Model 1/3/4 disk version ... $59.95
Model 1/3/4 cassette version $29.95
Model 2/12/16 version $69.95
MS-DOS or CP/M version ... $69.95

MONITOR #5

A comprehensive machine language monitor and debugger for the TRS-80 Models I, III, or 4.

Display memory in ASCII or hex. **Disassemble, move** and **compare** blocks. **Search, modify** memory. **Relocate** object programs. Read and write **cassettes** or **disk files.** Input and output of **disk sectors,** bypassing DOS. **RS-232-C commands** for terminal, send and receive. **Complete debugging package** includes **setting** and **displaying registers, single stepping** through machine instructions, setting **breakpoints,** and executing programs.

Monitor #5 .... $22.95

Howe Software

14 Lexington Rd., New City, NY 10956

Developing software for TRS-80 computers since 1979.

24-Hour TOLL-FREE Order Number:
Outside California call:
(800) 428-7825, ext. 169
Inside California call:
(800) 428-7824, ext. 169
Visa, Master Card, or COD orders only.

For information call:
(914) 634-1821

Add $3.00 postage & handling.
New York residents add sales tax.

*TRS-80 and SCRIPSIT are trademarks of Tandy Corp

imum dimension of A(60) in your program if you don't use arrays B and C. FastBas specifies the maximum letter for arrays as at least C, so dimensioning Z(20) accommodates a single array of A(520) just as easily as dimensioning A(520). An example of this appears in the sample program in Listing 1.

The added commands for string input and for string concatenation don't check for string length. However, the original string definition function does check, so the same rules apply as for arrays. If you expect a string to be twice as long as allocated due to these functions, don't use the next higher string variable.

Single-precision Commands	Description
+, −, *, /, and [(exponent)	Accepted math operations. Can include single-precision expressions with integer variables. Placement of parentheses is important. Example: A = A%*B − ((C/7)[2)
ABS	Can use single-precision expressions, single-precision variables, or integer variables in parentheses. Example: A = ABS(4*B% − 76)
ATN	Same as ABS. Example: A = ATN(4*B% − 76)
COS	Same as ABS. Example: A = COS(4*B% − 76)
EXP	Same as ABS. Example: A = EXP(4*B% − 76)
If...Then (Line number) Else	Any combination of =,<,> conditions are valid. Can replace Then with GOTO or Then GOTO. Only specify line numbers after Then or GOTO. Any number of If...Then...Else statements can follow each other. If first value is an integer variable, it must follow rules for integers. Example: IFA> = B%GOTO 400 IFA> = 45*3GOTO400ELSEIFA< CGOTO400IFA> = 45*3THEN400 ELSEIFA<CTHEN400
Input	Don't put quotation marks around string expressions in Input commands. Example: Input A
Let	Optional command. Can define single-precision variables to an expression that includes integer expressions. Don't include integer commands in the expression, only integer variables and single-precision variables and expressions. Placement of parentheses crucial for correct evaluation. Example: A = 50 + TAN(355/(113*4)) + (A%*256)
LOG	Same as ABS. Example: A = LOG(4*B% − 76)
Print	Same as integer Print. (Table 2)
PRINT@	Same as integer PRINT@. (Table 2)
RND(0)	Random number greater than 0 and less than 1. Example: A = RND(0)*10
SIN	Same as ABS. Example: A = SIN(4*B% − 76)
SQR	Same as ABS. Example: A = SQR(4*B% − 76)
TAN	Same as ABS. Example: A = TAN(4*B% − 76)

Table 3. Single-precision functions that can be compiled. These functions can use math operations.

Passing Variables

Two special functions, DEFUSR and USR, pass variables from Basic to a machine-language subroutine and back. You can use them to speed up a Basic program by having only part of it compile into a machine-language USR subroutine (see sample program in Listing 1).

A Basic program that calls the compiled subroutine starts at Line 10000. Line 10060 calls the machine-language subroutine compiled from lines 0–500. FastBas's DEFUSR function is in line 1310. Note that you must dimension the Basic portion of the program sufficiently when arrays pass back from the machine-language subroutine to Basic.

If the If statement condition has executed the Clear 50 command in line 1316, running the sample program (by pressing the enter key) results in the error message "Subscript out of range" in line 1340. To run the program, press the break key and type in RUN 10000.

Line 30 in the sample program instructs the compiler to write code to pass the variable N% to the USR subroutine using the statement N% = 0 + N%. Line 40 transfers the value of the A array in the same fashion.

After the USR subroutine performs its function, the redefined A array transfers back to Basic in line 140 by A(I%) = 1*A(I%).

The values of the variables on the right side of each equation are from lines 10000–10090 in the uncompiled Basic part of the program. They don't have to be the same variable name as on the left of the equals sign in lines 30, 40, and 140, but I recommend using identical variable names to assure complete compatibility with the uncompiled program. To try the program in Basic, change line 10060 to GOSUB30 and change line 500 to RETURN.

Manipulations in Memory

FastBas doesn't compile in a fixed memory location, but determines where protected memory begins and compiles from that point. If the program exceeds memory limits, it lowers the high memory pointer (HIMEM) by 512 bytes and restarts compilation. This process

String Commands	Description
ASC	ASCII value of a string. Example: A% = ASC(D$)
CHR$	Requires integer variable or integer number in parentheses. Example: B$ = CHR$(A%) + CHR$(65)
Input	Don't include string expressions in quotation marks. No error-checking for string length. Example: Input A$
Let	Optional command. Can define string variables as string expressions that include string concatenation and redefinition. No error-checking in string length occurs for concatenation and redefinition. Example: A$ = B$ A$ = "First String" + CHR$(C%) + "B$
Print	Same as integer Print. (Table 2)
PRINT@	Same as integer Print@. (Table 2)
VAL	Compiler truncates any decimal value to an integer. Example: A% = VAL(D$)
VARPTR	Example: A% = VARPTR(D$)

Table 4. String functions that can be compiled. You can define many string commands with integer variables.

Program Listing 1. FastBas and sample Basic compilation program.

```
0 ' *** PROGRAM LINES 10-500 AND 10000-10090 ARE A SAMPLE PROGRAM
READY FOR COMPILATION. RUN 1000 TO COMPILE.
10 ' *** NUMERICAL ARRAY SORTING (FROM LEVELII MANUAL)
20 ' *** PASS ARRAY A FROM BASIC TO USR ***
30 N%=0+N%
40 FOR I%=1 TO N%:I%=1*I%:A(I%)=0+A(I%):NEXT I%
50 ' *** SORTING PROGRAM BEGINS ***
60 D%=1
70 D%=D%+D%:IFD%<N%THEN70
80 D%=INT((D%-1)/2):IFD%=0 THEN 140
90 T%=N%-D%:FORI%=1TOT%:J%=I%
100 L%=J%+D%:IFA(L%)>A(J%)THEN 120
110 T=A(J%):A(J%)=A(L%):A(L%)=T:J%=J%-D%:IFJ%>0 THEN100
120 NEXTI%:GOTO80
130 ' *** PASS ARRAY A FROM USR TO BASIC ***
140 FOR I%=1 TO N%:I%=1*I%:A(I%)=1*A(I%):NEXT I%
500 END
501 '***************** BASIC COMPILER *********************
502 '** COPYRIGHT 1981 BY D. P. BERTSEKAS, BELMONT, MASS.
503 '** PERMISSION TO USE; NOT TO SELL
505 '** ENHANCEMENTS 1984 BY T. L. QUINDRY, BURKE, VA.
506 '**
510 '************** PEEK & POKE ROUTINES ******************
512 POKEM,P:PRINTP;:M=M+1:IFM<-12-MR*16384THENRETURNELSECLS:PRINT@
320,"PROGRAM HAS EXCEEDED PROTECTED MEMORY SIZE"
513 PRINT:PRINT"LOCATION TO COMPILE LOWERED BY 512 BYTES AND PROGR
AM BEING RERUN":POKE16562,PEEK(16562)-2:FORN=1TO2000:NEXT:GOTO1000
514 PC=PEEK(Q):PN=PEEK(Q+1):Q=Q+1:IFPC=32THEN514ELSEIFPC=0C=2:RETU
RNELSERETURN
518 IFPC<65ORPC>90THEN522ELSERETURN
520 IFPN<>37THEN522ELSEQ=Q+1:RETURN
522 PRINT:PRINT"ERROR LINE #";L1(L-1):END
524 PRINT@0,F$;:PRINT@64,CHR$(30):RETURN
529 ' ******** SINGLE PRECISION ASSIGNMENT ROUTINES ********
530 ' ROUTINE TO FIND VAR. ADDRESS PARAMETERS
535 GOSUB514:GOSUB518:V1=PC-65:IFPN<48ANDPN>57ANDPN<>213ANDPN<>40A
NDCF<>1THEN522
538 IFPN>47ANDPN<58MI=PN-47:GOSUB514ELSEMI=0
540 IFPN=213Z1=1:RETURN
```

Listing 1 continued

repeats until the program reaches a minimum HIMEM limit or until the program completely compiles.

The variable storage area is located above the compiled program and extends down from the top of memory as needed. All compiled subroutines use the same variable table area, so if you put more than one compiled program in memory, write the Dimension statement to accommodate the largest compiled subroutine.

If interference from the top of the compiled program and the allocated variable table area occurs during compilation, FastBas relocates the compiled program, corrects its jump addresses to accommodate the allocated area for variable storage, and corrects the HIMEM pointer.

Compilation can occur too low for the compiler to work. If the HIMEM pointer is too low in memory, FastBas adjusts it upwards and the program compiles at the higher memory position.

FastBas requires about 1,650 bytes above the Basic program to run. However, if FastBas relocates the HIMEM pointer to the minimum allowable value and you get an out-of-memory error, change the value + 1650 in line 1007 to a higher value.

Compilable Commands

Tables 1–4 list and define compilable Basic commands and explain how to use them. Table 1 lists general statements that compile. Table 2 lists the compilable integer statements. Table 3 lists compilable single-precision commands, and Table 4 contains compilable string commands.

Remember that functions that apply to integer variables give integer results, and the values that go within parentheses of the functions must be integer values or integer variables.

FastBas can't carry out arithmetic functions within parentheses when using the special integer functions, and has limited error-checking for this type of error. If FastBas finds an illegal operation, it ignores any remaining expressions on that program line.

Array locations must be integer variables. Functions that apply to single-precision variables can have single-precision variables, integer variables, or single-precision expressions within the parentheses of the function.

Using FastBas

To compile a Basic program, number the program lines from 0–500 and continue it from line 10000 if necessary. Save the program in ASCII, then merge it with FastBas. Delete the lines relating to the sample program, then type RUN 1000 to compile the program.

As an example, try compiling Pong in Program Listing 4. This two-player paddle game demonstrates how much faster compiled code runs compared to a Basic program.

Try running the Basic version first. Use the O and L keys to move the right paddle and the Q and A keys for the left paddle. The game stops when a player gets 16 points. The highest ball speed setting moves very slowly. The compiled program runs so fast that the highest speed setting is impossible.

After you've merged your Basic program and run FastBas, it displays each Basic line number and the compilation address (except for the line containing the Dimension statement), followed by all code in decimal. A colon separates each statement in the line.

By changing the Print statements in lines 512 and 1035 of Listing 1 to LPRINT statements, you can trace where each command compiles in memory.

FastBas's last display indicates the hexadecimal (hex) and decimal values for the start, end, and transfer address of the compiled program, along with the location of the compiled program and the variable table (see the Figure).

In this last screen you have all the information needed to save and run the compiled program or subroutine. Use the start, end, and transfer hex addresses with the Dump command to save the program to disk (see your DOS manual for correct syntax).

When entering Basic, you must protect high memory or your compiled program will crash. Use the decimal notation in the second row under the column labeled Start. When in Basic, define the USR function (DEFUSR) by using the third row value under Start. This is always a negative value.

To load the compiled program into memory, type in LOAD followed by a file name from DOS Ready and hit the enter key. If you're using another DOS, use the equivalent statement. You can save the program using the CMD "L" command from Basic. Be sure to in-

```
COMPILED PROGRAM CONFLICTS WITH VARIABLE STORAGE AREA.
RELOCATING PROGRAM . . .
 33  252  249  17  252  247  1  167  5  237  176  201
PROTECTED MEMORY POINTER CHANGED AND PROGRAM RELOCATED TO

                START           END             TRANSFER
HEXIDECIMAL     &HF7FC          &HFDA3          &HF7FC
DECIMAL         63484           64931           63484
DECIMAL         -2052           -605            -2052

VARIABLES STORED - &HFE3F
TO TOP OF MEMORY - &HFFFF


<ENTER> TO RUN MACHINE CODE . . .?
```

Figure. Final display after compiling the Pong program.

Listing 1 continued

```
545 IFPN=40GOSUB514:GOSUB514:GOSUB518:V2=PC-65:GOSUB520:GOSUB514EL
SEIFCF=1Z1=1:RETURN
550 IFPC<>41ANDPC<>44THEN522
555 IFPC=41Z1=2:RETURN
560 IFPC=44GOSUB514:GOSUB518:V3=PC-65:IFV1<0ORV1>=NTTHEN522ELSEGOS
UB520:Z1=3:GOSUB514:IFPC<>41OR(PN<>213ANDCF<>1)THEN522ELSERETURN
564 ' ADDRESS COMPUTATION ROUTINE
565 ONZ1GOSUB570,575,580:RETURN
570 IFMI>ISTHEN522ELSEC1=VF+(V1+MI*26)*4:GOSUB836:GOSUB902:RETURN
575 V7=V1:V8=V2:GOSUB610:RETURN
580 V7=V1:V8=V2:V9=V3:GOSUB620:RETURN
600 ' ARRAY PORTION OF ADDRESS ROUTINE
610 V0=V8:GOSUB912:GOSUB906:C1=VA+V7*DO*4:GOSUB836:GOSUB900:GOSUB9
04:RETURN:' 1-D ARRAY
620 V0=V9:GOSUB912:P=41:GOSUB512:C1=VD+4*NT*DT*DT+2*V7*DT:GOSUB836
:GOSUB900:GOSUB904:P=94:GOSUB512:P=35:GOSUB512:P=86:GOSUB512:V0=V8
:GOSUB912:GOSUB906:GOSUB904:RETURN:' 2-D ARRAY
699 ' ROUTINE TO EVALUATE SINGLE PRECISION EXPRESSIONS
700 GOSUB514:IFPC<>206GOTO704
702 E1=0:D1=0:GOSUB902:GOSUB926:GOSUB926:GOTO714:'TAKES CARE OF LE
ADING <-> SIGN
704 GOSUB730
706 GOSUB514:IFC=2ORPC=41ORPC=58ORPC=59ORPC=212ORPC=213ORPC=214ORP
C=141ORPC=202THENRETURN:'PEEK NEXT BYTE; IF TERMINATOR RETURN
708 GOSUB934:'MOVE INTERIM RESULT FROM 4121H STORAGE AREA TO STACK
710 ' MOVE NEW VARIABLE TO 4121H AREA; POP BCDE; OPERATE
712 IFPC=205GOSUB514:GOSUB730:GOSUB936:GOSUB940:GOTO706:'ADD
714 IFPC=206GOSUB514:GOSUB730:GOSUB936:GOSUB942:GOTO706:'SUBTRACT
716 IFPC=207GOSUB514:GOSUB730:GOSUB936:GOSUB944:GOTO706:'MULTIPLY
718 IFPC=208GOSUB514:GOSUB730:GOSUB936:GOSUB946:GOTO706:'DIVIDE
720 IFPC=209GOSUB514:GOSUB730:GOSUB936:GOSUB948:GOTO706:'EXPONENTI
ATE
722 GOTO522:'ERROR TRAP
729 ' ROUTINE TO EVALUATE CONSTANTS, VARIABLES & FUNCTIONS IN SING
LE PRECISION EXPRESSION & MOVE THEM TO 4121H STORAGE AREA
730 IF(PC<58ANDPC>47)ORPC=46GOSUB845:RETURN:'CONVERT CONSTANT TO 4
-BYTE REPRESENTATION; MOVE IT & RETURN
732 IFPC=222GOSUB514:IFPC<>40THEN522ELSEGOSUB514:IFPC<>48THEN522EL
SEGOSUB514:IFPC<>41THEN522ELSEP=205:GOSUB512:P=240:GOSUB512:P=20:G
OSUB512:RETURN:'RND(0)
734 IF(PC>220ANDPC<229)ORPC=217ORPC=40THEN756ELSEIFPC<65ORPC>90THE
N522:'IF FUNCTION GOTO 758
736 V4=PC-65:IFPN>47ANDPN<58THENME=PN-47:GOSUB514:Z2=1:GOTO746ELSE
IFPN<>40ANDPN<>37THENME=0:Z2=1:GOTO746ELSEIFPN=37THENZ2=4:GOSUB514
:GOTO746
738 GOSUB514:GOSUB514:GOSUB518:V5=PC-65:GOSUB520:GOSUB514
740 IFPC<>41ANDPC<>44THEN522
742 IFPC=41Z2=2:'1-D ARRAY
744 IFPC=44GOSUB514:GOSUB518:V6=PC-65:GOSUB520:Z2=3:GOSUB514:IFPC<
>41THEN522:'2-D ARRAY
746 ONZ2GOTO748,750,752,754
748 IFME>ISTHEN522ELSEC1=VF+(V4+ME*26)*4:GOSUB836:GOSUB902:GOSUB93
2:RETURN
750 V7=V4:V8=V5:GOSUB610:GOSUB932:RETURN
752 V7=V4:V8=V5:V9=V6:GOSUB620:GOSUB932:RETURN
754 V0=V4:GOSUB912:P=34:GOSUB512:P=33:GOSUB512:P=65:GOSUB512:P=205
```

Listing 1 continued

clude the /CMD extension in the filename if you save the program this way.

Model I TRSDOS doesn't have an equivalent command to load from Basic; load from DOS instead. If you use Model III TRSDOS and the Dump command, return to DOS by rebooting instead of using CMD "S".

If the compiled program contains an INPUT$ function and the string is more than four characters long, calling the routine from the Basic command mode instead of from a Basic program results in a syntax error on return to Basic. However, this doesn't affect operation of the compiled program.

If you compiled Pong, the final screen should look like that in the Figure. Assuming you're using TRSDOS 1.3, you then reboot the computer and type in DUMP PONG (START = 0F7FC, END = 0FDA3, TRA = 0F7FC) and hit the enter key to save the compiled program.

To load Pong, type in LOAD PONG/CMD from DOS, hit the enter key, type in BASIC – M:63484, hit the enter key, and type in DEFUSR = – 2052:A = USR(0) and hit the enter key. The compiled program starts running immediately.

Troubleshooting

FastBas checks for some syntax errors during compilation. However, other errors go undetected until you try to run the program.

If the compiled program runs but gives different results than its Basic equivalent, check arithmetic operations. The compiler and the interpreter carry out the order of arithmetic operations in a single-precision variable differently. Check also if a machine-code variable appears in the right side of an assignment statement before it's initialized.

If the compiled program crashes, check the For...Next loops; make sure they're set up properly and that the program doesn't jump out of a loop.

Modifications

Analyzing the code in memory with a disassembler provides many clues to machine-language programming. For example, you can examine single-precision math routines and write similar code in your programs using the compiled routines as a guide.

All essential statements used in writing FastBas are in the list of compilable statements, so it's possible to rewrite the entire compiler and structure it so the program can compile itself.

However, the size of the Basic program that you must write, the number of GOTO and GOSUB statements, and the size of the resulting code might make it impractical.

Also, you'd have to change many of the variables to single-precision to have the program compile itself, so an increase in speed of more than three times is unlikely.

It's an intriguing project, however, and I'll leave that as a challenge to anyone who cares to try. ■

Listing 1 continued

```
:GOSUB512:P=204:GOSUB512:P=10:GOSUB512:RETURN:' CONVERT INTEGER VA
R. TO SINGLE PRECISION
756 IFPC=40THEN776:'PARENTHESIS
758 Q=Q+1
760 IFPC=221THEN778:'SQR
762 IFPC=217THEN780:'ABS
764 IFPC=223THEN782:'LOG
766 IFPC=224THEN784:'EXP
768 IFPC=225THEN786:'COS
770 IFPC=226THEN788:'SIN
772 IFPC=227THEN790:'TAN
774 IFPC=228THEN792:'ATN
776 GOSUB700:GOTO794
778 GOSUB700:GOSUB952:GOTO794
780 GOSUB700:GOSUB954:GOTO794
782 GOSUB700:GOSUB956:GOTO794
784 GOSUB700:GOSUB958:GOTO794
786 GOSUB700:GOSUB960:GOTO794
788 GOSUB700:GOSUB962:GOTO794
790 GOSUB700:GOSUB964:GOTO794
792 GOSUB700:GOSUB966
794 IFPC<>41THEN522
796 RETURN
800 ' *************** CONVERSION ROUTINES ***********
809 ' ROUTINE TO FIND LSB & MSB OF INTEGER NUMERIC STRING
810 C$="":IFPC=206THENPC=45:GOSUB818ELSEGOSUB816:IFC$=""THENC1=-1:
RETURN
812 C1=VAL(C$)
814 D1=C1/256:E1=C1-D1*256:IFC1<0THEND1=D1+256:C1=-C1:RETURN:ELSER
ETURN
816 IFPC<48ORPC>57THENRETURN
818 C$=C$+CHR$(PC):GOSUB514:GOTO816
835 ' ROUTINES TO FIND LSB & MSB OF ADDRESSES ABOVE 1ST 32K
836 D1=C1/256:E1=C1-D1*256:D1=D1+256:RETURN
840 Z=VT+V1+V1:P1=Z/256:P=Z-P1*256:P1=P1+256:RETURN
844 ' ROUTINE TO CONVERT NUMERIC STRING TO 4-BYTE SINGLE PRECISION
 REPRESENTATION
845 C$=CHR$(PC)
846 GOSUB514:IF(PC<58ANDPC>47)ORPC=46C$=C$+CHR$(PC):GOTO846
847 R=VAL(C$):GOSUB848:E1=33:D1=65:GOSUB902:C1=B3:GOSUB910:P=35:GO
SUB512:C1=B2:GOSUB910:P=35:GOSUB512:C1=B1:GOSUB910:P=35:GOSUB512:C
1=BE:GOSUB910:Q=Q-1:RETURN
848 IFR=0THENBE=0:B1=0:B2=0:B3=0:RETURN
849 Y1=1:Y2=2:N=1:IFY1>RTHEN852
850 IFY2<=RTHENY1=Y1+Y1:Y2=Y2+Y2:N=N+1:GOTO850
851 GOTO853
852 IFR<Y1THENY1=Y1/2:Y2=Y2/2:N=N-1:GOTO852
853 BE=N+128:X1=0:R=R-Y1:GOSUB856:B1=B
854 GOSUB857:X1=X:GOSUB856:B2=B
855 GOSUB857:X1=X:GOSUB856:B3=B:RETURN
856 GOSUB857:X2=X:GOSUB857:X3=X:GOSUB857:X4=X:GOSUB857:X5=X:GOSUB8
57:X6=X:GOSUB857:X7=X:GOSUB857:X8=X:B=X1+X1+X2:B=B+B+X3:B=B+B+X4:B
=B+B+X5:B=B+B+X6:B=B+B+X7:B=B+B+X8:RETURN
857 Y1=Y1/2:RT=R-Y1:IFRT<0X=0:RETURN:ELSEX=1:R=RT:RETURN
879 ' ROUTINE TO POKE STRING IN TEMPORARY STORAGE AREA
880 C1=MF:GOSUB836:GOSUB900:NN=1
881 IFPC=34GOSUB890:RETURN
882 GOSUB889
883 IFFP=1ANDPEEK(Q)=32Q=Q+1:PC=32:NN=NN+1:GOTO885
884 GOSUB514:NN=NN+1
885 IFFP=1AND(PC=34ORC=2)GOSUB890:RETURN
886 IFFP=0AND(PC=58ORC=2)GOSUB890:RETURN
888 GOTO882
889 P=62:GOSUB512:P=PC:GOSUB512:P=18:GOSUB512:IFPC<>0THENP=19:GOSU
```

Listing 1 continued

Thomas L. Quindry is an Associate Editor of 80 Micro. *Contact him at TLQ Enterprises, 6237 Windward Drive, Burke, VA 22015.*

TRS-80 COMPUTERS

MODEL IV 64K	$959.00	MODEL 2000	$2060.00
2 DISKS RS 232		w/HD	3360.00
		COLOR COMPUTER	
MODEL IV P	850.00	64K	170.00
MODEL 12	2999.00	MODEL 100 8K	350.00
ONE DISK	2339.00	MODEL 100 24K	500.00

Tandy 1200 HD IBM compatible $2599
Terminal OT100 $699

DISK DRIVES

15 MEG HD	$2099.00
COLOR DISK 0	275.00
COLOR DISK 1	229.00

PRINTERS

ALL OKIDATA	**CALL**
GEMINI 10X	270.00
GEMINI 15X	380.00

DMP-110	$ 299.00	DMP-120	$ 379.00
DMP-105	159.00	DMP-430	699.00
DMP-500	1064.00	DMP-2100	1598.00
DMP-420	680.00	DWP-210	459.00
DWP-510	1149.00	CMP-2150	3135.00

Langley St. Clair Green and Amber Screen $89.00

BODEX CORP.
224 E. MAIN ST.
MARLBORO, MASS. 01752
Phone (617) 485-5115 or 481-1029

TRS-80 Trademark of Tandy Corp. ✓381

Multi-User Manufacturing Control Software

(XENIX$_{TM}$ OPERATING SYSTEM)

AT LAST

A system designed specifically for manufacturing

- INVENTORY CONTROL
- MATERIAL REQUIREMENTS PLANNING
- CUSTOMER ORDER ENTRY
- ACCOUNTING

Features: —Menu Driven—System Security
—User Designed Data Base
—Customization Available

For More Information, Call or Write: ***Software Concepts***
5786 Annarue Place
Columbus, Ohio 43229
(614) 895-7196

✓192

XENIX$_{TM}$ IS A TRADEMARK OF MICROSOFT CORPORATION

EXPAND TRS-80 MEMORY

TRS-80 to 16K, 32K, or 48K

**Model 1 = From 4K-16K Requires (1) One Kit
Model 3 = From 4K-48K Requires (3) Three Kits
Color = From 4K-16K Requires (1) Kit

****Model 1 equipped with Expansion Board up to 48K Two Kits Required – One Kit Required for each 16K of Expansion –**

TRS-16K3 200ns for Color & Model III. . . . $8.95
TRS-16K4 250ns for Model I. $6.95

TRS-80 Color 32K or 64K Conversion Kit

Easy to install kits come complete with 8 ea. 4164-2 (200ns) 64K dynamic RAMs and conversion documentation. Converts TRS-80 color computers with D, E, ET, F and NC circuit boards to 32K. Also converts TRS-80 color computer II to 64K. Flex DOS or OS-9 required to utilize full 64K RAM on all computers.

TRS-64K2. . . . . $38.95

NEW! TRS-80 Model IV 64K or 128K Conversion Kit

Easy to install kit comes complete with 8 each 4164 (200ns) 64K dynamic RAMs and conversion documentation. Converts TRS-80 Model IV computers from 16K to 64K. Also expands Model 4P from 64K to 128K.

TRS-64K2. . . . . $38.95
(Converts the Model IV from 16K to 64K or will expand the Model 4P from 64K to 128K)

TRS-64K2PAL (Model IV only). $59.95
(8-4164's with PAL Chip to expand from 64K to 128K)

DIP/IC INSERTER/EXTRACTOR TOOLS

MOS-1416 (Inserts all above RAM pkgs. 14/16 pin). . . . **$10.95**
EX-1 (Extracts 8-24 pin pkgs.). . . . . **$ 2.49**

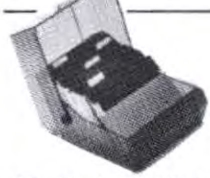
5¼" Diskettes

SSDD = Single Sided Double Density
DSDD = Double Sided Double Density

ULTRA MAGNETICS

Part No.	Description	Boxed	Price
UM51401	SSDD with Hub Ring	10	$24.95
UM52401	DSDD with Hub Ring	10	$33.95

All diskettes are soft-sectored and have hub rings.
Bulk prices available on request.



DISK MINDER

• Attractive, functional disk storage system • 36 (3", 3.25" and 3.5"), 50 (8") or 75 (5¼") disk storage capacity • Easy filing and retrieving • Protects disk from dust contamination • Molded from durable smoked plastic with front carrying handle • Size (DM36): 8⅝"L x 5½"W x 4⅞"D • Size (DM50 & DM75): 7"W x 6½"H x 9¼"D • Weight: 2 lbs.

Part No.	Description	Price
DM36	Stores 36 (3", 3.25" and 3.5") Diskettes. . . .	**$19.95 ea.**
DM75	Stores 75 (5¼") Diskettes.	**$19.95 ea.**
DM50	Stores 50 (8") Diskettes.	**$29.95 ea.**

We also specialize in integrated circuits, custom cables, power supplies, keyboards, and much, much more! Give us a call today!

INDUSTRIES ***Protect Yourself...***

DATASHIELD® Surge Protector

Eliminates voltage spikes and EMI-RFI noise before it can damage your equipment or cause data loss. Six month warranty.

Model 100. . . . . $69.95

DATA SHIELD® Back-Up Power Source also available.
Protect your TRS-80 Model computers.

PC200 (200 Watt Rating). $299.95
XT300 (300 Watt Rating). $399.95

$10.00 Minimum Order – U.S. Funds Only
California Residents Add 6½% Sales Tax
Shipping – Add 5% plus $1.50 Insurance
Send S.A.S.E. for Monthly Sales Flyer!

Spec Sheets – 30¢ each
Send $1.00 Postage for your
FREE 1985 JAMECO CATALOG
Prices Subject to Change

MasterCard **Jameco ELECTRONICS** VISA

1355 SHOREWAY ROAD, BELMONT, CA 94002 ✓534
Phone Orders Welcome (415) 592-8097 Telex: 176043

✓ See List of Advertisers on Page 145

Listing 1 continued

```
B512:RETURN:ELSERETURN
890 Q=Q-1:C=Ø:PC=Ø:GOSUB889:IFNN>SLTHENPRINT:PRINT"STRING TOO LONG
":GOTO522:ELSERETURN
899 ' *********** FREQUENTLY USED MACHINE CODES ************
900 P=17:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512:RETURN:'LD DE,E1D1
902 P=33:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512:RETURN:'LD HL,E1D1
904 P=25:GOSUB512:RETURN:'ADD HL,DE
906 P=41:GOSUB512:P=41:GOSUB512:RETURN:' ADD HL,HL; ADD HL,HL
908 P=235:GOSUB512:RETURN:' EXC HL,DE
910 P=54:GOSUB512:P=C1:GOSUB512:RETURN:'LD (HL),C1
912 C1=VT+VØ+VØ:GOSUB836:P=42:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512
:RETURN:' LD HL,(C1)
914 P=42:GOSUB512:GOSUB840:GOSUB512:P=P1:GOSUB512:RETURN:'LD HL,(P
1P)
916 P=34:GOSUB512:GOSUB840:GOSUB512:P=P1:GOSUB512:RETURN:'LD (PP1)
,HL
918 P=195:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512:RETURN:'JP E1D1
920 P=183:GOSUB512:P=237:GOSUB512:P=82:GOSUB512:RETURN:'OR A; SBC
HL,DE
922 P=40:GOSUB512:P=3:GOSUB512:RETURN:'JR Z,3
924 P=225:GOSUB512:RETURN:'POP HL
926 P=229:GOSUB512:RETURN:'PUSH HL
928 P=209:GOSUB512:RETURN:'POP DE
930 P=213:GOSUB512:RETURN:'PUSH DE
932 P=205:GOSUB512:P=177:GOSUB512:P=9:GOSUB512:RETURN:'MOVE VARIAB
LE TO 4121H AREA
934 P=205:GOSUB512:P=164:GOSUB512:P=9:GOSUB512:RETURN:'MOVE FROM 4
121H TO STACK
936 P=193:GOSUB512:GOSUB928:RETURN:'POP BC & DE
938 ' ARITHMETIC OPERATION & FUNCTION ROUTINES
940 P=205:GOSUB512:P=22:GOSUB512:P=7:GOSUB512:RETURN
942 P=205:GOSUB512:P=19:GOSUB512:P=7:GOSUB512:RETURN
944 P=205:GOSUB512:P=71:GOSUB512:P=8:GOSUB512:RETURN
946 P=205:GOSUB512:P=162:GOSUB512:P=8:GOSUB512:RETURN
948 P=205:GOSUB512:P=247:GOSUB512:P=19:GOSUB512:RETURN
950 P=205:GOSUB512:P=12:GOSUB512::P=10:GOSUB512:RETURN
952 P=205:GOSUB512:P=231:GOSUB512:P=19:GOSUB512:RETURN
954 P=205:GOSUB512:P=239:GOSUB512:P=10:GOSUB512:P=205:GOSUB512:P=1
19:GOSUB512:P=9:GOSUB512:RETURN
956 P=205:GOSUB512:P=9:GOSUB512:P=8:GOSUB512:RETURN
958 P=205:GOSUB512:P=57:GOSUB512:P=20:GOSUB512:RETURN
960 P=205:GOSUB512:P=65:GOSUB512:P=21:GOSUB512:RETURN
962 P=205:GOSUB512:P=71:GOSUB512:P=21:GOSUB512:RETURN
964 P=205:GOSUB512:P=168:GOSUB512:P=21:GOSUB512:RETURN
966 P=205:GOSUB512:P=189:GOSUB512:P=21:GOSUB512:RETURN
968 P=205:GOSUB512:P=167:GOSUB512:P=40:GOSUB512:RETURN
970 P=62:GOSUB512:P=4:GOSUB512:P=50:GOSUB512:P=175:GOSUB512:P=64:G
OSUB512:RETURN
972 P=205:GOSUB512:P=203:GOSUB512:P=9:GOSUB512:RETURN
999 ' ***************** MAIN PROGRAM **********************
1000 CLEAR200:DEFINTA-P,S-X,Z:DIML1(100),L2(100),A(50),A1(50),D(25
),E(25):F$="BASIC COMPILER"+STRING$(50," "):T$=CHR$(32)+CHR$(58)+C
HR$(32):CLS
1005 Q=PEEK(16548)+256*PEEK(16549):L=1:K=0:FP=0:CF=0:MC=PEEK(16561
)+PEEK(16562)*256+3-65536:M=MC:MR=0'IF YOU HAVE A 32K SYSTEM, MAKE
 MR=1; ELSE IF YOU HAVE A 48K SYSTEM, MR=0
1007 FM=PEEK(16633)+PEEK(16634)*256+1650-65536:IFFM>MCTHENC1=FM-3:
GOSUB814:PRINT@512,"LOCATION TO COMPILE CHANGED TO PROTECT BASIC P
ROGRAM":FORM1=1TO2000:NEXT:POKE16561,E1:POKE16562,D1:GOTO1000
1010 IS=10:DO=20:DT=20:SL=40:NO=26:NT=2:NS=26:'IS=# OF ADDITIONAL
S.P. VARIABLES PER LETTER; DO=DIM OF 1-D ARRAYS; DT=DIM OF 2-D ARR
AYS; SL=LENGTH OF STRINGS; NO=# OF 1-D ARRAYS ALLOWED; NT=# OF 2-D
 ARRAYS ALLOWED; NS=# OF STRINGS ALLOWED
1011 Q1=Q:Q=Q+4:GOSUB514:IFPC=184THENGOSUB7100'CLEAR
1012 IFPC=138THENGOSUB7110:IFIS=0THENIS=1'DIM
1013 Q=Q1:PRINT@448,"IS ="IS:PRINT"DO ="DO:PRINT"DT ="DT:PRINT"SL
="SL:PRINT"NO ="NO:PRINT"NT ="NT:PRINT"NS ="NS
1015 VT=-2*26-MR*16384:VF=-4*26*(1+IS)+VT:VA=-4*NO*DO+VF:VD=-4*NT*
DT*DT-2*NT*DT+VA:VS=-NS*(SL+1)+VD:VN=-(SL+1)+VS:PRINT@856,"ZERO VA
RIABLES":GOSUB7136'CLEAR
1020 ' MC=END OF VARIABLE STORAGE AREA AND ALSO THE START OF MACHI
NE CODE; VT=START OF INTEGER STORAGE; VF=START OF SIMPLE VARIABLE
STORAGE; VA=START OF 1-D ARRAY STORAGE; VD=START OF 2-D ARRAY STOR
AGE; VS=START OF STRING STORAGE; VN=TEMP. STORAGE
1021 IFNT>0GOSUB7000:' GENERATE CODE TO STORE 2-D ARRAY ADDRESSES
1025 M1=PEEK(Q)+PEEK(Q+1)*256:L1(L)=PEEK(Q+2)+PEEK(Q+3)*256
1030 GOSUB524
1035 PRINT:PRINT@960," LINE #"L1(L);M;":";:L2(L)=M:L=L+1:Q=Q+4
1040 IFL1(L-1)>500THEN1210
1045 C=0:GOSUB514:IFC=2THEN1140
1050 IFPC=133ORPC=134ORPC=136OR(PC>137ANDPC<140)ORPC=142ORPC=144OR
PC=148OR(PC>149ANDPC<177)OR(PC>178ANDPC<184)OR(PC>184ANDPC<189)OR(
PC>189ANDPC<202)ORPC=203ORPC=204ORPC=210ORPC=211THEN522:'ERROR TRA
P
1055 IFPC=215OR(PC>216ANDPC<221)OR(PC>228ANDPC<251)THEN522:'ERROR
TRAP
1056 IFPC=184GOSUB7135'CLEAR
1057 IFPC=140GOSUB514:' LET
```

Listing 1 continued

Listing 1 continued

```
1060 IFPC>64ANDPC<91ANDPN=37Q=Q-1:GOSUB2000:' INTEGER LET
1065 ' SINGLE PRECISION LET
1070 IFPC>64ANDPC<91ANDPN<>37ANDPN<>36Q=Q-1:GOSUB535:GOSUB514:GOSU
B514:IF(PC=49ANDPN=207)OR(PC=48ANDPN=205)GOSUB4000:ELSEQ=Q-2:GOSUB
514:IFPC<>213THEN522ELSEGOSUB700:GOSUB565:GOSUB972
1075 IFPC>64ANDPC<91ANDPN=36Q=Q-1:GOSUB4500:' STRING LET
1080 IFPC=178GOSUB2500:' PRINT
1085 IFPC=141GOSUB3500:' GOTO
1090 IFPC=143GOSUB3000:' IF ... THEN
1095 IFPC=145GOSUB3700:' GOSUB
1100 IFPC=146GOSUB3800:' RETURN
1105 IFPC=132THENP=205:GOSUB512:P=201:GOSUB512:P=1:GOSUB512:GOSUB5
14:IFPC<>58ANDC<>2THEN522:' CLS
1110 IFPC=137GOSUB2700:'INPUT
1115 IFPC=129GOSUB5000:' FOR
1120 IFPC=135GOSUB5500:' NEXT
1125 IFPC=130ORPC=131GOSUB6000:' SET & RESET
1127 IFPC=177GOSUB6500:' POKE
1130 IFPC=128P=205:GOSUB512:P=157:GOSUB512:P=10:GOSUB512:P=201:GOS
UB512:' END
1135 IFPEEK(Q-1)=58ORPEEK(Q-1)=149PRINTT$;:GOTO1045:' TEST FOR TER
MINATOR & ELSE TOKEN
1140 Q=M1:PRINT:GOTO1025:' START NEW LINE
1200 ' ********* ROUTINE TO ADJUST THE MACHINE CODE JUMPS ******
1210 GOSUB524:IFK=0THEN1250ELSEPRINT@960,"ADJUSTING JUMP ADDRESSES
 . . .";
1220 FORI=1TOK:DN=PEEK(A(I))+256*PEEK(A(I)+1):DH=0
1230 FORJ=1TOL:IFDN=L1(J)THENDH=L2(J):PRINTL1(J);
1240 NEXTJ:C1=DH:GOSUB836:POKEA(I),E1:POKEA(I)+1,D1:NEXTI
1250 CLS:IFVN-3>MTHEN1310ELSETR=M-VN+3:MO=TR/256:IFTR<>MO*256THENM
O=MO+1
1260 PRINT:PRINT"COMPILED PROGRAM CONFLICTS WITH VARIABLE STORAGE
AREA."
1280 IFMC-MO*256>FMTHENGOTO1286ELSEPRINT:PRINT"TO RECOMPILE PROGRA
M IN LOWER MEMORY WOULD OVERWRITE YOUR":PRINT"BASIC PROGRAM. TRY T
O REDUCE THE VARIABLE STORAGE AREA BY":PRINT"REDIMENSIONING VARIAB
LES. PROTECTED MEMORY HAS BEEN MOVED TO"
1285 PRINT"THE MINIMUM VALUE TO ALLOW THE BASIC COMPILER TO RUN.":
C1=FM-3:GOSUB814:POKE16561,E1:POKE16562,D1:END
1286 PRINT"RELOCATING PROGRAM . . .":M2=M:P=33:GOSUB512:C1=MC:GOSU
B814:P=E1:GOSUB512:P=D1:GOSUB512:P=17:GOSUB512:P=E1:GOSUB512:P=D1-
MO:GOSUB512:P=1:GOSUB512:C1=M2-MC:GOSUB814:P=E1:GOSUB512:P=D1:GOSU
B512
1287 P=237:GOSUB512:P=176:GOSUB512:P=201:GOSUB512:DEFUSR1=M2:MC=MC
-MO*256:M=M-MO*256-12
1288 IFK1>0THENFORI=1TOK1:POKEA1(I),PEEK(A1(I))-MO:NEXTI
1289 IFK>0THENFORI=1TOK:POKEA(I)+1,PEEK(A(I)+1)-MO:NEXTI
1290 PRINT:PRINT"PROTECTED MEMORY POINTER CHANGED AND PROGRAM RELO
CATED TO":GOTO1310
1299 ' ****** ROUTINE TO EXECUTE THE MACHINE CODE DIRECTLY *******
*
1300 GOSUB814:E2=E1/16:D2=D1/16:E1=E1-E2*16:D1=D1-D2*16:IFD2>9THEN
D2=D2+55ELSED2=D2+48
1301 IFD1>9THEND1=D1+55ELSED1=D1+48
1302 IFE2>9THENE2=E2+55ELSEE2=E2+48
1303 IFE1>9THENE1=E1+55ELSEE1=E1+48
1304 B$="&H"+CHR$(D2)+CHR$(D1)+CHR$(E2)+CHR$(E1):PRINTB$,:RETURN'C
ONVERSION TO HEX
1310 DEFUSR0=MC:PRINT@384,,"START","END","TRANSFER":PRINT"HEXIDECI
MAL",:C1=MC:GOSUB1300:C1=M:GOSUB1300:C1=MC:GOSUB1300:PRINT"DECIMAL
"TAB(15)65536+MCTAB(31)65536+MTAB(47)65536+MC:PRINT"DECIMAL",MC,M,
MC:PRINT
1315 PRINT"VARIABLES STORED - ";:C1=VN-3:GOSUB1300:PRINT:PRINT"TO
TOP OF MEMORY - ";:C1=-MR*16384-1:GOSUB1300:PRINT
1316 IFTR<>0THENPOKE16562,PEEK(16562)-MO:CLEAR50:X=USR1(0)
1320 PRINT@960,"<ENTER> TO RUN MACHINE CODE . . .";
1330 INPUTA$
1340 CLS:X=USR(0):END
1400 ' ********** END OF MAIN PROGRAM **************
1999 ' INTEGER ASSIGNMENT ROUTINE
2000 GOSUB514:GOSUB518:GOSUB520:V1=PC-65:GOSUB514:IFPC<>213THEN522
2010 GOSUB514:IF(PC=49ANDPN=207)OR(PC=48ANDPN=205)THEN2400
2011 IFPC=246THENGOSUB2017:P=126:GOSUB512:GOTO2027' ASC
2012 IFPC<>192THEN2015ELSEGOSUB2017:V1=V3:C1=VN-2:GOSUB814:P=34:GO
SUB512:P=E1:GOSUB512:P=D1:GOSUB512:P=6:GOSUB512:P=255:GOSUB512:P=1
26:GOSUB512:P=4:GOSUB512:P=35:GOSUB512:P=183:GOSUB512:P=32:GOSUB51
2:P=250:GOSUB512:P=120:GOSUB512
2013 C1=VN-3:GOSUB814:GOSUB902:P=119:GOSUB512:GOSUB916:GOSUB514:RE
TURN' VARPTR OF STRING
2015 IFPC<>245THEN2020ELSEGOSUB2017:P=126:GOSUB512:P=254:GOSUB512:
P=45:GOSUB512:P=32:GOSUB512:P=2:GOSUB512:P=55:GOSUB512:P=35:GOSUB5
12:P=245:GOSUB512
2016 P=205:GOSUB512:P=90:GOSUB512:P=30:GOSUB512:P=241:GOSUB512:P=4
8:GOSUB512:P=7:GOSUB512:P=33:GOSUB512:P=0:GOSUB512:GOSUB512:P=183:
GOSUB512:P=237:GOSUB512:P=82:GOSUB512:GOSUB908:GOSUB908:V1=V3:GOSU
B916:GOSUB514:RETURN' VAL
2017 V3=V1:Q=Q+1:GOSUB514:GOSUB518:IFPN<>36THEN522:ELSEQ=Q+1:V1=PC
```

Listing 1 continued

See List of Advertisers on Page 145

Listing 1 continued

```
-65:MF=VS+V1*(SL+1):C1=MF:GOSUB814:GOSUB902:GOSUB514:RETURN' LOCAT
E STRING
2020 IFPC=216GOSUB514:GOSUB700:GOSUB970:P=205:GOSUB512:P=61:GOSUB5
12:P=11:GOSUB512:IFPC<>41THEN522ELSEGOSUB514:GOSUB916:RETURN:' INT
2025 IFPC<>229THEN2030ELSEV3=V1:GOSUB514:GOSUB514:GOSUB810:IFC1=-1
THENV1=PC-65:GOSUB518:GOSUB520:GOSUB514:GOSUB914:P=126:GOSUB512ELS
EP=58:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512:' PEEK
2027 P=38:GOSUB512:P=0:GOSUB512:P=111:GOSUB512:V1=V3:GOSUB916:GOSU
B514:RETURN:' PEEK
2030 IFPC=198THENV3=V1:GOSUB6000:P=42:GOSUB512:P=33:GOSUB512:P=65:
GOSUB512:V1=V3:GOSUB916:RETURN:' POINT
2032 V2=V1:Q=Q-1:GOSUB514:IFPC=206ANDPN>47ANDPN<58GOSUB810:GOSUB90
2:GOTO2040ELSEIFPC=206THENE1=0:D1=0:GOSUB902:GOTO2040
2035 GOSUB810:IFC1<>-1GOSUB902ELSEV1=PC-65:GOSUB518:GOSUB520:GOSUB
914:GOSUB514
2040 IFPC=58ORC=2V1=V2:GOSUB916:RETURN
2045 IFPC=206ANDPN>47ANDPN<58SG=205ELSESG=PC:GOSUB514
2050 GOSUB2060:IFSG=205GOSUB904ELSEGOSUB920
2055 GOTO2040
2060 GOSUB810:IFC1<>-1GOSUB900:RETURNELSEV1=PC-65:GOSUB518:GOSUB52
0:P=237:GOSUB512:P=91:GOSUB512:GOSUB840:GOSUB512:P=P1:GOSUB512:GOS
UB514:RETURN
2399 ' INTEGER VARIABLE TRANSFER ROUTINE
2400 IFPC=49ANDPN=207CM=1ELSECM=0
2410 Q=Q+1:GOSUB514:MF=VN:GOSUB880:Q=Q+1:GOSUB902:P=205:GOSUB512:P
=13:GOSUB512:P=38:GOSUB512
2420 IFCM=0P=26:GOSUB512:P=111:GOSUB512:P=19:GOSUB512:P=26:GOSUB51
2:P=103:GOSUB512:GOSUB916:RETURN:' TRANSFER BASIC VARIABLE INTO US
R VARIABLE
2430 GOSUB914:P=125:GOSUB512:P=18:GOSUB512:P=19:GOSUB512:P=124:GOS
UB512:P=18:GOSUB512:RETURN:' TRANSFER USR VARIABLE INTO BASIC VARI
ABLE
2499 ' PRINT ROUTINE
2500 GOSUB514:IFPC=58ORC=2P1=13:GOSUB2670:RETURN
2502 IFPC<>64ANDPC<>96THEN2600ELSEGOSUB514
2503 ' PRINT @
2504 IFPC<58Q=Q-1:GOSUB2580:C1=C1+15360:GOSUB514:GOSUB814:GOSUB902
ELSEV1=PC-65:GOSUB518:GOSUB520:GOSUB514:GOSUB514:GOSUB914:D1=60:E1
=0:GOSUB900:GOSUB904
2508 IFPC<>247P=34:GOSUB512:P=32:GOSUB512:P=64:GOSUB512:GOTO2600EL
SEGOSUB514:IFPC<>40THEN522ELSEGOSUB2580:GOSUB910
2510 GOSUB514:IFPC=205GOSUB514:IFPC<>247THEN522ELSEP=35:GOSUB512:G
OTO2508
2512 IFPC=59GOSUB514
2514 IFPC<>58ANDC<>2THEN522ELSERETURN
2580 C$=""
2585 GOSUB514:IFPC<>41ANDPC<>44THENC$=C$+CHR$(PC):IFPC<48ORPC>57TH
EN522ELSEGOTO2585
2590 C1=VAL(C$):RETURN
2599 ' SINGLE PRECISION EXPRESSION & STRING PRINT ROUTINE
2600 IFPC>64ANDPC<91ANDPN=36Q=Q+1:V1=PC-65:C1=VS+V1*(SL+1):GOSUB26
80:GOTO2630
2610 IFPC=34PC=PEEK(Q):Q=Q+1:MF=VN:FP=1:GOSUB880:FP=0:GOSUB2680:GO
TO2630
2620 Q=Q-1:GOSUB700:GOSUB970:P=205:GOSUB512:P=189:GOSUB512:P=15:GO
SUB512:GOSUB968:' SINGLE PRECISION PRINT
2630 IFPC=44THEN522
2640 IFPC=59GOSUB514:IFPC<>58ANDC<>2THEN2600ELSERETURN
2650 IFPC=58ORC=2P1=13:GOSUB2670:RETURN
2660 GOTO522
2670 P=62:GOSUB512:P=P1:GOSUB512:P=205:GOSUB512:P=58:GOSUB512:P=3:
GOSUB512:RETURN:' SCREEN CONTROL ROUTINE
2680 GOSUB836:GOSUB902:GOSUB968:GOSUB514:IFPC=34GOSUB514:RETURNELS
ERETURN:' PRINT STRING
2699 ' INPUT
2700 P=205:GOSUB512:P=179:GOSUB512:P=27:GOSUB512:GOSUB514:IFPN=36T
HENGOSUB518:GOSUB908:Q=Q+1:V1=PC-65:MF=VS+V1*(SL+1)-1:C1=MF:GOSUB8
14:GOSUB902:GOTO4707ELSEP=35:GOSUB512:P=205:GOSUB512:P=108:GOSUB51
2:P=14:GOSUB512
2710 IFPN=37P=205:GOSUB512:P=127:GOSUB512:P=10:GOSUB512:V1=PC-65:G
OSUB840:E1=P:D1=P1:GOSUB902:P=237:GOSUB512:P=75:GOSUB512:P=33:GOSU
B512:P=65:GOSUB512:P=113:GOSUB512:P=35:GOSUB512:P=112:GOSUB512:GOS
UB514:GOSUB514:IFPC<>58ANDC<>2THEN522ELSERETURN
2720 Q=Q-1:CF=1:GOSUB535:CF=0:P=58:GOSUB512:P=175:GOSUB512:P=64:GO
SUB512:P=222:GOSUB512:P=4:GOSUB512:GOSUB922:P=205:GOSUB512:P=204:G
OSUB512:P=10:GOSUB512:GOSUB565:GOSUB972:GOSUB514:IFPC<>58ANDC<>2TH
EN522ELSERETURN
2999 ' INTEGER IF - THEN ROUTINE
3000 GOSUB514:IFPN<>37THEN3100ELSEGOSUB520:V1=PC-65:GOSUB914:GOSUB
908:GOSUB514
3005 IFPC=212ANDPN=213ORPC=213ANDPN=212W1=1:Q=Q+1:GOTO3035
3010 IFPC=214ANDPN=213ORPC=213ANDPN=214W1=2:Q=Q+1:GOTO3035
3015 IFPC=212ANDPN=214ORPC=214ANDPN=212W1=3:Q=Q+1:GOTO3035
3020 IFPC=212W1=4
3025 IFPC=214W1=5
3030 IFPC=213W1=6
3035 GOSUB514:IFPC<58ORPC=206GOSUB810:GOSUB902ELSEGOSUB518:GOSUB52
0:V1=PC-65:GOSUB914:GOSUB514
3040 IFPC<>202ANDPC<>141THEN522ELSEGOSUB514:IFPC=141GOSUB514
3045 GOSUB810:IFC1<0ORC1>500THEN522
3050 P=205:GOSUB512:P=57:GOSUB512:P=10:GOSUB512
3055 GOTO3150
3099 ' SINGLE PRECISION IF - THEN ROUTINE
3100 Q=Q-1:GOSUB700:GOSUB934
3105 IFPC=212ANDPN=213ORPC=213ANDPN=212W1=1:Q=Q+1:GOTO3135
3110 IFPC=214ANDPN=213ORPC=213ANDPN=214W1=2:Q=Q+1:GOTO3135
3115 IFPC=212ANDPN=214ORPC=214ANDPN=212W1=3:Q=Q+1:GOTO3135
3120 IFPC=212W1=4
3125 IFPC=214W1=5
3130 IFPC=213W1=6
3135 GOSUB700:GOSUB936:GOSUB950
3140 IFPC<>202ANDPC<>141THEN522ELSEGOSUB514:IFPC=141GOSUB514
3145 GOSUB810:IFC1<0ORC1>500THEN522
3150 D=D1:E=E1:ONW1GOTO3155,3160,3165,3170,3175,3180
3155 GOSUB3185:GOSUB3200:RETURN
3160 GOSUB3185:GOSUB3190:RETURN
3165 GOSUB3195:RETURN
3170 P=40:GOSUB512:P=3:GOSUB512:GOSUB3200:RETURN
3175 P=40:GOSUB512:P=3:GOSUB512:GOSUB3190:RETURN
3180 GOSUB3185:RETURN
3185 P=202:GOSUB3300:RETURN
3190 P=242:GOSUB3300:RETURN
3195 P=194:GOSUB3300:RETURN
3200 P=250:GOSUB3300:RETURN
3205 P=195:GOSUB3300:RETURN
3300 GOSUB512:K=K+1:A(K)=M:P=E:GOSUB512:P=D:GOSUB512:RETURN
3310 K1=K1+1:A1(K1)=M:RETURN
3499 ' GOTO ROUTINE
3500 GOSUB514:GOSUB810:IFC1<0ORC1>500THEN522ELSED=D1:E=E1:GOSUB320
5:RETURN
3699 ' GOSUB ROUTINE
3700 GOSUB514:GOSUB810:IFC1<1ORC1>500THEN522ELSED=D1:E=E1:P=205:GO
```

Listing 1 continued

Listing 1 continued

```
SUB3300:RETURN
3799 ' RETURN
3800 GOSUB924:P=233:GOSUB512:RETURN
3999 ' SINGLE PRECISION VARIABLE TRANSFER ROUTINE
4000 IFPC=49ANDPN=207THENCM=1ELSECM=0
4010 GOSUB514:GOSUB514:MF=VN:GOSUB880:GOSUB514:GOSUB902:P=205:GOSU
B512:P=13:GOSUB512:P=38:GOSUB512
4020 IFCM=0GOSUB908:GOSUB932:GOSUB565:GOSUB972:RETURN
4030 GOSUB930:GOSUB565:GOSUB932:GOSUB924:GOSUB972:RETURN
4499 ' STRING ASSIGNMENT ROUTINE
4500 GOSUB514:V1=PC-65:MF=VS+V1*(SL+1)
4510 GOSUB514:IFPC<>36THEN522
4520 GOSUB514:IFPC<>213THEN522
4530 GOSUB514:IFPC=247THEN4600ELSEIFPC>64ANDPC<91THEN4700ELSEIFPC<
>34THEN522
4540 PC=PEEK(Q):Q=Q+1:FP=1:GOSUB880:FP=0
4550 GOSUB514:IFPC=34GOSUB514
4560 IFPC=205ANDPN=247THENGOSUB908:GOTO4710
4570 IFPC=205ANDPN=34THENPC=PEEK(Q+1):Q=Q+2:FP=1:NN=1:GOSUB881:FP=
0:GOTO4550ELSEIFPC=205THENGOSUB514:IFPC>64ANDPC<91ANDPN=36THENQ=Q+
1:GOSUB908:P=43:GOSUB512:GOTO4705ELSE522
4580 RETURN
4599 ' STRING ASSIGNMENT USING CHR$
4600 C1=MF:GOSUB836:GOSUB902
4610 GOSUB514:IFPC<>40THEN522ELSEC$=""
4615 GOSUB514:IFPC>64ANDPC<91THENV1=PC-65:GOSUB520:GOSUB514:IFPC<>
41THEN522ELSEP=58:GOSUB512:GOSUB840:GOSUB512:P=P1:GOSUB512:P=119:G
OSUB512:GOTO4660
4620 IF(PC<48ORPC>57)ANDPC<>41THEN522
4630 IFPC<>41THENC$=C$+CHR$(PC):GOSUB514:GOTO4620
4640 C1=VAL(C$)
4650 GOSUB910
4660 GOSUB514:IFPC=205GOSUB514:IFPC<>247THENP=35:GOSUB512:Q=Q-2:GO
SUB514:GOTO4720ELSEP=35:GOSUB512:GOTO4610
4670 IFPC=58ORC=2THENP=35:GOSUB512:C1=0:GOSUB910:RETURNELSEGOTO522
4699 'STRING ASSIGNMENT USING PREVIOUSLY DEFINED VARIABLE
4700 IFPN<>36THEN522ELSEQ=Q+1:C1=MF-1:GOSUB814:GOSUB902
4705 V1=PC-65:MX=VS+V1*(SL+1)-1:C1=MX:GOSUB836:GOSUB900
4707 P=35:GOSUB512:P=19:GOSUB512:P=26:GOSUB512:P=119:GOSUB512:P=18
3:GOSUB512:P=32:GOSUB512:P=249:GOSUB512:GOSUB514
4710 IFPC=205ANDPN=247THENQ=Q-1:P=43:GOSUB512:GOTO4660
4720 IFPC=205ANDPN=34THENGOSUB908:GOTO4570ELSEIFPC=205THENGOSUB514
:GOSUB518:IFPN<>36THEN522ELSEQ=Q+1:P=43:GOSUB512:GOTO4705
4730 RETURN
4999 ' FOR ROUTINE
5000 C1=M+7:GOSUB836:GOSUB514:GOSUB518:GOSUB520:V1=PC-65:GOSUB840:
GOSUB514:D(V1)=D1:E(V1)=E1:IFPC<>213THEN522
5010 GOSUB514:IFPC<65ORPC=206GOSUB810:J1=0:ID=D1:IE=E1:ELSEJ1=1:V2
=PC-65:GOSUB520:C1=VT+V2*2:GOSUB836:ID=D1:IE=E1:GOSUB514
5020 IFPC<>189THEN522
5030 GOSUB514:IFPC<65ORPC=206GOSUB810:J2=0:FD=D1:FE=E1ELSEJ2=1:V3=
PC-65:GOSUB520:C1=VT+V3*2:GOSUB836:FD=D1:FE=E1:Q=Q+1
5040 IFJ2=0THENP=33ELSEP=42
5050 GOSUB512:P=FE:GOSUB512:P=FD:GOSUB512:GOSUB926
5060 IFJ1=0THENE1=IE:D1=ID:GOSUB902
5070 IFJ1=1THENP=42:GOSUB512:P=IE:GOSUB512:P=ID:GOSUB512
5080 GOSUB916:IFPEEK(Q-1)<>58ANDPEEK(Q-1)<>0THEN522ELSERETURN
5499 ' NEXT ROUTINE
5500 GOSUB514:GOSUB518:GOSUB520:V1=PC-65:GOSUB914:GOSUB928:GOSUB93
0:GOSUB926:GOSUB920:GOSUB924:P=35:GOSUB512:P=194:GOSUB512:P=E(V1):
GOSUB512:P=D(V1):GOSUB3310:GOSUB512:GOSUB924:GOSUB514:RETURN
5999 ' POINT, SET & RESET
6000 IFPC=130THENW=1ELSEIFPC=131THENW=128ELSEIFPC=198THENW=0
6010 MA=M
6020 GOSUB514:IFPC<>40THEN522ELSEGOSUB514:GOSUB810:IFC1=-1GOSUB518
:GOSUB520:V1=PC-65:GOSUB840:D2=P1:E2=P:C2=1ELSEE2=E1:C2=0:IFPC<>44
THEN522
6030 IFC2=1GOSUB514:IFPC<>44THEN522
6040 GOSUB514:GOSUB810:IFC1=-1GOSUB518:GOSUB520:V1=PC-65:GOSUB840:
D3=P1:E3=P:C3=1ELSEE3=E1:C3=0:IFPC<>41THEN522
6050 IFC3=1GOSUB514:IFPC<>41THEN522
6060 GOSUB514:IFPC<>58ANDC<>2THEN522
6070 C1=MA+18+C2+C3:GOSUB836:GOSUB902:M=M-1:GOSUB3310:M=M+1:GOSUB9
26:E1=126:D1=7:GOSUB902:P=62:GOSUB512:P=W:GOSUB512:P=245:GOSUB512
6080 IFC2=1THENP=58:GOSUB512:P=E2:GOSUB512:P=D2:GOSUB512:ELSEP=62:
GOSUB512:P=E2:GOSUB512
6090 P=245:GOSUB512
6100 IFC3=1THENP=58:GOSUB512:P=E3:GOSUB512:P=D3:GOSUB512ELSEP=62:G
OSUB512:P=E3:GOSUB512
6110 E1=80:D1=1:GOSUB918:RETURN
6499 ' POKE ROUTINE
6500 GOSUB514:GOSUB810:IFC1=-1THENGOSUB518:GOSUB520:V1=PC-65:GOSUB
914:GOSUB514ELSEGOSUB902
6510 IFPC<>44THEN522
6520 GOSUB514:GOSUB810:IFC1=-1GOSUB518:GOSUB520:V1=PC-65:GOSUB840:
E1=P:D1=P1:P=58:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512:GOSUB514ELSEP
=62:GOSUB512:P=E1:GOSUB512
6530 IFPC<>58ANDC<>2THEN522
6540 P=119:GOSUB512:RETURN
6999 ' ROUTINE FOR CODE TO STORE 2-D ARRAY ADDRESSES
7000 GOSUB524:PRINT"CODE TO STORE 2-D ARRAY ADDRESSES":PRINT:C1=VD
:GOSUB836:GOSUB902
7010 C1=VD+4*NT*DT*DT:GOSUB836:P=221:GOSUB512:GOSUB902:C1=4*DT:GOS
UB814:GOSUB900
7020 C1=NT*DT:GOSUB814:P=1:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB512
7030 P=221:GOSUB512:P=117:GOSUB512:P=0:GOSUB512
7040 P=221:GOSUB512:P=35:GOSUB512
7050 P=221:GOSUB512:P=116:GOSUB512:P=0:GOSUB512
7060 P=221:GOSUB512:P=35:GOSUB512:GOSUB904:P=13:GOSUB512
7070 C1=M-12:GOSUB836:P=194:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB3310:
GOSUB512
7080 P=5:GOSUB512:P=14:GOSUB512:P=255:GOSUB512:C1=M-18:GOSUB836:P=
242:GOSUB512:P=E1:GOSUB512:P=D1:GOSUB3310:GOSUB512
7090 PRINT@856,"MAIN CODE BEGINS";:RETURN
7100 GOSUB514:Q1=Q:IFPC=58THEN7104ELSEIFC<>2THENGOTO7100'CLEAR
7102 Q=Q+4
7104 GOSUB514:RETURN
7110 IS=1:DO=0:DT=0:NO=0:NT=0:NS=2'DIM
7112 C$="":GOSUB514:Q1=Q:IFPC=58THENQ=Q-4:Q1=Q:RETURNELSEIFC=2THEN
RETURNELSEIFPC=44THEN7112
7114 V=1:IFPC<65ORPC>90THENGOTO7126ELSEV3=PC-64
7116 IFPN=36THENQ=Q+1:V=2:ELSEIFPN>47ANDPN<58THENIS=PN-47:Q=Q+1:GO
TO7112
7118 GOSUB514:IFPC<>40THENGOTO7126
7120 GOSUB514:IFPC=41THEN7130ELSEIFPC=44THENDT=VAL(C$):C$="":GOTO7
122ELSEIFPC<65ANDPC>90THENGOTO7126ELSEC$=C$+CHR$(PC):GOTO7120
7122 GOSUB514:IFPC=41THEN7124ELSEIFPC=<65ANDPC>90THENGOTO7126ELSEC
$=C$+CHR$(PC):GOTO7122
7124 NT=V3:IFDT=VAL(C$)THENGOTO7112
7126 PRINT"ERROR IN DIM STATEMENT":END
7130 IFV=1THENNO=V3:DO=VAL(C$)
7132 IFV=2THENNS=V3:SL=VAL(C$)
7134 GOTO7112
7135 GOSUB514:IFPC<>58ANDC<>2THEN7135
```

Listing 1 continued

Listing 1 continued

```
7136 C1=VN:GOSUB814:P=175:GOSUB512:GOSUB902:P=119:GOSUB512:C1=VN+1
:GOSUB814:GOSUB900:C1=-MR*16384-VN-1:GOSUB814:P=1:GOSUB512:P=E1:GO
SUB512:P=D1:GOSUB512:P=237:GOSUB512:P=176:GOSUB512:RETURN'CLEAR
7200 ' ************* END OF THE COMPILER ***************
10000 ' ***** BASIC PORTION OF THE ARRAY SORTING PROGRAM *****
10010 DIM A(400):INPUT "ARRAY DIMENSION";N%
10020 FOR I%=1 TO N%:A(I%)=200*RND(0):NEXT I%
10030 FOR I%=1 TO N%:PRINT A(I%);:NEXT I%:PRINT
10040 PRINT "PRESS A KEY TO RUN"
10050 A$=INKEY$:IF A$="" THEN 10050
10060 X=USR(0)
10070 PRINT:PRINT "DONE":PRINT:PRINT "PRESS A KEY TO PRINT THE SOR
TED ARRAY"
10080 A$=INKEY$:IF A$="" THEN 10080
10090 FOR I%=1 TO N%:PRINT A(I%);:NEXT I%
```

End

Program Listing 2. Demonstration program using string Input, ASC, Clear, and Dimension statements.

```
10 CLEAR50:DIMA$(40)
20 CLS
30 PRINT:PRINT"Answer Yes, No, or Quit";:INPUTA$:A%=ASC(A$)
40 'THE ASCII VALUE FOR Q IS 81, q IS 113
50 IFA%=81THEN120ELSEIFA%=113THEN120
60 'THE ASCII VALUE FOR N IS 78, n IS 110
70 IFA%=78THEN100ELSEIFA%=110THEN100
80 'THE ASCII VALUE FOR Y IS 89, y IS 121
90  IFA%=89THEN110ELSEIFA%=121THEN110ELSEGOTO30
100 PRINT"The answer was no.":GOTO30
110 PRINT"The answer was yes.":GOTO30
120 PRINT:PRINT"You asked to quit."
130 PRINT"You are now returned to BASIC."
500 END
```

End

Program Listing 3. Using substitute programming techniques for compilation purposes.

```
10 CLEAR50:DIMB$(60)
20 CLS
30 PRINT"EXAMPLES OF VARPTR(A$), LEN(A$), AND LEFT$(A$,5)"
40 PRINT"USING SUBSTITUTE PROGRAM FOR COMPILATION PURPOSES":PRINT
50 PRINT"ENTER A TEN CHARACTER STRING":INPUTA$
60 B$=A$:A%=VARPTR(B$):PRINT"VARPTR LOCATION IS ";A%:B%=PEEK(A%)
70 IFB%<=10THEN80ELSEPRINT"STRING IS GREATER THAN TEN CHARACTERS":
GOTO50
80 IFB%=10THEN90ELSEPRINT"STRING IS LESS THAN TEN CHARACTERS":GOTO
50
90 B%=A%+1:B%=PEEK(B%):C%=A%+2:C%=PEEK(C%)
100 B=C%*256+B%:IFB<32768THEN110ELSEB=B-65536
110 B%=INT(B):PRINT"STRING LOCATION IS ";B%
120 'THE NEXT LINE WILL NOT GIVE THE SAME RESULTS IN BASIC
130 B%=B%+5:POKEB%,0
140 PRINT"THE ORIGINAL STRING IS . . . ";A$
150 PRINT"THE EQUIVALENT OF B$=LEFT$(A$,5) IS . . . ";B$
500 END
```

End

Program Listing 4. Basic Pong game.

```
5 DIMA$(40),A1
10 Q%=20:A%=25
15 L%=25:O%=20
20 CLS:PRINT"How fast (1 to 1000)?":INPUT D%:CLS:M%=0:S%=0:PRINT
@132,M%:PRINT@186,S%
25 FORY%=9TO39:SET(0,Y%):SET(127.Y%):NEXTY%
30 C%=1
35 PRINT@158,"Pong";
40 K%=1
45 FORI%=20TO25:SET(7,I%):SET(120,I%):NEXTI%
50 FORX%=1TO126:SET(X%,9):SET(X%,39):NEXTX%
55 Y%=10:Z%=INT(100*RND(0)):X%=Z%+10
60 SET(X%,Y%)
65 FORI%=1TOD%:NEXTI%
70 RESET(X%,Y%)
75 V%=PEEK(14337):U%=PEEK(14338):W%=PEEK(14340):Z%=V%+U%+W%
80 IFZ%=0THEN105
85 IFU%<>128THEN90:ELSEGOSUB205
90 IFU%<>16THEN95:ELSEGOSUB210
95 IFW%<>2THEN100:ELSEGOSUB215
100 IFV%<>2THEN105:ELSEGOSUB220
105 Z%=PEEK(14400):IFZ%=4THEN500
110 X%=X%+C%
115 Y%=Y%+K%
120 IFY%>=39THEN125ELSEIFY%>9THEN140
125 IFX%>122THEN185
130 IFX%<6THEN180
135 B%=POINT(X%,Y%):IFB%<0THEN160
140 IFX%>122THEN185
145 IFX%<5THEN180
150 B%=POINT(X%,Y%):IFB%<0THEN170
155 GOTO60
160 K%=-K%
165 GOTO115
170 C%=-C%
175 GOTO110
180 S%=S%+1:PRINT@186,S%;:IFS%>15THEN190:ELSEGOTO55
185 M%=M%+1:PRINT@132,M%;:IFM%>15THEN190:ELSEGOTO55
190 PRINT@982,"Play Again? (Y/N)";
195 Z%=PEEK(14344):IFZ%=2THEN20
200 Z%=PEEK(14338):IFZ%=64THEN500ELSEGOTO195
205 RESET(120,L%):O%=O%-1:L%=L%-1:SET(120,O%):RETURN
210 RESET(120,O%):O%=O%+1:L%=L%+1:SET(120,L%):RETURN
215 RESET(7,A%):Q%=Q%-1:A%=A%-1:SET(7,Q%):RETURN
220 RESET(7,Q%):Q%=Q%+1:A%=A%+1:SET(7,A%):RETURN
500 END
```

End

back-track

The Super User Friendly Disk Utility
For TRS-80* Models I, III and 4III

- Duplicate Most Protected Disks
- Edit (Zap) Sectors
- Powerful Disk Formater

menu driven with help command for the novice
short cuts for the experienced user
self booting, does not require DOS
simultaneous Hex and ASCII display
reads SD, DD or Mixed Density with auto density detect
reads single or double sided, 5 1/4" or 8" disks

YOU CAN NOW —

Read, edit, write and step through sectors and tracks(including information between sectors)
Duplicate most CP/M** disks for other P.C.'s
Duplicate standard format, special format & data disks
Identify protection secrets and create your own schemes
Read track address; see actual track and sector numbers
Repair CRC errors
Exchange data address marks
Format 128, 256, 512 or 1024 byte sectors
Interface with Monitor/Disassembler
Analyze unknown DOS systems
Write special tracks; make protected disks
Print video display through RS232 or parallel port
AND MORE

✓265

$38.95 Plus $1.50 S & H

for unprotected disk and 31-page manual

MasterCard VISA

Tennessee residents include 4.5% sales tax
For personal checks allow 2 weeks
Specify TRS-80* model I, III, or 4 III

(615)482-2007
RIDGE-TEC, INC.
740 Robertsville Road
Oak Ridge, TN 37830

*TRS-80 trademark Tandy Corp. **CP/M trademark Digital Research Corp.

NEW FOR PROFILE® USERS

PROfix * III™
RESTRUCTURE & TRANSFER UTILITY
$49.95

For PROFILE® III PLUS
Model III/4

Needs change—Experience teaches—and if you could redefine your data base today, you probably would change a few things—

WELL, NOW YOU CAN!

PROfix*III Provides you, as a user of Profile III Plus, with a powerful, easy-to-use package that allows you to reorganize your data base TO MEET TODAY'S NEEDS, and then moves your existing data into your new file structure.

*** **FEATURES** ***

- EASY TO USE—no special charts to fill out
- FAST—assembly language speed
- RE-ARRANGE FIELDS—even across segments
- DROP OR ADD FIELDS
- CHANGE FIELD LENGTHS—R/L justify data
- CHANGE NUMBER OF SEGMENTS
- CREATE SORTED FILE—drop deleted fields
- LITERAL INSERTION—specified fields
- COMPLETE USER MANUAL—with examples

TO ORDER—Send $49.95
Plus $2.50 for Handling and Postage
Check, M.O., VISA/MC
(for charge card, give expiration date, number)
—To—
BLUE RIDGE SOFTWARE
230 Chesterfield Road
Lynchburg, VA 24502

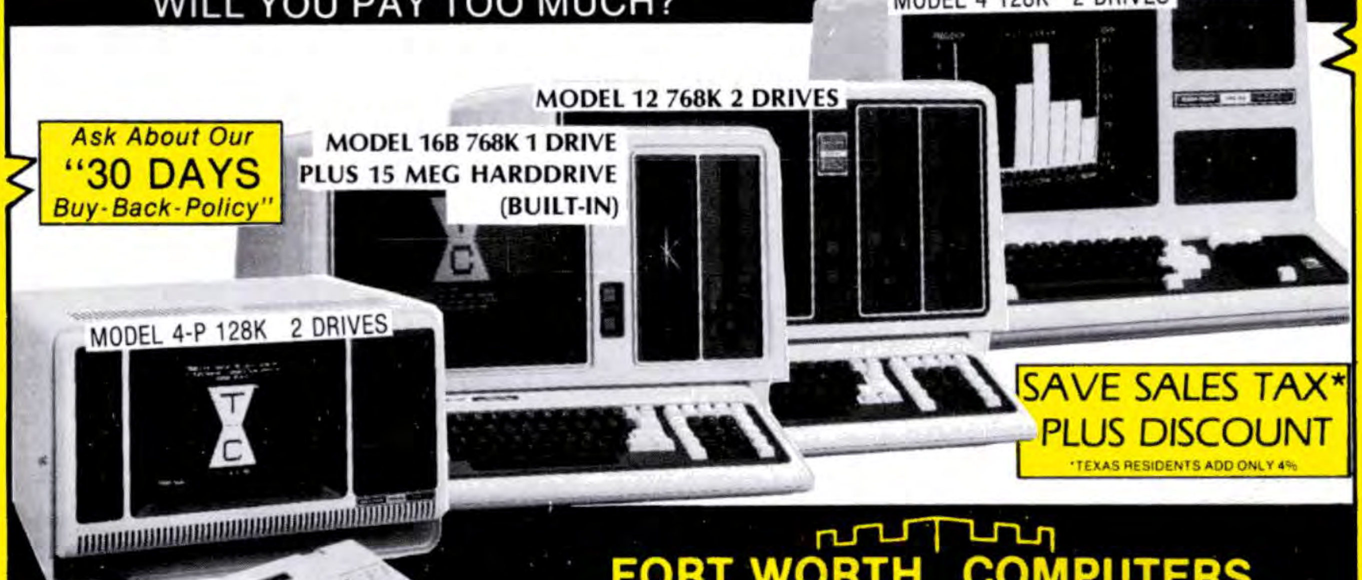
For phone orders or more information
Call (804) 239-0574
$1.00 off on phone orders!
Virginia residents add 4% sales tax
Most orders filled within 24 hours
(allow 2 weeks for checks to clear)

✓186

Profile is a registered trademark of Tandy Corp.

TRS-80™ "CAN YOU BUY DIRECT?"

WILL YOU PAY TOO MUCH?



Ask About Our "30 DAYS Buy-Back-Policy"

SAVE SALES TAX*
PLUS DISCOUNT
*TEXAS RESIDENTS ADD ONLY 4%

FORT WORTH COMPUTERS

WE ARE SERIOUS ABOUT SAVING YOU MONEY

(Located 30 miles from Fort Worth)
377 Plaza • GRANBURY • NR FORT WORTH, TEXAS 76048

TOLL FREE NUMBER: 1-800-433-S-A-V-E

Monday thru Friday — 9:00 a.m. to 5:00 p.m. Texas Time
(Order Inquiries/Customer Service &) IN TEXAS: 817-573-4111

✓214

TM: TRADE MARK OF RADIO SHACK

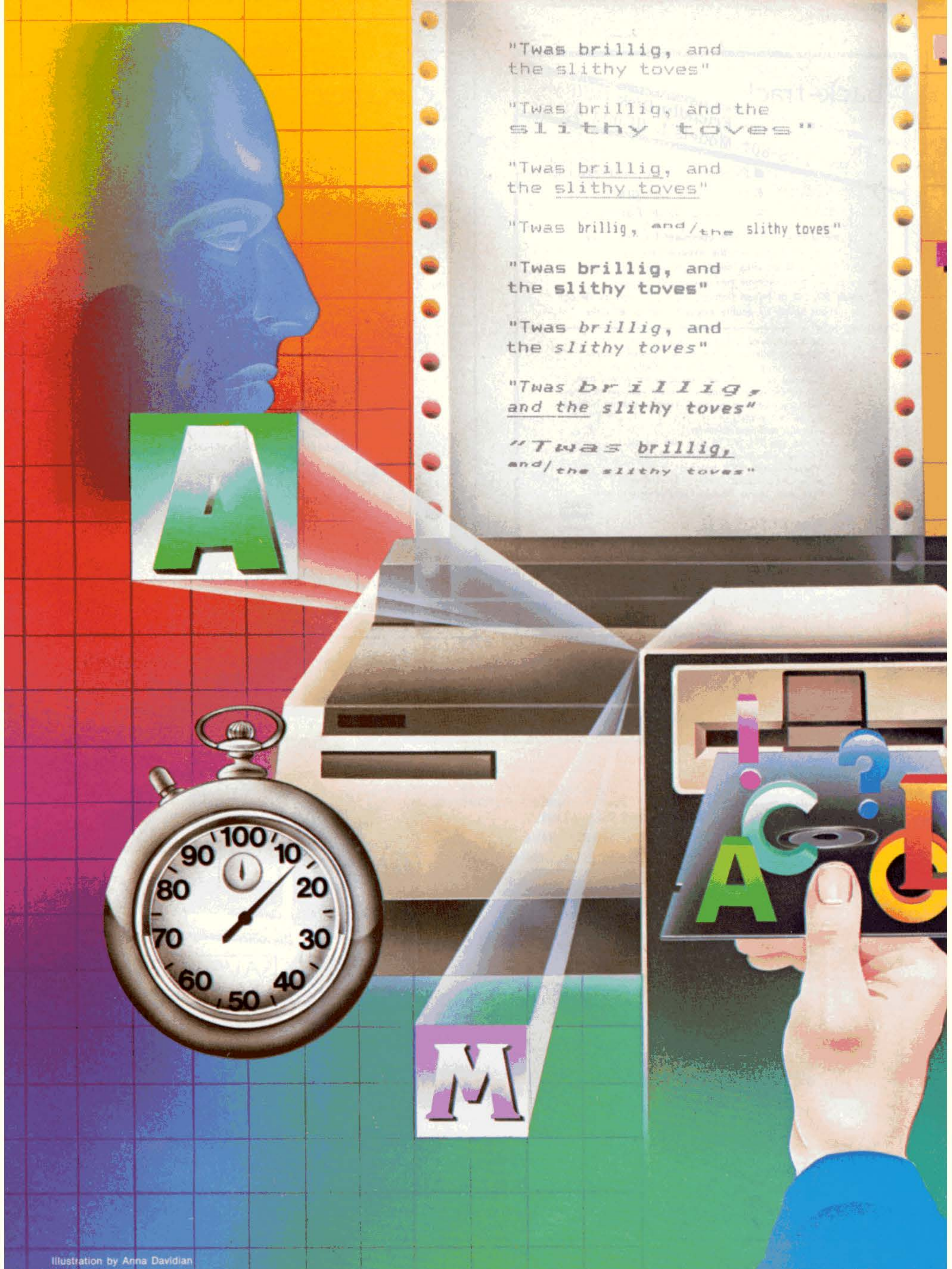
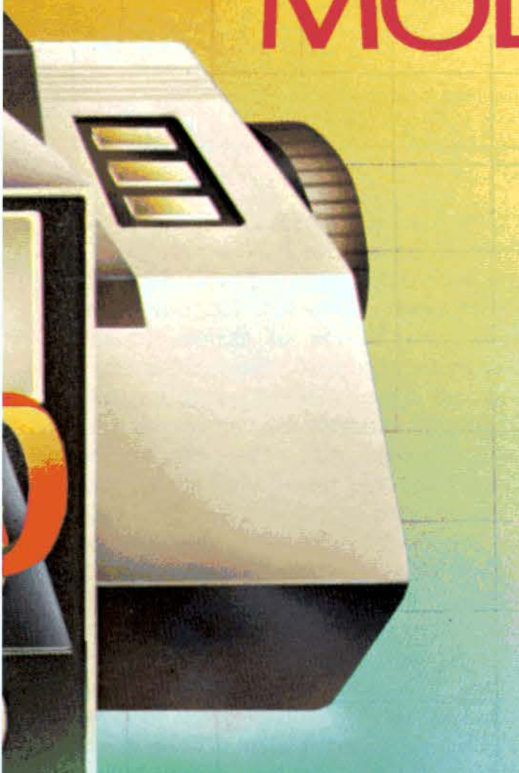


Illustration by Anna Davidian

MODEL 4 SCRIPSIT THE WRITE WAY



by Clifford Knight

What Model 4 Scripsit should have been—ScripAid reassigns Scripsit's command keys, allows text preview on-screen, and provides a potpourri of type fonts.

Like most Model 4 owners, I looked forward to Model 4 Scripsit with great anticipation. But I was disappointed when I found that Model 4 Scripsit was nothing more than a reassembled version of the Model III program. Tandy even included the old keycap "pasties" for Scripsit's command keys. I decided to write a complete enhancement package called ScripAid that includes all the features I want in Model 4 Scripsit.

ScripAid reassigns Scripsit's command keys to their mnemonic counterparts (see Table 1), lets you preview text on your screen before printing it out, and supports compressed, double-width, emphasized, italic, underline, double-strike, and super- or subscript characters (see the Figure).

ScripAid consists of five programs. Program Listings 1 and 2 provide the actual Scripsit enhancements. Program Listing 3 installs the enhancement programs with a JCL file. Program Listing 4 lets you develop a customized printer driver for ScripAid. Program Listing 5 gives you a help file template.

The Programs

Program Listing 1, Enhance/SRC, is the source code that implements ScripAid's enhanced commands. It also loads a printer driver and, if applicable, resets the Model 4's extra memory banks and changes Scripsit's start-of-text buffer pointers to allow more memory space for ScripAid routines.

Program Listing 2, Rekey/SRC, contains patches to the command codes that change the more confusing Scripsit commands to easily remembered keys (yes, you can throw away the pasties). It also includes patches that alter the default left- and right-margin settings and the video line width. You can alter these fur-

```
>TM=5
>C=Y
\3*** ScripAid \4printer control demonstration\$ ***
>C=N
\#

This is an example of the power of \6\3ScripAid\#\&.

\3You can set the emphasized mode at the start of your text, and then add \6DOUB
LE-STRIKE\& for added visual impact.

\#You can switch to \2DOUBLE-WIDTH\" and back.

Or underline \5here\% and \5there\% as you wish.

Even \1compressed\! and \'super\&/\7sub\& scripts may be used.

\3Y\#\2O\"\3\6U\&\# can \4do it all\$ \3 with \6\5ScripAid\%\&.
```

```
             *** ScripAid printer con rol demonstration ***

    This is an example of the power of ScripAid.

    You can set the emphasized mode at the start of your text, and
    then add DOUBLE-STRIKE for added visual impact.

    You can switch to D O U B L E - W I D T H and back.

    Or underline here and there as you wish.

    Even compressed and super/sub scripts may be used.

    YOU can do it all  with ScripAid.
```

Figure. Text with ScripAid commands embedded (top) and as it appears when you print it out (bottom).

ther if you want to boot up the program with other values set. In addition, I modified all prompts and error messages to provide upper- and lowercase characters.

Program Listing 3, the JCL file, installs the Scripsit overlays assembled from the source code so you can use ScripAid.

Program Listing 4, Driver/BAS, lets you customize a printer driver file. It prompts you to assign printer control codes for each ScripAid print command. For instance, \1 is the ScripAid command that prints a compressed character. When Driver prompts you to input the code that makes your printer print a compressed character, you should find the right code in your printer manual and type it in.

Every printer requires a different driver as there are no standard printer control codes. Table 2 (p. 65) shows the codes I used with my Epson MX-80 printer. Table 3 (p. 65) defines the print commands you embed in your text.

Program Listing 5, Helper, gives you the information you need to develop a help file.

Creating ScripAid

Type in and assemble Program Listings 1 and 2, and save them as Enhance/OVL and Rekey/OVL.

Now you're ready to build the JCL file in Program Listing 3. Use the DOS Build command or Scripsit (saved in ASCII format) and write it to the same disk that contains the overlay files. (I recommend that you use a system disk in drive 1 as the target for all the ScripAid files. This way, you'll end up with the entire ScripAid program on a single disk.)

If you're a single-drive user, write all files to drive zero, then delete all references to drive 1 from the JCL file.

To create ScripAid/CMD, first insert a back-up copy (for added security) of the original Scripsit program in drive zero. The system disk containing both the ScripAid overlay files and the JCL file should be in drive 1. Now type in DO SCRIPAID:1. This executes the DOS commands of the JCL file, transforming your version of Scripsit to the upgraded ScripAid. Single-drive users should copy their Scripsit to a system disk and assemble the two overlay files on the same disk. Then execute the modified JCL (mentioned above) and you're ready to use ScripAid.

Although you don't have to use ScripAid's new print codes, you do have to include at least one printer driver so the program can load and execute. You should name this driver Default/PDC and put it on the same disk as ScripAid. If you have access to more than one printer, you can develop driver files for each printer. When calling ScripAid, specify the driver you want on the TRSDOS command line by typing in SCRIPAID followed by the driver file name.

You can create a driver file for your printer with the help of Program Listing 4. Here, the first screen prompts you to enter either a file name for the file you want to create, or hit control-Q to quit. If you hit the enter key, the program uses the file name Default.

Once you specify the file name,

The Key Box

LOAD 80

Model 4
64K RAM
Assembly Language
Disk Basic
TRSDOS 6.X.X
Model 4 Scripsit
Editor/Assembler

✓ See List of Advertisers on Page 145

ScripAid creates two files, one with the extension /PDC for the driver, and one with the extension /HLP for the driver help screen. The next screen displays the file name used, the current code assignments, and the shifted complements for keys 1–9. A prompt at the bottom of the screen gives you the following options: to enter the key you want programmed, to save the current assignments (S), or to abort and cancel all changes (A).

If you hit any of the control-code keys, you'll be prompted to enter up to five decimal numbers that, in turn, form the format control string assigned to the previously selected key. After entering the five values you'll see that they've been assigned to the selected key. If your printer's control sequence doesn't require 5 bytes (values) then hit the enter key in response to the next CODE # prompt to terminate data entry.

Then ScripAid asks you to enter a short description (up to 70 characters) of this control string's function. The program displays this information when you request printer help from ScripAid. When you've finished programming, type in S to save the files. ScripAid then returns you to the first menu) from which you can opt to create another driver or quit ScripAid.

If you want to include a help file with ScripAid, Program Listing 5 will help you build your own. Such a file, however, takes up more than 20K of disk space, which is why I find a quick-reference card more convenient.

The Source Code

Program Listing 1, ScripAid's source code, is fully commented, but it's important that you understand the main logic flow. First, the program defines the supervisor calls (SVCs) and then assigns the hook vectors to and from Scripsit. Lines 880–1150 alter the sign-

Reassigned Scripsit Commands

General Commands	
Clear-B	Enter text block
Clear-I	Insert text
Clear-G	Start paragraph
Clear-L	End line
Clear-N	Force new page
Clear-Z	Enter window mode
Delete Commands	
Clear-G	Delete paragraph
Clear-L	Delete line
Clear-S	Delete spaces to left of cursor
Clear-W	Delete word
Insert Commands	
Clear-B	Insert block
Clear-L	Insert line
Exchange Commands	
Clear-B	Exchange block
Clear-G	Exchange paragraph
Clear-W	Exchange words

Enhanced Scripsit Commands

Disk-related	
Break-Q	Query disk directory
Break-K	Kill named file
Printer related	
Break-P,Cnn	Send CHR$(nn) to printer
Break-P,T	Send top-of-form character to printer
Break-P,V	Preview text (use 0–9 to control speed, @ to pause, shift/clear to abort)
Help Screens	
Break-HK	Keyboard help file
Break-HP	Printer help file
Bank Switching	
Break-Bnn	Make memory bank nn active buffer

Table 1. ScripAid commands.

ScripAid Commands	Command Descriptions
\1	Set compressed-character mode
\!	Reset compressed-character mode
\2	Set double-width mode
\"	Reset double-width mode
\3	Set emphasized mode (except in compressed, sub-, or superscript modes)
\#	Reset emphasized mode
\4	Set italics mode
\$	Reset italics mode
\5	Set underline mode
\%	Reset underline mode
\6	Set double-strike mode
\&	Reset underline, sub-, and superscript modes
\7	Set subscript mode
\'	Set superscript mode
\8	
\(	
\9	
\)	

Table 2. Suggested control key assignments for the Epson MX-80 Type III printer.

ScripAid Commands	Printer Codes
\1	15
\!	18
\2	27 87 1
\"	27 87 0
\3	27 69
\#	27 70
\4	27 52
\$	27 53
\5	27 45 1
\%	27 45 0
\6	27 71
\&	27 72 27 84
\7	27 83 1
\'	27 83 0
\8	
\(	
\9	
\)	

Table 3. Definitions of the print commands.

Program Listing 1. Enhance/SRC, the source code for ScripAid enhancements.

```
00100 ;********************** ENHANCE ************************
00110 ;
00120 ;  An enhancement package for Model 4 SCRIPSIT adding:
00130 ;        1> Directories from command line
00140 ;        2> Kill files from command line
00150 ;        3> Embedded printer control codes
00160 ;        4> Printer codes from command line
00170 ;        5> Formatted output to video monitor
00180 ;        6> Send TOF character to printer
00190 ;        7> Bank switching for 3 text buffers
00200 ;        8> On-line Help available
00210 ;
00220 ;  Install via:
00230 ;
00240 ;      1. memory (clear)
00250 ;      2. load scripsit/cmd
00260 ;      3. load rekey/ovl
00270 ;      4. load enhance/ovl
00280 ;      5. dump scripaid/cmd (s=x'3800',e=x'6970',t=x'6810')
00290 ;
00300 ;*****************************************************
00310 ;
00320 ;
00330 ;        Supervisor Call Equates.................
00340 DSPLY   EQU     0AH     ;DISPLAY A MESSAGE LINE
00350 LOGOT   EQU     0CH     ;DISPLAY & LOG A MESSAGE
00360 PAUSE   EQU     10H     ;DELAY FOR "BC" COUNTS
00370 CHNIO   EQU     14H     ;PASS TO NEXT MODULE IN DEV CHAIN
00380 ERROR   EQU     1AH     ;POST AN ERROR MESSAGE
00390 HIGH    EQU     64H     ;GET/PUT HIGH$/LOW$
00400 FLAGS   EQU     65H     ;POINT IY TO SYSTEM FLAGS
00410 FSPEC   EQU     4EH     ;VERIFY FILESPEC
00420 BANK    EQU     66H     ;MEMORY BANK SVC
00430 FEXT    EQU     4FH     ;ADD FILESPEC IF NEEDED
00440 OPEN    EQU     3BH     ;OPEN A FILE
00450 GET     EQU     03H     ;GET A BYTE FROM FCB OR DCB
00460 CLOSE   EQU     3CH     ;CLOSE A FILE
00470 CMNDR   EQU     19H     ;DO CMD & RET
00480 DECHEX  EQU     60H     ;ASCII -> BIN IN BC
00490 PRT     EQU     06H     ;CHAR TO PRINTER
00500 KEY     EQU     01H     ;WAIT F/ KEY
00510 DSP     EQU     02H     ;CHAR TO VIDEO
00520 KEYIN   EQU     09H     ;GET A LINE (*KI)
00530 DODIR   EQU     22H     ;DSP DIR OF DRIVE
00540 REMOV   EQU     39H     ;KILL A FILE
00550 EXIT    EQU     16H     ;RETURN TO DOS
00560 KBD     EQU     08H     ;WAIT FOR KEY
00570 HEXDEC  EQU     61H     ;CONVERT BIN-> ASCII
00580 VDCTL   EQU     0FH     ;VIDEO CONTROL SVC
00590 POSN    EQU     42H     ;POSITION TO RECORD
00600 READ    EQU     43H     ;READ A RECORD (DISK)
00610 ;
00620 ;
00630 ;        Scripsit Address Equates...............
00640 ;
00650 PRCALL  EQU     46C9H   ;LOCATION OF @PRT SVC CALL
00660 PRTCHK  EQU     4AE1H   ;GET *PR STATUS ROUTINE
00670 SCRPIN  EQU     5416H   ;NORMAL RE-ENRTY POINT
00680 ERRINV  EQU     5401H   ;"INVALID COMMAND" ENTRY
00690 PREND   EQU     54EDH   ;HOOK F/ END OF PRINT ROUT
00700 SCRIPS  EQU     3900H   ;ENTRY TO SCRIPSIT
00710 SCRXIT  EQU     383DH   ;EXIT CODE
00720 DOUT    EQU     5554H   ;FORMATTED OUTPUT CALL
00730 KSCAN   EQU     47B6H   ;KBSCAN IN SCRIPSIT
00740 OLDBUF  EQU     622BH   ;OLD TEXT BUFFER START
00750 CKPRPM  EQU     4A9CH   ;CK PRINT PARMS
00760 PRPMCK  EQU     4AF1H   ;SCIPSIT PRINT PARMCK
00770 BRKCK   EQU     48EEH   ;SCRIPSIT <BRK> PARMCK
00780 BRKTAB  EQU     4F6DH   ;<BRK> PARM TABLE
00790 RETBRK  EQU     48F1H   ;RET TO <BRK> ROUT
00800 SCRPBF  EQU     5F5AH   ;SCRIPSIT INT DISK I/O
00810 PRRETN  EQU     4BA5H   ;RET AFTER *PR OUT
00820 PRCLN   EQU     54FCH   ;CLEAN UP AFTER *PR
00830 RETVEC  EQU     5EB2H   ;RETURN VECTOR
00840 WRTVID  EQU     4811H   ;REWRITE VIDEO IMAGE
00850 ;
00860 ;
00870 ;         *** Patch Sign-on Banner ***
00880           ORG     3EFBH
00890           DEFB    78              ;LENGTH OF MESSAGE
00900 ;         DEFM    ' * ScripAid * - Model 4 Version 3.1 - (c) 1984 by Paladin
 Software - C. Knight'
00910 *LIST OFF
00920           DEFM    ' * ScripAid * - Model 4 Version 3.1 - (c) 1984 by Paladin
 Software - C. Knight'
00930 *LIST ON
00940 ;
00950 ;
00960 ;         *** Patch to eliminate reset of <BREAK> vector ***
00970 ;
00980           ORG     3906H
00990           DEFW    0000H
01000           DEFW    0000H
01010           DEFW    0000H
01020 ;
01030 ;
01040 ;         *** Set jumps to extension code ***
01050           ORG     PRCALL
01060           CALL    PRTOUT
01070 ;
01080           ORG     CKPRPM
01090           CALL    CKPARM
```

Listing 1 continued

on banner, eliminate the code to reset the break vector (which creates problems with the spooler), and arranges jumps from Scripsit to ScripAid.

The initialization routine, assembled at memory address 6810 hexadecimal (hex), loads the selected or default printer driver and resets buffer pointers in Scripsit, making room for ScripAid. As the routine loads the driver codes, it saves the file name with the extension /HLP for later use as a printer Help file. The routine also verifies and initializes the optional memory buffers, if available.

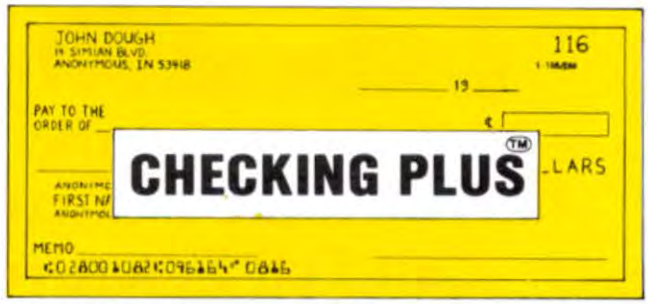
The code for ScripAid actually starts at line 3050 of Program Listing 1. The PALBRK routine intercepts the break/special-command call and checks for valid ScripAid commands, such as Query, Kill, Bank, and Help. You call these by entering the first letter of each command, followed by any needed parameters.

The CKPARM routine in line 6840 of the source code parses the commands added to the Scripsit Print routine. Then ScripAid scans for Scripsit's invisible and pause commands. A reset carry flag (NC) indicates that Scripsit has located a valid parameter. Under these circumstances, ScripAid relinquishes control to the Scripsit parameter. If the Scripsit parameters aren't found, ScripAid tests for these valid ScripAid commands: video (V), top-of-form (T), and control-code output (C).

ScripAid outputs formatted text to the video by altering the output SVC number from 6 (@PRT) to 2 (@DSP), which reroutes each character to the monitor instead of to the printer port. Before output begins, however, ScripAid links to the Scripsit routine that scans for the shift-clear (abort output) key sequence. This link calls a ScripAid routine to scan for the @ key (which pauses the output) or one of the number

ANYONE CAN USE OUR SOFTWARE!

MAIL PAC II and CHECKING PLUS represent a new generation of computer software. Our software doesn't even need a manual! Anyone can quickly learn to use either package by following simple, explicit on-screen instructions. But, for those of you who insist -- we've included a complete user's manual as well.



MAIL PAC II FEATURES:

High Capacity --
Stores from one to one million names in Zip Code, Numerical or Alphabetical order. The only limitation is your disk storage space.

Flexibility --
Prints your mailing list on 1, 2, 3, or 4-across labels (with up to 5 user-defined lines on each label) or as a compact, user-designed directory. The record length is completely user-defined, and each field within each of your records is completely variable (allowing storage of any number of characters for any particular address entry). Devote fields to telephone numbers, codes, or even special messages related to each particular name on file.

Powerful --
A built-in word processor allows you to create personalized form letters for each address on your list (or just a particular group of addresses).

Ease of Use --
Create new mailing lists, review existing lists, handle changes of address, delete cancelled names, sort lists, and purge duplicate names from your files. Complete on-screen instructions tell you in plain english exactly how to accomplish all of these tasks.

MAIL PAC II $99.95

CHECKING PLUS FEATURES:

Complete Check Register --
Checking Plus stores your entire check register in a disk file, and then uses the data to balance your account, track your expenses, and help you make budget projections. Review the entire checkbook, enter checks, deposits, fees and adjustments, mark outstanding checks when paid, and get an instant cash balance at any time. All data can be viewed onscreen or printed out in report form.

Tax Preparation --
Storage of monthly and yearly totals and other important information aids in income tax preparation, for your personal use or for your accountant.

Handles the Details --
Store names and addresses for frequently written checks, then print checks to fit standard window envelopes, eliminating the need for extra typing. You can even store any comment, explanation or other message (up to 255 characters) related to a particular check.

Automatic Monthly Bill Payments --
Enter amounts and names of payees for all of your fixed-cost monthly payments, and then sit back while the system automatically prints checks.

CHECKING PLUS $99.95

Both Programs Require 48K And At Least One Disk Drive (2 Drives For Higher Storage Capacities). And All Hard Disk Systems Are Supported.

FREE–TRS-80 Mod I,III & 4 programs supplied on DosPlus (minimum system). Complete DosPlus also available.

FOR YOUR TRS-80™ • APPLE™ • IBM PC™ • NEC™ • OSBORNE™ • XEROX™ • KAYPRO™ • TELEVIDEO™ • ZENITH™ • SANYO™ DEC™ • TI PROFESSIONAL COMPUTER™ • SUPERBRAIN JR.™ • EPSON™ • Any CP/M™ Computer

CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

OUR SOFTWARE CATALOG

H & E Computronics, Inc., has mailed more than 1 million software catalogs since 1978. Send $2 for our new 64-page catalog today! (We also send you our catalog FREE with every order). DEALER INQUIRIES WELCOME

30-DAY MONEY BACK GUARANTEE

*** ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE ***
DELIVERY SUBJECT TO AVAILABILITY

COMPUTRONICS

50 N. PASCACK RD., SPRING VALLEY, N.Y. 10977

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA & MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA & MEXICO

24 HOUR ORDER LINE
(914) 425-1535

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

Listing 1 continued

```
01100 ;
01110          ORG    BRKCK
01120          JP     PALBRK
01130 ;
01140          ORG    SCRXIT
01150          JP     EXIT1
01160 ;
01170 ;
01180 ;
01190 ;
01200 ;        *** SCRIPSIT Extension Code ***
01210 ;
01220          ORG    6810H
01230 ;
01240 ENTRY    LD     (OLDSP),SP        ;SAVE STACK POINTER
01250          CALL   INITF             ;LOAD DRIVER FILE
01260          CALL   BNKONE            ;SET PTRS F/ BANK1
01270          CALL   CLRBNK            ;RESET BANKS 1&2
01280          JP     SCRIPS            ;GOTO SCRIPSIT
01290 ;
01300 ;
01310 ;
01320 ;
01330 INITF    LD     A,(HL)            ;GET NXT CMD CHAR
01340          CALL   EOLNCK            ;END OF LINE ?
01350          JR     NZ,MOVFSP         ;NZ= FIL SPEC
01360          LD     HL,DEFNAM         ;HL-> "DEFAULT/PDC"
01370 MOVFSP   LD     DE,FCB            ;DE-> FCB
01380          LD     A,FSPEC           ;SVC
01390          RST    28H
01400          JR     Z,FSPOK           ;Z= FILESPEC OK
01410          LD     HL,BADFIL         ;HL-> "BAD FILE" MSG
01420          JP     LOGERR            ;GOTO LOG ERR & RET
01430 ;
01440 FSPOK    LD     HL,PDCTXT         ;HL-> DEFAULT EXTENSI(
01450          LD     A,FEXT            ;SVC
01460          RST    28H               ;ADD EXTENSION
01470          PUSH   DE                ;SAVE FCB POINTER
01480          CALL   SAVFSP            ;FSPEC-> HOLD
01490          LD     HL,BANNER         ;HL-> BANNER MESSAGE
01500          CALL   DSPMSG            ;OUTPUT SAME
01510          POP    HL
01520          PUSH   HL                ;HL-> FILESPEC
01530          CALL   DSPMSG
01540          LD     C,0DH             ;C/R
01550          LD     A,DSP
01560          RST    28H
01570          POP    DE                ;DE-> FCB
01580 ;
01590 EXTN     LD     HL,SCRPBF         ;HL-> DISK I/O BUFFER
01600          LD     B,90              ;LRL=90
01610          LD     A,OPEN            ;SVC
01620          RST    28H
01630          JP     NZ,ERR
01640          LD     HL,CTLTAB         ;HL-> COMMAND TABLE
01650 ;
01660 RDLOOP   LD     B,90              ;90 BYTES MAX
01670          LD     DE,FCB            ;DE-> FCB
01680          LD     A,GET             ;SVC
01690          RST    28H
01700          JR     NZ,EOF            ;ERROR COULD BE EOF
01710          LD     (HL),A            ;STORE IN CMD TABLE
01720          INC    HL                ;ELSE BUMP PTR
01730          DJNZ   RDLOOP            ;LOOP 'TIL EOF
01740 ;
01750 EOF      CP     1CH               ;ERROR = EOF ?
01760          PUSH   AF                ;SAVE ERROR STATUS
01770          LD     A,CLOSE           ;SVC
01780          RST    28H
01790          JR     NZ,ERR
01800          POP    AF                ;GET ORIGINAL ERR
01810          JR     NZ,ERR
01820          RET                      ;ELSE RETURN
01830 ;
01840 ;
01850 SAVFSP   LD     HL,PHLPFS         ;HL-> HOLDING AREA
01860 SAVFSL   LD     A,(DE)            ;GET A CHAR
01870          INC    DE
01880          LD     (HL),A            ;SAVE IT
01890          INC    HL
01900          CP     '/'               ;EXT YET?
01910          JR     NZ,SAVFSL
01920          LD     (HL),'H'
01930          INC    HL
01940          LD     (HL),'L'
01950          INC    HL
01960          LD     (HL),'P'
01970          INC    HL
01980          LD     (HL),0DH          ;"HLP" AS EXTENSION
01990          RET                      ;DONE
02000 ;
02010 ;
02020 ;
02030 EOLNCK   CP     0DH               ;C/R ?
02040          RET    Z
02050          CP     03H               ;ETX
02060          RET                      ;GO BACK W/ STATUS
02070 ;
02080 ;
02090 CLRBNK   LD     C,1               ;BANK 1
02100          CALL   GETCLR
02110          LD     C,2               ;BANK 2
02120 GETCLR   LD     B,2               ;BANK AVAIL?
02130          PUSH   BC                ;SAVE #
02140          LD     A,BANK
02150          RST    28H               ;DO IT
```

Listing 1 continued

keys (0–9) used to control video output speed. ScripAid also bypasses the printer ready test and intercepts the end-of-output routine to restore printer output. It ignores any embedded control codes during video output.

The top-of-form (TOF) and control-code routines send characters directly to the printer. A single byte value, 0C hex, is the standard TOF code. The P,C option calls the SVC @HEXDEC, converting the ASCII decimal input to binary for output to the printer.

Line 5020 is the entry to the printer driver routine. After you save this routine, ScripAid tests the last byte sent. If it's a backslash character (\), ScripAid branches to NXTST, where it checks the current character. (To type a backslash, hit the standard slash key and the clear key at the same time.)

If the current character is something other than a backslash character, ScripAid scans the printer control code table. If ScripAid finds a match, it outputs the appropriate control string from the 90-byte control code list loaded when you called up ScripAid.

Note that ScripAid sends control codes only if the printer (SVC 6) is the active output device. This avoids confusion in the video preview mode. An invalid control code translates to a space character for printer output.

The final portion of the source code defines the messages and storage areas ScripAid uses.

Using ScripAid

As you're writing text, you can embed ScripAid's printer codes directly in your document to activate your printer's special abilities. The Figure shows an example of text with printer codes embedded and of the final product.

ScripAid has two classes of commands: the reassigned Scripsit editing commands and the enhanced com-

✓ See List of Advertisers on Page 145

mands I added. You invoke all of the reassigned Scripsit commands by first hitting the clear key and another key to put you in the appropriate editing mode (clear-D for delete, clear-I for insert, and so on), just as you would with the standard program. Then hit clear and the relevant command key to actually make changes (see Table 1). The reassigned Scripsit commands perform the same functions they did in the original commercial program, so I won't explain them here.

You execute all of my enhanced commands by hitting the break key and an associated character or characters (see Table 1). Query (Q), clears the screen and then, like all ScripAid commands, scans the remainder of the command line for additional parameters, such as a file extension to be matched or a drive specification. Query displays the disk name, creation date, and available space for all visible, non-system files on drive zero as the default. You must specify the extension and/or drive number that you want matched. To return to ScripAid, press any key.

Listing 1 continued

```
         POP     BC              ;GET #
         RET     NZ              ;NZ=NOT AVAIL
         LD      B,0             ;SELECT BANK
         LD      A,BANK
         RST     28H
         PUSH    BC              ;SAVE OLD BANK
         LD      A,8CH
         LD      (8000H),A       ;INTO BUFFER
         XOR     A
         LD      (8001H),A       ;NO TEXT
         POP     BC              ;GET OLD BANK #
         LD      A,BANK          ;@BANK SVC
         RST     28H
         RET                     ;DONE
;
;
;
;BADFIL  DEFM    'Invalid filespec!'
;        DEFB    0DH
*LIST OFF
BADFIL   DEFM    'Invalid filespec!'
         DEFB    0DH
*LIST ON
;
;
DSPMSG   PUSH    DE              ;SAVE DE
         LD      A,DSPLY         ;SVC, DISPLAY MSG @ (HL)
         RST     28H
         POP     DE
         RET     Z               ;Z= O.K.
         LD      L,A             ;L= ERROR CODE
         LD      H,00H
         OR      0CH             ;SET RETURN BITS
ERR      LD      C,A             ;PASS IN "C"
         LD      A,ERROR         ;SVC, LOG AN ERROR
         RST     28H
         JR      DOSXIT          ;RET W/ ERROR
;
;
LOGERR   LD      A,LOGOT         ;SVC, DISPLAY & LOG MSG
         RST     28H
         LD      HL,0FFFFH       ;SET ERROR RETURN
DOSXIT   LD      SP,$-$          ;RESET SP
OLDSP    EQU     $-2
         RET                     ;GOTO DOS READY
;
;
FCB      DEFS    32              ;FCB FOR PDC INPUT
;DEFNAM  DEFM    'DEFAULT/'
;PDCTXT  DEFM    'PDC'
;        DEFB    0DH
*LIST OFF
DEFNAM   DEFM    'DEFAULT/'
PDCTXT   DEFM    'PDC'
         DEFB    0DH
*LIST ON
;
;BANNER  DEFB    0FH             ;CURSOR OFF
;        DEFB    1CH             ;HOME CURSOR
;        DEFB    1FH             ;CLS
;        DEFB    0CDH
;        DEFM    '*** ScripAid ***  (c) 1984 Paladin'
;        DEFM    ' Software'
;        DEFW    0A0AH
;        DEFW    0A0AH
;        DEFW    0A0AH
;        DEFB    0D4H
;        DEFM    'Loading driver --> '
;        DEFB    03H
*LIST OFF
BANNER   DEFB    0FH
         DEFB    1CH
         DEFB    1FH
         DEFB    0CDH
         DEFM    '*** ScripAid ***  (c) 1984 Paladin Softwa..
         DEFW    0A0AH
         DEFW    0A0AH
         DEFW    0A0AH
         DEFB    0D4H
```

Listing 1 continued

Kill (K) checks for the entry of a valid file name, opens the file, then calls the @REMOV SVC to kill the file from the directory of the specified or default drive. Note that you must first open, like all ScripAid commands, to remove it. When you finish editing text and want to preview what it will look like when it's printed out, press the clear-P-V keys.

Bank (B), followed by any digit from 1 to 3, incorporates the specified 32K memory bank as the active text buffer. Bank 1 is the active bank at power-up and has the largest capacity (35K to 39K depending on your system configuration). Banks 2 and 3 each hold about 32K of text. If a requested bank isn't available, a message is displayed. When switching banks, the selected video line width and paragraph indent are saved or restored.

If you've created your own help file (see Program Listing 5), the ScripAid Help command (H), followed by K (keyboard) or P (printer), calls for the appropriate help screen. These are stored as ASCII files with a record length of 80 bytes. The up- and down-arrow keys page forward and backward through the help file. You should note that the letter H, when not followed by another character, calls for the hyphenation command.

End Notes

As with any enhancement program, it's necessary to create the enhanced commands around the original program. Whenever you do this, however, you must make some compromises, and ScripAid has two of which you should be aware.

The first problem is that the embedded printer control codes aren't treated as invisible format commands by Scripsit's formatting routines. Because of this, full justification (flush-left and -right margins) won't function properly with embedded codes. To eliminate this problem, you can turn off full justification (J = N) and carefully assemble any lines using special functions.

The second problem concerns text that's previewed on the monitor with the P,V command. If the text contains page number blocks, it's video display is marred by momentary flashes of the editing screen whenever it calculates a new page number. You can avoid this problem by leaving out page number blocks until you're ready to print the document. ■

You can write to Clifford Knight at Box 253, Manomet, MA 02345.

Listing 1 continued

```
                DEFM    'Loading driver --> '
                DEFB    03H
*LIST ON
;
ENINIT          EQU     $
;
;
;
;               *** Start of ScripAid ***
;
                ORG     OLDBUF          ;STARTS @ OLD TEXT BUF
;
PALBRK          INC     HL              ;HL-> NEXT CHAR
                CP      'Q'             ;QUERY?
                JR      Z,QUERY         ;IF SO
                CP      'K'             ;KILL ?
                JP      Z,KILL          ;DO IT
                CP      'B'             ;BANK?
                JP      Z,BANKS         ;GO TO IT
                CP      'H'             ;HELP?
                JR      NZ,GOODBK       ;NZ= NOT "H"
                LD      C,A             ;ELSE SAVE CHAR
                LD      A,(HL)          ;GET NEXT CHAR
                CP      0DH             ;CR?
                JP      NZ,HELP         ;IF NOT, GET HELP
                LD      A,C             ;GET ORIG CHAR
GOODBK          LD      HL,BRKTAB       ;HL-> BRK PARMS
                JP      RETBRK          ;GO BACK
;
;
QUERY           CALL    SKPSPL+1        ;SKIP SPACES
                JR      Z,NOEXT         ;Z= CR FOUND
                CP      '/'             ;EXT SPECIFIED?
                CALL    Z,MOVEXT        ;IF FOUND
                JR      Z,USREXT
NOEXT           PUSH    HL              ;SAVE CMDLINE PTR
                LD      HL,EXTMSG-1     ;HL-> "$$$"
                CALL    MOVEXT          ;MOVE TO FCB
                POP     HL              ;GET PTR
USREXT          LD      A,(HL)          ;GET A CHAR
                CP      ':'             ;DRIVE SPECIFIED?
                INC     HL
                JR      Z,USRDRV        ;IF SO
                CP      0DH             ;CR?
                JR      NZ,USREXT       ;IF NOT EOLN
                LD      (HL),'0'        ;SET DRV 0
USRDRV          CALL    CLRVID          ;CLEAR VIDEO
                LD      A,(HL)          ;GET DRIVE #
                AND     07H             ;0-7 ONLY
                LD      C,A             ;INTO C
                LD      (PARM1),A       ;SAVE DRIVE #
                LD      HL,SCRPBF       ;HL-> BUFFER AREA
                LD      B,04            ;SET GET FREE
                LD      A,DODIR         ;@DODIR SVC
                RST     28H
                JP      NZ,DSKERR
                LD      HL,DRVMSG       ;HL-> "DRIVE:"
                LD      A,DSPLY
                RST     28H
                LD      A,(PARM1)       ;GET DRIVE #
                ADD     A,30H           ;CONV TO ASCII
                LD      C,A
                LD      A,DSP
                RST     28H
                LD      HL,NAMSG        ;HL->"Pack ID: "
                LD      A,DSPLY
                RST     28H
                LD      HL,SCRPBF       ;HL-> FREE STRING
                CALL    EIGHT           ;OUTPUT DISK NAME
                PUSH    HL              ;SAVE DISK INFO PTR
                LD      HL,CREMSG       ;HL->"Created:"
                LD      A,DSPLY
                RST     28H
                POP     HL              ;GET INFO PTR
                CALL    EIGHT           ;OUTPUT CREATION DATE
                PUSH    HL              ;SAVE INFO PTR
                LD      HL,FREMSG       ;HL-> "Free:"
                LD      A,DSPLY
                RST     28H
                POP     HL
                LD      E,(HL)          ;LSB TOTAL SPACE
                INC     HL
                LD      D,(HL)          ;MSB TOTAL
                INC     HL              ;HL-> LSB FREE
                LD      A,(HL)          ;GET LSB
                INC     HL
                LD      H,(HL)          ;GET MSB
                LD      L,A             ;HL= FREE SPACE
                PUSH    DE              ;SAVE TOTAL
                CALL    KOUT            ;OUTPUT FREESP
                LD      C,'/'
                LD      A,DSP
                RST     28H
                POP     DE              ;GET TOTAL
                EX      DE,HL
                CALL    KOUT
                LD      HL,BLKLIN       ;HL-> 0A0DH
                LD      A,DSPLY
                RST     28H
                LD      A,(PARM1)       ;GET DRIVE #
                LD      C,A             ;INTO C
                LD      B,2             ;SET DIR F/ "EXT"(HL)
                LD      HL,EXTBUF       ;HL-> EXTENSION TO MATCH
                LD      A,DODIR
                RST     28H             ;DIR TO VIDEO
```

Listing 1 continued

Listing 1 continued

```
                JR      NZ,DSKERR
                CALL    BOTLIN          ;CURSOR TO LINE 23
                LD      HL,VDMSG        ;HL-> "CONTINUE?"
                LD      A,DSPLY
                RST     28H
;
;
;
WAITBK          LD      A,KEY           ;@KEY SVC
                RST     28H
GOBACK          LD      C,0FH           ;CURSOR OFF
                LD      A,DSP
                RST     28H
                CALL    CLRVID          ;CLEAR SCREEN
                JP      SCRPIN          ;BACK TO SCRIPSIT
;
;
KILL            CALL    SKPSPL+1        ;SKIP SPACES
                JR      Z,GOERR         ;IF SO
                LD      DE,FCB1         ;DE-> FCB
                LD      A,FSPEC         ;@FSPEC
                RST     28H
                JR      NZ,DSKERR
                LD      B,0             ;LRL=256
                LD      A,OPEN          ;@OPEN TO KILL
                RST     28H
                JR      NZ,DSKERR
                LD      A,REMOV         ;@REMOV
                RST     28H
                JR      NZ,DSKERR
                JR      GOBACK
;
;
GOERR           LD      C,0FH           ;CURSOR OFF
                LD      A,DSP
                RST     28H
                JP      ERRINV          ;INVALID CMD
;
;
;
DSKERR          OR      0C0H            ;SET RET & MSG
                LD      C,A             ;ERROR -> C
                PUSH    BC              ;SAVE ERROR
                CALL    BOTLIN          ;POSITION CURSOR
                POP     BC              ;GET ERROR
                LD      A,ERROR         ;@ERROR SVC
                RST     28H
                JR      WAITBK
;
;
TOP             POP     HL              ;FIX STACK
                LD      C,0CH           ;ASCII FORM FEED
                LD      A,PRT           ;@PRT SVC
                RST     28H
                JR      GOBACK          ;BACK TO SCRIPSIT
;
CONTRL          POP     AF              ;FIX STACK
                INC     HL              ;HL-> NEXT CHAR
CTLP1           CALL    SKPSPL+1        ;SKIP SPACES
                JP      Z,GOBACK        ;IF END
                CP      ','             ;DELIMITER ?
                JR      NZ,CONT1
                INC     HL              ;ELSE BUMP PAST
CONT1           LD      A,DECHEX        ;@DECHEX SVC
                RST     28H
                LD      A,PRT
                RST     28H
                JR      CTLP1           ;LOOP
;
VIDEO           LD      A,02H           ;@DSP SVC
                LD      (OUTSVC),A      ;SET VIDEO SVC
                LD      A,0EFH          ;RST 28H OPCODE
                LD      (PASRST),A      ;INTO @DELAY CALL
                LD      HL,0C9AFH       ;XOR A & RET OPCODES
                LD      (PRTCHK),HL     ;INTO CHECK ROUT
                LD      HL,VDWAIT       ;HL-> END WAIT ROUT
                LD      (PREND),HL      ;INTO SCRIPSIT VECTOR
                LD      A,0CDH          ;CALL OPCODE
                LD      (KSCAN),A       ;INTO SCRIPSIT
                LD      HL,DLYCK        ;HL-> DELAY CK ROUT
                LD      (KSCAN+1),HL    ;INTO SCRIPSIT
                CALL    BOTLIN
                OR      A               ;RESET CARRY FLAG
                RET                     ;GO BACK & OUTPUT
;
;
;
PRTCNT          LD      A,06H           ;@PRT SVC
                LD      (OUTSVC),A      ;STORE IT
                LD      HL,0D5C5H       ;PUSH BC & PUSH DE
                LD      (PRTCHK),HL     ;STORE THEM
                LD      HL,DOUT         ;HL-> FORMAT OUT ROUT
                LD      (PREND),HL      ;RESET VECTOR
                LD      A,28H           ;JR  Z  OPCODE
                LD      (KSCAN),A
                LD      HL,0AF02H       ;JR 2 BYTES, XOR A
                LD      (KSCAN+1),HL    ;RESET JUMP
                XOR     A               ;NOP OPCODE
                LD      (PASRST),A      ;INTO @DELAY CALL
                RET                     ;GO BACK
;
;
PRTOUT          PUSH    HL
                PUSH    DE              ;SAVE CURRENT CHAR
                PUSH    AF              ;SAVE ACCUM
                PUSH    BC
```

Listing 1 continued

Listing 1 continued

```
        LD      A,$-$           ;GET PREV CHAR
PRVCHR  EQU     $-1
        CP      '\'             ;"\"?
        LD      A,C             ;GET CURRENT CHAR
        JR      Z,NXTST         ;IF PREVIOUS="\"
SAVPRV  CP      '\'
        LD      (PRVCHR),A      ;SAVE AS PREVIOUS
        JR      Z,PRTXIT        ;IF CURR="\", NO ACT
NOSET   LD      BC,0301H        ;DELAY COUNT
        LD      A,PAUSE         ;@PAUSE SVC
PASRST  NOP                     ;INIT IN "PRINT"
        POP     BC              ;GET CHAR
        PUSH    BC              ;SAVE AGAIN
        LD      A,PRT           ;@PRT SVC
OUTSVC  EQU     $-1
        RST     28H
PRTXIT  POP     BC
SKPOUT  POP     AF              ;RESTORE ACCUM
        POP     DE              ;GET DE
        POP     HL
        RET                     ;DONE
;
;
NXTST   CP      '\'             ;2nd "\"?
        JR      NZ,CTLCK        ;IF SO
SAVSPC  LD      A,20H           ;" "
        JR      SAVPRV          ;SAVE AS PREV
;
;
CTLCK   LD      A,(OUTSVC)      ;GET OUTPUT SVC
        CP      02H             ;@DSP?
        POP     BC              ;GET CURRENT CHAR
        JR      NZ,CTLCN1       ;NZ=@PRT
        LD      A,20H           ;ASCII SPACE
        LD      (PRVCHR),A      ;SAVE AS PREVIOUS
        JR      SKPOUT          ;SKIP OUTPUT
;
;
;
CTLCN1  LD      HL,CTLTAB       ;HL-> CTL STRINGS
        LD      DE,LOOKTB       ;DE-> LOOKUP TABLE
        LD      B,18            ;18 CODES
LOOKLP  LD      A,(DE)          ;GET ONE
        INC     DE              ;DE-> NEXT CODE
        CP      C               ;MATCH?
        JR      Z,CTLFND        ;IF SO
        INC     HL
        INC     HL
        INC     HL
        INC     HL              ;5 BYTES/ENTRY
        INC     HL              ;HL-> NEXT CTL STR$
        DJNZ    LOOKLP          ;LOOP
        LD      C,20H           ;ELSE PASS A SPACE
CHRNUL  PUSH    BC              ;SAVE AS CURRENT
        JR      SAVSPC          ;GO DO IT
;
CTLFND  LD      B,05H           ;5 BYTES/CTL $
CTLOOP  LD      C,(HL)          ;GET ONE
        INC     HL              ;HL-> NEXT
        INC     C
        JR      Z,CTLDON        ;Z= END OF CTL$
        DEC     C               ;RESTORE
        LD      A,PRT           ;@PRT SVC
        RST     28H
        DJNZ    CTLOOP          ;LOOP
;
CTLDON  XOR     A               ;CLEAR ACCUM
        LD      C,A             ;SET A NULL
        JR      CHRNUL          ;OUTPUT NULL, SAVE " "
;
;
DLYCK   JR      NZ,CKCN1        ;IF NOT <SFT><CLR>
        CALL    PRTCNT          ;RESET *PR OUTPUT
        LD      HL,PRRETN
                                        ;HL-> CONT ROUT
        LD      (RETVEC),HL     ;INTO RET VECTOR
        JP      PRCLN           ;CONTINUE
;
CKCN1   CP      '@'             ;PAUSE ?
        JR      NZ,CKCN2        ;IF NOT
        PUSH    DE              ;SAVE TEXT PTR
        LD      A,KEY           ;@KEY SVC
        RST     28H             ;WAIT
        POP     DE
CKCN2   SUB     30H             ;CONVERT TO DEC
        JR      C,NOTNUM
        CP      0AH             ;>"9"?
        JR      NC,NOTNUM
        LD      (NOSET+2),A     ;INTO CODE ABOVE
NOTNUM  XOR     A
        RET
;
;
CLRVID  PUSH    HL
        LD      HL,CLRMSG       ;HL-> CLS STRING
        LD      A,DSPLY
        RST     28H
        POP     HL
        RET                     ;DONE
;
;
;
VDWAIT  CALL    DOUT            ;DO OUTPUT
        PUSH    DE
        PUSH    HL              ;SAVE REGS
        CALL    BOTLIN          ;POS'N CURSOR
```

Listing 1 continued

Listing 1 continued

```
        EI
        LD      HL,VDMSG        ;HL-> HIT RET TO CONT
        LD      A,DSPLY
        RST     28H
        LD      A,KEY           ;@KEY
        RST     28H
        LD      C,0FH           ;CURSOR OFF
        LD      A,DSP           ;@DSP SVC
        RST     28H
        POP     HL
        POP     DE
        JP      PRTCNT
;
;
EIGHT   LD      B,8             ;8 BYTES
EILOOP  LD      C,(HL)          ;GET A BYTE
        LD      A,DSP
        RST     28H
        INC     HL              ;HL-> NEXT BYTE
        DJNZ    EILOOP
        RET
;
;
KOUT    LD      DE,SCRPBF       ;DE-> BUFFER
        PUSH    DE              ;SAVE BUFFER START
        LD      A,HEXDEC
        RST     28H
        POP     HL              ;HL-> BUFFER
        LD      B,5
        CALL    EILOOP          ;OUTPUT 5 BYTES
        LD      C,'K'
        LD      A,DSP
        RST     28H
        RET                     ;DONE
;
;
MOVEXT  PUSH    AF              ;SAVE CHAR
        LD      B,3             ;3 CHARS/EXT
        LD      DE,EXTBUF       ;DE-> EXT BUFFER
EXTLP   INC     HL              ;BUMP PTR
        LD      A,(HL)          ;GET A CMDLINE CHAR
        CP      '0'
        JR      C,FILLUP        ;C= <"0"
        CP      '9'+1
        JR      C,OKEXT         ;C= "0"-"9"
        AND     5FH             ;FOLD -> U/C
        CP      'A'
        JR      C,FILLUP        ;C= >"9", <"A"
        CP      'Z'+1
        JR      NC,FILLUP       ;NC= >"Z"
OKEXT   LD      (DE),A          ;INTO BUFFER
        INC     DE              ;BUMP PTR
        DJNZ    EXTLP           ;LOOP
        POP     AF
        RET
;
FILLUP  LD      A,'$'           ;WCC
        DEC     HL              ;BACKUP PTR
        JR      OKEXT
;
;
;
;
BOTLIN  LD      HL,1700H        ;LINE 23, COL 0
        LD      B,03H           ;SET CURSOR OP
        LD      A,VDCTL         ;@VDCTL SVC
        RST     28H
        LD      C,1EH           ;CLEAR TO END OF LINE
        LD      A,DSP           ;@DSP SVC
        RST     28H
        RET
;
;
CKPARM  PUSH    DE              ;SAVE CMD LINE PTR
        CALL    PRPMCK          ;LOOK F/ ,P ,I ETC
        POP     HL              ;HL-> CMD LINE
        RET     NC              ;NC= PARM FOUND
        LD      A,(HL)          ;ELSE GET NEXT CHAR
        CP      'V'             ;VIDEO ?
        JP      Z,VIDEO         ;IF SO
        CP      'T'             ;TOP OF FORM ?
        JP      Z,TOP
        CP      'C'             ;CONTROL CODES?
        JP      Z,CONTRL
        SCF                     ;ELSE SET CARRY
        RET                     ;DONE
;
;
HELP    CALL    SKPSPC          ;SKIP SPACES
        SUB     'P'             ;PRINTER HELP?
        LD      HL,HLPFS        ;HL-> REG HELPFS
        LD      A,11            ;12 PAGES IN MAIN HELP
        JR      NZ,GETHLP       ;NZ= NOT *PR HELP
        LD      HL,PHLPFS       ;HL-> PRINTER FS
        XOR     A               ;MAX PAGE=0
GETHLP  LD      (MAXPAG),A      ;SAVE MAX PAGE #
        LD      DE,FCB1         ;DE-> FCB
        LD      A,FSPEC         ;@FSPEC
        RST     28H
        JP      NZ,DSKERR
        LD      A,OPEN          ;@OPEN
        LD      B,80            ;80 BYTE LRL
        LD      HL,SCRPBF       ;HL-> BUFFER
        RST     28H
        JP      NZ,DSKERR
PAGE0   XOR     A               ;CURRENT PAGE = 0
```

Listing 1 continued

Listing 1 continued

```
07170 SAVPAG  LD      (PARM1),A       ;SAVE CURR PAGE #
07180         CALL    DSPPAG          ;OUTPUT A PAGE
07190         JP      NZ,DSKERR
07200         CALL    BOTLIN          ;CURSOR @ LINE 23
07210         LD      HL,HLPMSG
07220         LD      A,DSPLY         ;@DSPLY
07230         RST     28H
07240 HLPLP   LD      A,KEY           ;@KEY
07250         RST     28H
07260         CP      0AH             ;DOWN ARROW
07250         RST     28H
07260         CP      0AH             ;DOWN ARROW
07270         JR      Z,NXTPAG
07280         CP      0BH             ;UP ARROW ?
07290         JR      Z,PRVPAG
07300         CP      80H
07310         JR      NZ,HLPLP        ;LOOP IF NOT
07320         LD      DE,FCB1
07330         LD      A,CLOSE
07340         RST     28H
07350         JP      NZ,DSKERR
07360         JP      GOBACK
07390 ;
07400 PRVPAG  LD      A,(PARM1)       ;GET CURR PAGE #
07410         OR      A               ;TEST IT
07420         JR      NZ,DECPAG       ;NZ, CURR<>0
07430         LD      A,(MAXPAG)      ;ELSE GET LAST #
07440         INC     A
07450 DECPAG  DEC     A
07460         JR      SAVPAG          ;GO TOIT
07470 ;
07480 NXTPAG  LD      HL,MAXPAG       ;HL-> MAXPAGE #
07490         LD      A,(PARM1)       ;A= CURR #
07500         CP      (HL)            ;TEST
07510         JR      Z,PAGE0         ;IF AT MAX
07520         INC     A               ;ELSE BUMP #
07530         JR      SAVPAG          ;DO IT
07540 ;
07550 DSPPAG  SLA     A
07560         SLA     A               ;A= PAGE*4
07570         LD      C,A             ;SAVE IT
07580         SLA     A
07590         SLA     A               ;A= PAGE*16
07600         ADD     A,C             ;A= PAGE*20
07610         LD      C,A             ;INTO C
07620         LD      B,00H           ;BC= REC #
07630         LD      DE.FCB1         ;DE-> FCB
07640         LD      A,POSN          ;@POSN SVC
07650         RST     28H
07660         RET     NZ
07670         CALL    CLRVID
07680         LD      B,20            ;20 LINES/PAGE
07690 PAGLP   PUSH    BC              ;SAVE LINE COUNT
07700         LD      DE,FCB1
07710         LD      A,READ          ;@READ
07720         LD      HL,LINBUF       ;80 CHAR BUFFER
07730         PUSH    HL              ;SAVE IT
07740         RST     28H
07750         POP     HL              ;HL-> LINE START
07760         LD      A,0DH           ;C/R
07770         LD      (LINBUF+79),A   ;FLAG END OF LINE
07780         LD      A,DSPLY         ;@DSPLY
07790         RST     28H
07800         POP     BC              ;GET COUNT
07810         DJNZ    PAGLP
07820         RET                     ;ELSE DONE
07830 ;
07840 ;
07850 BANKS   CALL    SKPSPL+1        ;GET NEXT & SKIP " "
07860         SUB     31H             ;CONVERT TO BINARY
07870         JR      C,BNKERR
07880         CP      03H             ;>"2"?
07890         JR      NC,BNKERR
07900         LD      C,A             ;GET BANK # IN C
07910         PUSH    BC              ;SAVE IT
07920         LD      B,02            ;BANK AVAIL?
07930         LD      A,BANK          ;@BANK SVC
07940         RST     28H
07950         POP     BC              ;FIX STACK/GET BANK#
07960         JR      NZ,BNKERR       ;IF ERROR
07970         PUSH    BC              ;SAVE IT
07980 ;
07990         LD      HL,CSRET        ;HL-> RET ADDR
08000         PUSH    HL              ;SET SAME
08010         LD      A,1BH           ;<SHFT><UP>
08020         JP      39BEH           ;GO DO IT
08030 ;
08040 CSRET   LD      (5ECCH),DE      ;SAVE CURRENT LOC
08050         CALL    BNK23           ;SET PTRS
08060         POP     BC              ;GET BANK #
08070         LD      A,C             ;INTO A
08080         PUSH    BC              ;SAVE AGAIN
08090         OR      A               ;TEST IT
08100         CALL    Z,BNKONE        ;IF Z, SPECIAL RESET
08110         LD      A,(CURBNK)      ;GET CURRENT BANK
08120         CALL    FNDBNK          ;HL-> STRG AREA
08130 SVBNKL  LD      E,(IX+0)
08140         LD      D,(IX+1)        ;DE-> ADDR TO MOVE
08150         CALL    MOVPTR          ;MOVE IT
08160         DJNZ    SVBNKL          ;LOOP
08170         POP     BC              ;GET REQUESTED BANK#
08180         PUSH    BC              ;SAVE AGAIN
08190         LD      B,0             ;SET SELECT FUNCTION
08200         LD      A,BANK          ;@BANK SVC
08210         RST     28H             ;SET NEW BANK
08220         LD      B,1             ;RESET "OLD" BANK
```

Listing 1 continued

See List of Advertisers on Page 145

Listing 1 continued

```
08230          LD      A,BANK
08240          RST     28H
08250          POP     BC              ;GET "NEW" BANK #
08260          LD      A,C             ;A = BANK#
08270          LD      (CURBNK),A      ;SAVE AS CURRENT
08280          CALL    FNDBNK          ;HL-> STRG AREA
08290          EX      DE,HL           ;DE-> STRG AREA
08300 LDBNKL   LD      L,(IX+0)
08310          LD      H,(IX+1)        ;HL-> ADDR TO MOVE
08320          CALL    MOVPTR
08330          DJNZ    LDBNKL          ;LOOP
08340          CALL    WRTVID          ;RELOAD VIDEO IMAGE
08350          JP      GOBACK          ;BACK TO SCRIPSIT
08360 ;
08370 ;
08380 BNKERR   CALL    BOTLIN          ;POSITION CURSOR
08390          LD      HL,BNKMSG       ;HL-> BANK NOT AVAIL
08400          LD      A,DSPLY
08410          RST     28H
08420          JP      WAITBK
08430 ;
08440 ;
08450 MOVPTR   LD      A,(DE)          ;GET LSB STR/CURR
08460          LD      (HL),A          ;SAVE IT
08470          INC     HL
08480          INC     DE              ;BUMP PTRS
08490          LD      A,(DE)
08500          LD      (HL),A          ;MOVE MSB
08510          INC     HL
08520          INC     DE
08530          INC     IX
08540          INC     IX              ;BUMP PTRS
08550          RET                     ;DONE
08560 ;
08570 ;
08580 FNDBNK   LD      HL,SVPTR1       ;HL->START OF STORE
08590          LD      IX,PTRTAB       ;IX-> RELO ADDRS
08600          LD      DE,22           ;22 BYTES/AREA
08610          LD      B,0BH           ;11 WORDS TO MOVE
08620 FNDLP    DEC     A               ;TEST A
08630          RET     M               ;M= FOUND
08640          ADD     HL,DE           ;HL-> NEXT
08650          JR      FNDLP           ;LOOP
08660 ;
08670 ;
08680 BNKONE   LD      DE,LAST+1       ;DE-> BANK1 BUFF
08690          JR      SETPTR
08700 BNK23    LD      DE,8001H        ;DE-> BANK2&3 BUFF
08710 SETPTR   LD      B,11
08720          LD      IX,STRTAB       ;IX-> ADDR TABLE
08730          CALL    RSTPTR          ;DO 11 ADDRS
08740          DEC     DE              ;ONE BACK
08750          LD      B,2             ;DO 2 MORE
08760 RSTPTR   LD      L,(IX+0)
08770          LD      H,(IX+1)        ;HL-> LOC TO ALTER
08780          LD      (HL),E          ;STORE LSB
08790          INC     HL
08800          LD      (HL),D          ;STORE MSB
08810          INC     IX
08820          INC     IX              ;IX-> NEXT ADDR
08830          DJNZ    RSTPTR
08840          RET                     ;DONE
08850 ;
08860 ;
08870 SKPSPL   INC     HL              ;BUMP PTR
08880          LD      A,(HL)          ;GET CHAR
08890 SKPSPC   CP      ' '             ;SPACE ?
08900          JR      Z,SKPSPL        ;SKIP SPACES
08910          CP      0DH             ;C/R?
08920          RET                     ;ELSE GO BACK
08930 ;
08940 ;
08950 EXIT1    LD      B,0             ;SELECT BANK FUNC
08960          LD      C,0             ;BANK # 0
08970          LD      A,BANK          ;@BANK SVC
08980          RST     28H
08990          LD      HL,0000H        ;NO ERROR
09000          LD      A,EXIT          ;@EXIT SVC
09010          RST     28H             ;DO IT
09020 ;
09030 ;
09040 ;
09050 CURBNK   DEFB    $-$
09060 ;
09070 PTRTAB   DEFW    5EDEH           ;A040
09080          DEFW    5EE6H           ;A048
09090          DEFW    5EB6H           ;AN92
09100          DEFW    5ECCH           ;A022
09110          DEFW    5EDCH           ;A038
09120          DEFW    5EE0H           ;A042
09130          DEFW    5EB4H           ;AN90
09140          DEFW    5EE2H
09150          DEFW    5EE4H           ;
09160          DEFW    5EC0H
09170          DEFW    5EEDH
09180 ;
09190 ;
09200 STRTAB   DEFW    3945H
09210          DEFW    3AF4H
09220          DEFW    3C16H
09230          DEFW    40A6H
09240          DEFW    40DEH
09250          DEFW    4532H
09260          DEFW    4B6BH
09270          DEFW    4E3EH
09280          DEFW    52B0H
09290          DEFW    58A9H
09300          DEFW    58B7H
09310 ;
09320          DEFW    396EH
09330          DEFW    42EEH
09340 ;
09350 ;
09360 SVPTR1   DEFW    $-$
09370          DEFW    $-$
09380          DEFW    $-$
09390          DEFW    $-$
09400          DEFW    $-$
09410          DEFW    $-$
09420          DEFW    $-$
09430          DEFW    $-$
09440          DEFW    $-$
09450          DEFW    $-$
09460          DEFW    $-$
09470 ;
09480 SVPTR2   DEFW    0FFFFH
09490          DEFW    0FFFFH
09500          DEFW    0FFFEH
09510          DEFW    8001H
09520          DEFW    8001H
09530          DEFW    8001H
09540          DEFW    8001H
09550          DEFW    0FFFFH
09560          DEFW    8001H
09570          DEFW    0005H
09580          DEFW    0044H
09590 ;
09600 SVPTR3   DEFW    0FFFFH
09610          DEFW    0FFFFH
09620          DEFW    0FFFEH
09630          DEFW    8001H
09640          DEFW    8001H
09650          DEFW    8001H
09660          DEFW    8001H
09670          DEFW    0FFFFH
09680          DEFW    8001H
```

Listing 1 continued

Listing 1 continued

```
        DEFW    0005H
        DEFW    0044H
;
;
;*** MESSAGES ***
;
CLRMSG  DEFB    1CH             ;HOME CURSOR
        DEFB    1FH             ;CLEAR TO END OF SCREEN
        DEFB    03H             ;ASCII ETX
;
BLKLIN  DEFB    0AH
        DEFB    0DH             ;BLANK LINE
;
;VDMSG  DEFM    'Hit <ENTER> to continue...'
;       DEFB    0EH             ;CURSOR ON
;       DEFB    03H
;
;DRVMSG DEFB    0C2H            ;3 SPACES
;       DEFM    'Drive: '
;       DEFB    03H
;
;NAMSG  DEFB    0C3H
;       DEFM    'Pack ID: '
;       DEFB    03H
;
;CREMSG DEFB    0C3H
;       DEFM    'Created: '
;       DEFB    03H
;
;FREMSG DEFB    0C3H
;       DEFM    'Free: '
;       DEFB    03H
;
;EXTMSG DEFM    '$$$'
;
;HLPMSG DEFM    'Use up/down arrows to page, <BREAK> to
return to ScripAid...'
;       DEFB    0EH
;       DEFB    03H
;
;HLPFS  DEFM    'ScripAid/HLP'
;       DEFB    0DH
;
;BNKMSG DEFM    'Bank NOT available'
;       DEFB    03H
;
;LOOKTB DEFM    '1!2"3#4$5%6&7'
;       DEFB    27H
;       DEFM    '8(9)'
;
*LIST OFF
VDMSG   DEFM    'Hit <ENTER> to continue...'
        DEFB    0EH
        DEFB    03H
DRVMSG  DEFB    0C2H
        DEFM    'Drive: '
        DEFB    03H
NAMSG   DEFB    0C3H
        DEFM    'Pack ID: '
        DEFB    03H
CREMSG  DEFB    0C3H
        DEFM    'Created: '
        DEFB    03H
FREMSG  DEFB    0C3H
        DEFM    'Free: '
        DEFB    03H
EXTMSG  DEFM    '$$$'
HLPMSG  DEFM    'Use up/down arrows to page, <BREAK> to
return to ScripAid...'
        DEFB    0EH
        DEFB    03H
HLPFS   DEFM    'ScripAid/HLP'
        DEFB    0DH
BNKMSG  DEFM    'Bank NOT available'
        DEFB    03H
LOOKTB  DEFM    '1!2"3#4$5%6&7'
        DEFB    27H
        DEFM    '8(9)'
*LIST ON
;
;
CTLTAB  DEFS    90              ;90 BYTES FOR CTL CODES
;
LINBUF  DEFS    80              ;80 BYTE HELP LINE BUFFER
;
EXTBUF  DEFS    3               ;3 BYTE EXTENSION
;
FCB1    DEFS    32              ;FCB FOR HELP
;
PHLPFS  DEFS    14              ;HOLDS *PR DRIVER FS
;
PARM1   DEFB    $-$             ;HOLD HELP PAGE
MAXPAG  DEFB    $-$             ;MAX HELP PAGE
;
LAST    DEFL    $               ;LAST BYTE USED
;
;
        END     ENTRY
```

End

Program Listing 2. Rekey/SRC, the source code for keyboard and message modifications.

```
;********************** REKEY ****************************
;
;    Mnemonic keyboard overlay for Model 4 SCRIPSIT
;
;*******************************************************
;
;       *** Patches for keyboard commands ***
;
;       <CLR><X>-> <CLR><L>, Line - Insert routine
        ORG     3D79H
        DEFB    0CCH
;
;       <CLR><Q>-> <CLR><B>, Block - Insert routine
        ORG     3D7DH
        DEFB    0C2H
;
;       <CLR><W>-> <CLR><Z>, Window - Main Commands
;       <CLR><Q>-> <CLR><B>, Block
        ORG     5E28H
        DEFB    0DAH
        DEFB    0C2H
;
;       <CLR><S>-> <CLR><I>, Insert - Main Commands
        ORG     5E2DH
        DEFB    0C9H
;
;       <CLR><X>-> <CLR><L>, Line - Main Commands
;       <CLR><C>-> <CLR><G>, ParaGraph
;       <CLR><V>-> <CLR><N>, New page
        ORG     5E30H
        DEFB    0CCH
        DEFB    0C7H
        DEFB    0CEH
;
;       <CLR><F>-> <CLR><S>, Spaces - Delete routine
;       <CLR><X>-> <CLR><L>, Line
;       <CLR><C>-> <CLR>[illegible] ParaGraph
        ORG     5E69H
        DEFB    0D3H
        DEFB    0CCH
        DEFB    0C7H
;
;       <CLR><Z>-> <CLR><W>, Word - Delete routine
```

Listing 2 continued

```
        ORG     5E6DH
        DEFB    0D7H
;
;       <CLR><Q>-> <CLR><B>, Block - Exchange routine
;       <CLR><Z>-> <CLR><W>, Word
;       <CLR><C>-> <CLR><G>, ParaGraph
        ORG     5E95H
        DEFB    0C2H
        DEFB    0D7H
        DEFB    0C7H
;
;
;
;       *** Patches to default values ***
;       Patch Video Width to 68 characters
        ORG     3975H
        DEFB    68
;
;       Patch left & right margins to 6 & 74
        ORG     5E9FH
        DEFB    6
        DEFB    74
;
;       Patch justification = no
        ORG     5EA2H
        DEFB    'N'
;
;
;       *** Patch Sign-on Banner ***
        ORG     3EFBH
        DEFB    78              ;LENGTH OF MESSAGE
        DEFM    ' * ScripAid * - Model 4 Version 2.1 -
(c) 1984 by Paladin Software - C. Knight'
;
;
;
;       *** Patches to system messages ***
;
        ORG     3F4CH
        DEFM    'nsert command mode'
        ORG     438FH
        DEFM    'lock command mode'
        ORG     43A2H
        DEFM    'elete command mode'
        ORG     43B6H
        DEFM    'xchange command mode'
        ORG     43CCH
        DEFM    'elete or Unmark block (D or U)?'
        ORG     43EEH
        DEFM    'dd, Even or Standard (O, E, or S)?'
        ORG     4413H
        DEFM    'ame of block to insert?'
        ORG     442DH
        DEFM    'ame of block to exchange?'
        ORG     4E4EH
        DEFM    'ound'
        ORG     4E55H
        DEFM    'eleted'
        ORG     4E5EH
        DEFM    'eplaced'
        ORG     4E68H
        DEFM    'annot find'
        ORG     4E75H
        DEFM    'haracters free:'
        ORG     4E87H
        DEFM    'ocument length:'
        ORG     4E99H
        DEFM    'ideo line width:'
        ORG     4EACH
        DEFM    'aragraph indent:'
        ORG     4EBFH
        DEFM    'ursor line number:'
        ORG     4ED4H
        DEFM    'urrent file name:'
        ORG     4EE8H
        DEFM    'ocument has NOT been named'
        ORG     4F04H
        DEFM    'pecial command?'
        ORG     4F16H
        DEFM    'ress ENTER to continue'
        ORG     4F2EH
        DEFM    'ot zone'
        ORG     4F40H
        DEFM    'yphenation complete'
        ORG     5445H
        DEFM    'ab command mode'
        ORG     5456H
        DEFM    'indow command mode'
        ORG     546AH
        DEFM    'nter repeat command'
        ORG     547FH
        DEFM    'epeat how many times'
        ORG     5497H
        DEFM    'elete paragraph (Y or'
        ORG     54B3H
        DEFM    'elete to end ot text (Y or'
        ORG     58E3H
        DEFM    'esting for errors'
        ORG     58F6H
        DEFM    'ress ENTER to print next page'
        ORG     5CA5H
        DEFM    'nvalid command'
        ORG     5CB5H
        DEFM    'o more room'
        ORG     5CC2H
        DEFM    'ursor within block'
        ORG     5CD6H
        DEFM    'lock NOT found'
        ORG     5CE6H
        DEFM    'arker error'
        ORG     5CF3H
        DEFM    'ine too long'
        ORG     5D01H
        DEFM    'age number format error'
        ORG     5D1AH
        DEFM    'age number overflow'
        ORG     5D35H
        DEFM    'interface NOT ready'
        ORG     5D4AH
        DEFM    'argin format error'
        ORG     5D5EH
        DEFM    'ormat line error'
        ORG     5D70H
        DEFM    'eader or footer too long'
        ORG     5D8AH
        DEFM    'ine printer NOT ready'
        ORG     5DA3H
        DEFM    'search key'
        ORG     5DB0H
        DEFM    ' replacement text'
        ORG     5DC3H
        DEFM    'ape loading error'
        ORG     5DD8H
        DEFM    'document file name'
        ORG     5DECH
        DEFM    'age formatting error'
;
;
        END
```

End

Program Listing 3. JCL produces a JCL file for ScripAid.

```
BUILD ScripAid/JCL:1
MEMORY(CLEAR)
LOAD SCRIPSIT/CMD
LOAD REKEY/OVL
LOAD ENHANCE/OVL
DUMP ScripAid/CMD:1 (S=X'3800',E=X'6970',T=X'6810')
```

End

Program Listing 4. Driver, a Basic program to create a printer driver.

```
10 CLS:DEFINT A-Z:DIM PRTCTL$(17),DEFCTL$(17):CLR$=CHR$(31):GOTO 60000
100 SEL$=INKEY$:IF SEL$="" THEN 100
110 IF INSTR(VERF$,SEL$)=0 THEN SOUND 2,0:GOTO 100 ELSE RETURN
200 SEL$=INKEY$:IF SEL$="" THEN 200 ELSE IF SEL$=CHR$(17) THEN GTSTR$=SEL$:RETUR
N ELSE IF SEL$=CHR$(13) THEN RETURN
205 IF SEL$=CHR$(8) AND LEN(GTSTR$)>0 THEN PRINT SEL$;:GTSTR$=LEFT$(GTSTR$,LEN(G
TSTR$)-1):GOTO 200
220 IF SEL$>=" " AND LEN(GTSTR$)<FLD THEN PRINT SEL$;:GTSTR$=GTSTR$+SEL$:GOTO 20
0 ELSE 200
500 PRINT TAB(40-LEN(HEAD$)/2)HEAD$:RETURN
10000 PRINT@(19,0),CLR$;:CCODE$=FNIV$(" \"+MID$(CTL$,CTLNUM,1)+" ")
10010 PRINT TAB(8)"enter decimal codes for embedded command "CCODE$
10020 FOR J=0 TO 4
10030 PRINT@(21,J*15+1),CLR$"code "FNIV$(STR$(J+1)+" ")" ";
10040 INPUT CD$:IF CD$="" THEN CD=255:GOTO 11000
10050 CD=VAL(CD$):IF NOT(CD<0 OR CD>254) THEN 11000
10060 PRINT@(23,10),FNIV$(" ***** ILLEGAL VALUE ***** ")+"  <ENTER> to continue.
..";:INPUT EN$:GOTO 10030
11000 MID$(PRTCTL$(CTLNUM-1),J+1,1)=CHR$(CD):IF CD=255 THEN J=J+1:IF J<5 THEN 11
000
11010 NEXT J:Y=INT((CTLNUM)/2+.5)+7:X=ABS(CTLNUM MOD 2-1)*33+14
11020 PRINT@(Y,X),;:FOR J=1 TO 5
11030 CODE=ASC(MID$(PRTCTL$(CTLNUM-1),J,1))
11040 IF CODE=255 THEN PRINT"    "; ELSE PRINT USING"####";CODE;
11050 NEXT J
11060 PRINT@(19,0),CLR$;:PRINT TAB(6)"current HELP screen entry for "CCODE$" is:
"
11070 PRINT TAB(5);:GET 3,CTLNUM+2:PRINT LEFT$(MENU$,70)
11080 PRINT@(21,5),"enter new HELP line or <ENTER> for no change...":SEL$=""
11090 PRINT@(22,0),CLR$;:PRINT TAB(5)CCODE$" = ";:GTSTR$="":FLD=54:GOSUB 200
11100 IF GTSTR$<>"" THEN LSET MENU$="   \"+MID$(CTL$,CTLNUM,1)+" = "+GTSTR$:PUT
3,CTLNUM+2:SEL$=INKEY$:GOTO 50090 ELSE 50090
20000 PUT 1,1:GOTO 52000
30000 FSPEC$="default/pdc"+DRV$:OPEN"r",2,FSPEC$,90
30010 FOR J=0 TO 17:FIELD 2,5*J AS DUMMY2$,5 AS DEFCTL$(J):LSET DEFCTL$(J)=PRTCT
L$(J):NEXT
30020 PUT 2,1:CLOSE 2:MSPEC$="default/hlp"+DRV$:OPEN"r",2,MSPEC$,80:FIELD 2,80 A
S DEFMENU$
30030 FOR J=1 TO 20:GET 3,J:LSET DEFMENU$=MENU$:PUT 2,J:NEXT:GOTO 52000
50000 PRINT@(6,0),CLR$;
50010 HEAD$="driver ID - "+FNIV$(" "+FSPEC$+" ")+"              the following code
s are assigned...":GOSUB 500
50020 GET 1,1
50030 PRINT@(8,0),CLR$;
50040 FOR J=1 TO 18 STEP 2:FOR K=0 TO 1
50050 PRINT TAB(9+K*33)FNIV$(" \"+MID$(CTL$,J+K,1)+" ")" =";
50060 FOR L=1 TO 5:CODE=ASC(MID$(PRTCTL$(J-1+K),L,1))
50070 IF CODE=255 THEN L=5 ELSE PRINT USING"####";CODE;
50080 NEXT L:NEXT K:PRINT:NEXT J
50090 PRINT@(19,0),CLR$;:PRINT TAB(8)FNIV$(" <key> to alter         <S> to save
file       <A> to abort ")
50100 PRINT@(21,10),"enter selection...";:VERF$=CTL$+"SsAa":GOSUB 100
50110 CTLNUM=INSTR(CTL$,SEL$):IF CTLNUM>0 THEN 10000 ELSE PRINT@(6,0),CLR$;
51000 IF SEL$="A"  OR SEL$="a" THEN SAV=0:GOTO 52000 ELSE SAV=-1
51010 PRINT@(8,12),CLR$"save as..."
51020 PRINT:PRINT TAB(30)FNIV$("<1>")" ----- "FSPEC$
51030 PRINT:PRINT TAB(30)FNIV$("<2>")" ----- DEFAULT/PDC"+DRV$
51035 PRINT:PRINT:PRINT TAB(30)FNIV$("<A>")" ----- to abandon file"
51040 PRINT:PRINT TAB(13)"enter selection...";:VERF$="12aA":GOSUB 100
51050 IF SEL$="A" OR SEL$="a" THEN SAV=0:GOTO 52000
51060 ON VAL(SEL$) GOTO 20000,30000
52000 CLOSE:PRINT@(6,0),CLR$;:PRINT@(11,20),;
52010 IF SAV THEN PRINT"file saved as "FNIV$(" "+FSPEC$+" ");:GOTO 53000
52020 PRINT"changes to "FNIV$(" "+FSPEC$+" ")" abandoned...";
53000 FOR J=0 TO 5000:NEXT:GOTO 60010
60000 CLS:GOSUB 61000
60010 PRINT @(8,8),CLR$"processing options...":PRINT:PRINT TAB(26)FNIV$(" filesp
ec ")" to edit/create":PRINT:PRINT TAB(26)FNIV$(" <ENTER>  ")" to edit "FNIV$("
DEFAULT/PDC "):PRINT:PRINT TAB(26)FNIV$(" <CTL><Q> ")" to end program":PRINT
60015 GTSTR$="":PRINT TAB(9)"enter selection...";:FLD=24:GOSUB 200:FSPEC$=GTSTR$
:IF FSPEC$="" THEN FSPEC$="DEFAULT"
60020 EXT=INSTR(FSPEC$,"/"):DRV=INSTR(FSPEC$,":")
60025 IF FSPEC$=CHR$(17) THEN 65000
60030 IF DRV>0 THEN DRV$=MID$(FSPEC$,DRV,2):FSPEC$=LEFT$(FSPEC$,DRV-1)
60040 IF EXT>0 THEN FSPEC$=LEFT$(FSPEC$,EXT-1)
60050 MSPEC$=FSPEC$+"/HLP"+DRV$:FSPEC$=FSPEC$+"/PDC"+DRV$:ON ERROR GOTO 62000
60060 LOD$="loading ":OPEN"i",1,FSPEC$:CLOSE
60070 PRINT@(6,0),CLR$:PRINT@(10,25),LOD$;FNIV$(FSPEC$)"...";:OPEN"r",1,FSPEC$,9
0:OPEN"r",3,MSPEC$,80:FIELD 3,80 AS MENU$
60080 FOR J=0 TO 17:FIELD 1,5*J AS DUMMY$,5 AS PRTCTL$(J):NEXT
60090 IF NEWFILE THEN FOR J=0 TO 17:LSET PRTCTL$(J)=STRING$(5,255):NEXT:PUT 1,1:
FOR J=1 TO 2:LSET MENU$=STRING$(80,20):PUT 3,J:NEXT:FOR J=3 TO 20:LSET MENU$="
 \"+MID$(CTL$,J-2,1)+" = "+STRING$(72,20):PUT 3,J:NEXT
60100 GOTO 50000
61000 SYSTEM"system(blink=95)":EQU$=STRING$(80,"="):HYPH$=STRING$(80,"-")
61010 INV$=CHR$(16):NOV$=CHR$(17):DEF FNIV$(Z$)=INV$+Z$+NOV$
61020 DEF FNCLR(Y,X)=PRINT @(X,Y),CHR$(31);
61030 CTL$="1!2"+CHR$(34)+"3#4$5%6&7'8(9)":GOTO 63000
62000 IF ERR=53 THEN NEWFILE=-1:LOD$="creating ":RESUME NEXT ELSE PRINT@(8,0),CL
R$;
62010 PRINT@(11,20),;:HEAD$=FNIV$("***** FATAL ERROR *****"):GOSUB 500
62020 PRINT:PRINT TAB(8)"job aborted...............":CLOSE:END
63000 CLS:HEAD$="          "+FNIV$(" ScripAid/CMD ")+" copyright 1894 by "+FNIV$("
 Paladin Software ")
63010 PRINT EQU$;:GOSUB 500:PRINT EQU$;:PRINT@(1,0),"|";:PRINT@(1,79),"|";
63020 PRINT@(3,0),"|";:HEAD$="Printer Driver Customization module":GOSUB 500
63030 PRINT HYPH$:PRINT@(3,79),"|":RETURN
65000 PRINT@(6,0),CLR$;:PRINT@(8,12),"return to..."
65010 PRINT:PRINT TAB(32)FNIV$("<1>")" ----- BASIC"
65020 PRINT:PRINT TAB(32)FNIV$("<2>")" ----- TRSDOS"
65030 PRINT:PRINT TAB(13)"enter selection... ";:VERF$="12":GOSUB 100
65040 IF SEL$="1" THEN CLS:END ELSE CLS:SYSTEM
65050 CLS:SYSTEM
```

End

Program Listing 5. Helper, a Basic program to create help file.

```
5 SYSTEM"system(blink=95)"
10 CLS:OPEN"r",1,"ScripAid/hlp:1",80:FIELD 1,79 AS H$,1 AS CR$
20 LSET CR$=CHR$(13):INPUT "Page ";PAG:PAG=PAG-1
30 LSET H$="     ***** ScripAid *****    by Paladin Software    (c) 1984   HELP
Page"+STR$(PAG+1):PUT 1,1+PAG*20:CLS:PRINT LEFT$(H$,79)
40 LSET H$=STRING$(79,32):PRINT:PUT 1,2+PAG*20
50 FOR J=3 TO 20:GET 1,J+PAG*20:PRINT LEFT$(H$,79):NEXT
60 PRINT@(22,3),CHR$(16)" Enter line to edit (3-20), <C> to clear page, <S> to s
ave page "CHR$(17)CHR$(30);:X$="":GOSUB 100:SEL$=X$
70 LIN=VAL(SEL$):IF LIN=0 THEN 1000
80 IF LIN<3 OR LIN>20 THEN 60 ELSE PRINT@(LIN-1,0),;:X$="":GOSUB 100:HLP$=X$
90 LSET H$=HLP$:PUT 1,LIN+PAG*20:GOTO 60
100 A$=INKEY$:IF A$="" THEN 100 ELSE IF A$=CHR$(13) THEN RETURN
110 IF A$=CHR$(8) AND LEN(X$)>0 THEN PRINT A$;:X$=LEFT$(X$,LEN(X$)-1):GOTO 100
120 IF A$>CHR$(31) THEN PRINT A$;:X$=X$+A$:GOTO 100
1000 IF SEL$="C" OR SEL$="c" THEN FOR J=3 TO 20:LSET H$=STRING$(79,32):LSET CR$=
CHR$(13):PUT 1,J+PAG*20:NEXT:GOTO 30
1010 IF SEL$="S" OR SEL$="s" THEN PAG=PAG+1:GOTO 30
1020 IF SEL$="" THEN CLOSE:CLS
```

End

Illustration by John Pirman

NovaCalc

Whether you're making five-year fiscal projections or managing your home budget, NovaCalc tracks your finances with all the features of a commercial spreadsheet.

by Daniel Garms

Forget your balance sheets and home-budget programs. Here's a full-featured, menu-driven Model I/III spreadsheet that gives you all the capabilities of its commercial counterparts, including a rapid-entry mode that lets you enter information into the spreadsheet without waiting for the program to recalculate related values.

NovaCalc also totals and averages a series of cells and has a built-in four-function calculator that lets you make calculations before entering the results in the spreadsheet. It accepts label (text), numeric, and formula entries of up to 12 characters.

Operating NovaCalc

When you run NovaCalc, it prompts you to select 1-, 2-, 3-, or 4-digit precision and to choose from four spreadsheet formats (24 columns by 26 rows, 16 by 39, 12 by 52, or 8 by 78). NovaCalc then displays the spreadsheet with a command prompt and cell status line at the top of the screen.

NovaCalc provides a total of 624 cells. You identify cells by column and row. For example, A01 represents the cell at column A, row 1. Use the arrow keys to move the cell pointer (cursor) from cell to cell.

The slash key (/) puts NovaCalc in the command mode. Press this key and a list of commands appears at the screen top. NovaCalc supports 13 commands (see Table 1). You invoke the desired command by pressing its first character.

The clear key is the escape character. Press it to cancel the current routine and return to the spreadsheet. When used in the rapid-entry mode, the clear key cancels the current routine and updates the spreadsheet for all previous entries made in this mode.

The rapid-entry mode works when you enter data or use the calculator routine. Hit any arrow key instead of the enter key to make another entry without recalculating all the affected cells.

The contents of the cell at the cursor position appear on the cell status line as you make an entry. To recalculate the spreadsheet after an entry, hit the enter key.

Pressing the clear key or executing any command except Alpha, Value, or Calculator (described below) exits the rapid-entry mode and recalculates the spreadsheet. The program also exits the mode if the cursor scrolls to a column or row that's off-screen.

Labeling Cells

You use the Alpha command (A) to label a cell. When you invoke this command, the cell status line displays a lowercase (a) at the top of the screen and in the cell. However, this simply serves as an indicator; you still have all 12 character spaces available for entering text.

Type in your entry at the prompt and press the enter key to update the spreadsheet or the clear key to cancel the entry.

You can enter any combination of alphanumeric characters as a label except an open parenthesis or quotation marks. NovaCalc's value interpretation routine uses an open parenthesis to detect alpha entries. Basic uses quotation marks as delimiters when separating and storing string variables, and NovaCalc saves all variables as strings.

You can't combine cells with label entries in mathematical functions, but you can use label cells in a column or row without affecting the total. The program's value interpretation routine sets the value of any alpha or blank cell to zero.

Blanking Out Cells

Blank (B) deletes the contents of a cell. It erases cell formulas and values and updates the spreadsheet.

If the cell you're blanking out derives from a formula in another cell, that formula remains unaffected, but the value of the blanked cell equals zero.

Copying Values

The Copy command (C) copies a cell's value to other cells so you don't have to reenter information. Copy can also link cells so that NovaCalc updates them whenever the source cell changes.

To link cells, type Y at the link prompt, or type N to copy without linking. Once linked, the destination cells contain a value like A01C, for example. A01 represents the source cell and C instructs the value interpretation routine to get the spreadsheet value from the source cell.

Command	Description
Alpha	Alpha command for entering label data
Blank	Blank command clears cell and deletes formula used to create it
Copy	Copy command copies value of one cell to another cell
Delete	Delete command erases cell formula while retaining cell's value
Go To	GOTO command positions cursor at specified cell location
Help	Help command displays explanations for all NovaCalc commands
Load	Load command loads saved spreadsheets
Print	Prints out the portion of the spreadsheet on the screen
Quit	Quit command ends session and prompts you to save data
Replicate	Replicate command projects and modifies over a series of cells
Save	Save command saves data to disk
Value	Value command for entering formulas and values
Calculate (=)	Activates four-function calculator

Table 1. Commands for NovaCalc.

You can't link alpha cells; the program unlinks any cell containing an alpha value prior to copying it to a new location.

The next prompt asks you if you want the cell copied to another cell location (C), or copied several locations down (D) or right (R).

The C option displays the current cursor location followed by an equals sign. You enter the target location. For example, entering A02 at this prompt copies the current cell to the second row of column A and updates the spreadsheet.

When you copy down or right, the program asks you how many times you want to copy the cell. Enter a number and hit the enter key. The program copies the value of the current cell location down or right the specified number of times and updates the screen.

Deleting Formulas

Delete (D) erases the formula used to calculate a cell without removing the cell's current value. It's useful for unlinking cells created with the Copy command. Press the D key to delete the formula and update the spreadsheet.

The GOTO Command

The GOTO command (G) lets you move the cursor to any cell by entering its location. This saves moving through the spreadsheet with the arrow keys. GOTO is especially useful when you want to move to a cell that's off-screen, or when you want to print a specific area of the spreadsheet.

GOTO prompts you for a cell location, then makes that location the upper left-hand cell on the display. For example, typing H, 1, and 7 resets the display to show columns H, I, J, and K and rows 17–29. If the cell specified is too far down to display a full screen, NovaCalc doesn't print the cell in the upper left-hand corner; instead, it marks the position with the cursor and displays a full screen.

Press the clear key to cancel GOTO and recalculate the spreadsheet. If you specify a cell location outside the limits of the spreadsheet, the program displays an "Entry Error" message and stays at the current cell location.

The Help Command

The Help command (H) lists NovaCalc's commands and gives you a short definition of each. Press the clear key to return to the spreadsheet.

Loading a Spreadsheet

To load files from disk, press the L key. Enter the appropriate file number (1–9) at the prompt. The message "Loading" appears on the screen as the file loads.

When the program finishes loading, it recalculates the spreadsheet and resets the cursor position to cell A01.

NovaCalc saves spreadsheet cells in an order dependent on the number of rows and columns. Before loading a file, be sure to choose the appropriate spreadsheet format. Otherwise, cells can reference formulas that don't exist.

Printing Spreadsheets

The Print command (P) prints out the portion of the spreadsheet on the display (four rows and 13 columns). You can print the values or the formulas.

Print prompts you to hit the enter key when the printer is ready. To cancel the Print command, press the clear key.

The program next asks you to press the D key to print display values, or the F key to print formulas. Press the appropriate key and hit the enter key. The program displays a "Printing" message and prints the spreadsheet.

Ending a Session

To end a session or restart the program, press the Q key. Quit prompts you to save data before ending the session.

The program then branches to the Save routine or prompts you to return to the spreadsheet (R), create a new spreadsheet (C), or exit to Basic ready (E).

The Replicate Command

The Replicate command projects and modifies a formula over a series of cells. This saves you from entering similar formulas in contiguous cells. Replicate works down or to the right.

For example, assume columns A, B, and C are labeled Income, Expenses, and Remaining Amount, and the formula for cell C01 is C01 = A01 – B01. If rows 1–12 represent January through December, and you want the same formula for each month, you must type C02 = A02 – B02, C03 = A03 – B03, and so on.

To do this automatically, type in R to enter the Replicate routine, then type D at the prompt for down, and 11 at the prompt for the number of times you want to replicate the formula.

If the number of rows or columns specified exceeds the number of remaining rows or columns in the spreadsheet, the program adjusts your entry before executing the command.

To replicate a formula to all remaining cells, enter any number exceeding the number of columns or rows in the spreadsheet. Press the clear key to cancel Replicate.

Saving Spreadsheets

The Save command (S) lets you save up to nine spreadsheet templates and data. The routine prompts you to save the file to a number from 1–9. Select a number or press the clear key to return to the command prompt.

NovaCalc displays a "Saving" message as it saves the spreadsheet to disk, then prompts you to return to the spreadsheet (R), create a new spreadsheet (N), or exit the program (E).

The Value Command

The Value command lets you enter numeric values or formulas. To make an entry, press the V key, type in the

The Key Box LOAD 80

Models I and III
48K RAM
Disk Basic
Printer (optional)

✓ See List of Advertisers on Page 145

numeric value or formula and hit the enter key.

NovaCalc accepts any positive or negative number value as numeric input. Input numeric values requiring a mathematical operator with the Calculator routine.

To input a numeric value, type in the value and hit any arrow key to continue or the enter key to recalculate.

NovaCalc formulas add, subtract, multiply, or divide two cells or a cell and a numeric value. To build a formula, move the cursor to the cell where you want the result to appear. The cell location appears at the screen top in the format A01 = .

Type in a cell location, a mathematical operator, and a numeric value, or a cell location. Now hit the enter key and the program calculates the value and puts it in the cell.

NovaCalc retains the formula used to calculate the cell and displays it on the cell status line when you position the cursor at the cell. The program updates the display for formula changes when it recalculates the spreadsheet.

In addition to the four mathematical operators listed above, NovaCalc also totals (T) or averages (A) a series of cells (see Table 2 for a list of mathematical functions).

To total a series of cells in the same column or row, enter a starting cell location, press the T key, enter an ending cell location, and hit the enter key.

The program adds the contents of the cell locations from the starting cell to the ending cell and puts the result at the cursor location.

To average a series of cells in the same column or row, enter a starting cell location, type A, enter an ending cell location, and hit the enter key.

NovaCalc averages all cells from the first to the second location and displays the results at the cursor location.

Using the Calculator Function

The Calculator command adds, subtracts, multiplies, and divides values. To use it, hit the equals key. The routine displays the prompt START WITH <C>ELL VALUE X OR <Z>ERO?

Press the C key to use the current cell value as the starting subtotal. Press the Z key to start with a subtotal of zero.

Calculator displays the message SUBTOTAL = X, where X is the value you specified above, and the prompt ENTER VALUE:.

Type in a value and press the appropriate mathematical function (+ , − , *, or /). Calculator performs the function on the value to create a new subtotal, and displays it at the screen top.

With a starting subtotal of zero, typing 1, 2, 3, and a plus sign adds 123 to zero for a new subtotal of 123 (the program ignores instructions to divide by zero).

When you're finished, hit the equals key to set the cell value to the new subtotal, or hit the clear key to exit the routine.

To use the rapid-entry mode with the Calculator routine, press any arrow key instead of the equals key. This places the value of the subtotal in the cell, but does not update the rest of the worksheet.

The Final Analysis

While NovaCalc is a full-featured spreadsheet, it lacks the speed of commercial programs. Occasionally, NovaCalc must do a dynamic string allocation. When this happens, the computer goes through memory to clean out the old variables and appears to hang up for several minutes.

However, this is a minor inconvenience when weighed against the cost of commercial spreadsheet programs. If it means saving $100 or more, most home applications can probably wait a few minutes for the same results. ■

Contact Daniel D. Garms at P.O. Box 3742, Reston, VA 22090.

Mathematical Operator	Definition
A	Averages a series of cells in the same column or row
T	Totals a series of cells in the same column or row
+	Adds second entry to first entry
−	Subtracts second entry from first entry
*	Multiplies first entry by second entry
/	Divides first entry by second entry

Table 2. Mathematical operators for NovaCalc.

Program Listing. NovaCalc spreadsheet program.

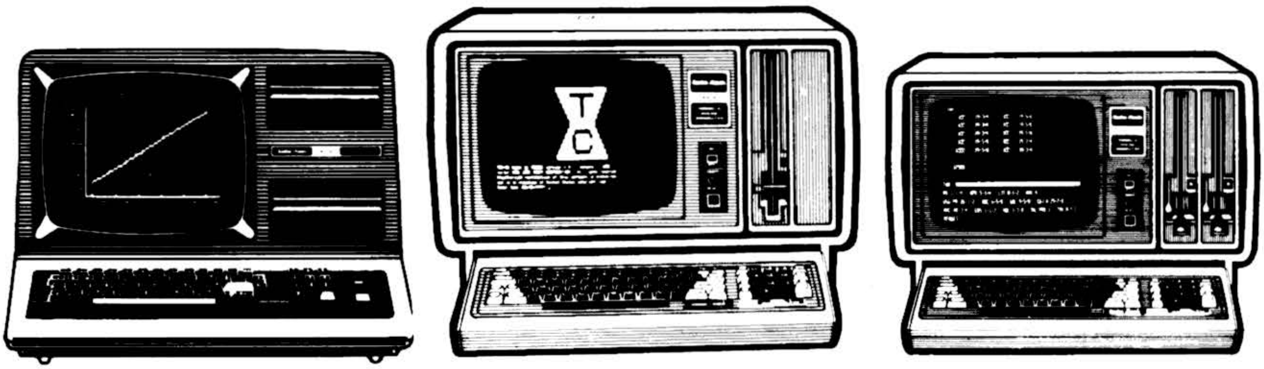
```
10 CLS:CLEAR150:A$=STRING$(128,179):PRINT@0,A$;:PRINT@267,"NovaC
alc Financial Package Spreadsheet 1.0";:PRINT@896,"Copyright 198
3 by Daniel D. Garms";:FORD=1TO1500:NEXT:CLEAR18000:N$=STRING$(6
4,32)
20 PRINT@896,"Set decimal precision to how many points? (1, 2, 3
, or 4) ";
30 I$=INKEY$:IFI$=""THEN30ELSEIFI$<"1"ORI$>"4"THEN30
40 N=VAL(I$):PRINTN;:T$=STRING$(N,48):I$="1":T$=I$+T$:N=VAL(T$)
50 PRINT@395,"Format spreadsheet as:":PRINT:PRINT,"1.  24 column
s by 26 rows":PRINT."2.  16 columns by 39 rows":PRINT,"3.  12 co
lumns by 52 rows":PRINT."4.   8 columns by 78 rows"
60 I$=INKEY$:IFI$=""THEN60ELSEIFI$<"1"ORI$>"4"THEN60
70 IFI$="1"THENH=24ELSEIFI$="2"THENH=16ELSEIFI$="3"THENH=12ELSEH
=8
80 U=624/H:DIMB$(U,H):DIMD$(U,H):A$="FILE T":CLS
90 REMSCREENSETUP
100 B=197:I=1:J=1:K=1:L=1:E=0
110 CLS
120 C=J:R=I:PRINT@0,N$;N$;STRING$(4,143);CHR$(191);STRING$(6,143
);" ";CHR$(J+64);" ";STRING$(11,143);" ";CHR$(J+65);" ";STRING$(
11,143);" ";CHR$(J+66);" ";STRING$(11,143);" ";CHR$(J+67);" ";ST
RING$(6,143);CHR$(191);CHR$(32);
130 R$=STR$(R):R$=RIGHT$(R$,2):IFR>9THEN150
140 R$=RIGHT$(R$,1):T$="0":R$=T$+R$
150 PRINTCHR$(32);R$;CHR$(32);CHR$(191);STRING$(57,32);CHR$(191)
;
160 R=R+1:IFR=I+13THEN180
170 PRINTCHR$(32);:GOTO130
180 REMPRINTCELLS
190 ONERRORGOTO2740:PRINT@0,N$;:PRINT@0,"NovaCalc Spreadsheet ";
A$;" (Press </> for commands)";
200 PRINT@64,N$;:PRINT@64,"* * * CALCULATING * * *";:C=J:R=I:P=1
98
210 PRINT@P,CHR$(32);:V$=MID$(B$(R,C),1,1):IFV$=>"A"ANDV$<CHR$(H
+65)THEN290ELSEIFV$=>"0"ANDV$=<"9"THEN250ELSEIFV$="+"THEN230ELSE
IFV$="-"THEN240ELSEIFV$=""THEN220ELSEIFV$="("THEN270ELSE280
220 D$(R,C)="":GOTO630
230 F#=0:T$=MID$(B$(R,C),2):S#=VAL(T$):GOTO350
240 F#=0:T$=MID$(B$(R,C),2):S#=VAL(T$):GOTO360
250 F=VAL(B$(R,C)):S=VAL(V$):IFF<STHEN280
260 F#=0:S#=VAL(B$(R,C)):GOTO350
```

Listing continued

NOCONA

ELECTRONICS

MODEL 4 **MODEL 16** **MODEL 12**



MOD 2000 W/MONO MONITOR	2299.00	MOD 2000 W/COLOR MONITOR	2669.00
16K MODEL 4 CASS VERS.	579.00	MOD 2000 1 DRIVE W/HARD DISK	3299.00
64K MODEL 4 2 DR RS 232	935.00	DMP 105 PRINTER	145.00
64K MODEL 4 PORTABLE	935.00	DMP 110 PRINTER	269.00
80K MODEL 12 1 DR	2189.00	DMP 120 PRINTER	359.00
80K MODEL 12 2 DR	2719.00	DMP 200 PRINTER	379.00
256K MODEL 16 1 DR	3099.00	DMP 430 PRINTER	585.00
256K MODEL 16 2 DR	3629.00	DMP 500 PRINTER	999.00
256K MODEL 16 1 DR W/HD	4999.00	DMP 2100 PRINTER	825.00
8K MODEL 100 PORTABLE	459.00	DMP 2100 PRINTER	1499.00
24K MODEL 100 PORTABLE	569.00	DWP 210 DAISYWHEEL	429.00
15 MEG HD MOD 12/16 W/KIT	2318.00	DWP 410 DAISYWHEEL	539.00
15 MEG HD MODEL III/4 W/KIT	1999.00	DWP 510 DAISYWHEEL	1139.00
10 MEG HD MOD 2000 INTERNAL	1439.00	DWP II DAISYWHEEL	819.00

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255

✓ 232

Listing continued

```
270 D$(R,C)=MID$(B$(R,C),4):GOTO630
280 D$(R,C)="ENTRY ERROR ":GOTO630
290 T=LEN(B$(R,C)):IFT<4THEN280
300 W$=MID$(B$(R,C),2,2):X$=MID$(B$(R,C),4,1):V=ASC(V$)-64:W=VAL
(W$):IFV>HORW>UTHEN280ELSEF#=VAL(D$(W,V)):IFX$="C"THEN400
310 Y$=MID$(B$(R,C),5,1):IFY$<"0"ORY$>"9"THEN330
320 T=VAL(Y$):Y$=MID$(B$(R,C),5):S#=VAL(Y$):IFS#<TTHEN280ELSE340
330 Z$=MID$(B$(R,C),6,2):Y=ASC(Y$)-64:Z=VAL(Z$):IFY>HORZ>UTHEN28
0ELSES#=VAL(D$(Z,Y))
340 IFX$="+"THEN350ELSEIFX$="-"THEN360ELSEIFX$="*"THEN370ELSEIFX
$="/"THEN380ELSEIFX$="A"THEN410ELSEIFX$="T"THEN420ELSE280
350 T#=F#+S#:GOTO550
360 T#=F#-S#:GOTO550
370 T#=F#*S#:GOTO550
380 IFS#=0THEN280
390 T#=F#/S#:GOTO550
400 T#=F#:GOTO550
410 D=1
420 IFV=YTHEN490
430 F#=VAL(D$(W,V))
440 V=V+1:S#=VAL(D$(W,V))
450 T#=F#+S#:D=D+1:IFV=YTHEN470
460 F#=T#:GOTO440
470 IFX$="T"THEN550
480 T#=T#/D:GOTO550
490 F#=VAL(D$(W,V))
500 W=W+1:S#=VAL(D$(W,V))
510 T#=F#+S#:D=D+1:IFW=ZTHEN530
520 F#=T#:GOTO500
530 IFX$="T"THEN550
540 T#=T#/D
550 IFT#=>0THEN570
560 T#=T#*N:T#=T#-.5:T#=FIX(T#):GOTO580
570 T#=T#*N:T#=T#+.5:T#=FIX(T#):
580 Q$=STR$(N):Q=LEN(Q$)-2:P$=".":T$=STR$(T#):F$=RIGHT$(T$,Q):F=
LEN(T$):IFF=<QTHENF=Q+1
590 T$=LEFT$(T$,F-Q):F=VAL(F$):F$=STR$(F):F$=MID$(F$,2)
600 F=LEN(F$):IFF=QTHEN620
610 S$=STRING$(Q-F,48):F$=S$+F$
620 T$=T$+P$+F$:D$(R,C)=T$
630 F=LEN(D$(R,C)):IFF>12THEN650
640 F=12-F:IFV$="("THEN670ELSE660
650 D$(R,C)=LEFT$(D$(R,C),12):F=0:PRINT@P,"%";
660 PRINTSTRING$(F,32);D$(R,C);:GOTO680
670 PRINTD$(R,C);STRING$(F,32);
680 R=R+1:IFR=I+13THEN700
690 P=P+64:GOTO210
700 R=I:P=P-754:C=C+1:IFC=J+4THEN710ELSE210
710 IFE=1THEN720ELSE730
720 I=I+13:D=D+1:GOTO1900
730 REMCURSORCONTROL
740 C=L:R=K:R$=STR$(R):R$=RIGHT$(R$,2):IFR>9THEN750ELSER$=RIGHT$
(R$,1):T$="0":R$=T$+R$
750 PRINT@B,CHR$(183);:PRINT@B+14,CHR$(187);
760 L$=CHR$(C+64):T$=" = ":L$=L$+R$+T$:PRINT@64,N$;:PRINT@64,L$;
B$(R,C);
770 I$=INKEY$
780 IFI$=""THEN770ELSEIFI$=CHR$(13)THEN770ELSEIFI$=CHR$(34)THEN7
70ELSEIFI$=CHR$(31)THEN190ELSEIFI$=CHR$(91)THEN790ELSEIFI$=CHR$(
10)THEN840ELSEIFI$=CHR$(8)THEN890ELSEIFI$=CHR$(9)THEN940ELSEIFI$
="/"THEN990ELSE770
790 PRINT@B,CHR$(32);:PRINT@B+14,CHR$(32);:K=K-1:IFK<1THENK=1ELS
E810
800 PRINT@B,CHR$(183);:PRINT@B+14,CHR$(187);:GOTO770
810 IFK<ITHEN830
820 B=B-64:GOTO740
830 I=I-1:GOTO120
840 PRINT@B,CHR$(32);:PRINT@B+14,CHR$(32);:K=K+1:IFK>UTHENK=K-1E
LSE860
850 PRINT@B,CHR$(183);:PRINT@B+14,CHR$(187);:GOTO770
860 IFK=I+13THEN880
870 B=B+64:GOTO740
880 I=I+1:GOTO120
890 PRINT@B,CHR$(32);:PRINT@B+14,CHR$(32);:L=L-1:IFL<1THENL=1ELS
E910
900 PRINT@B,CHR$(183);:PRINT@B+14,CHR$(187);:GOTO770
910 IFL<JTHEN930
920 B=B-14:GOTO740
930 J=J-1:GOTO120
940 PRINT@B,CHR$(32);:PRINT@B+14,CHR$(32);:L=L+1:IFL>HTHENL=L-1E
LSE960
950 PRINT@B,CHR$(183);:PRINT@B+14,CHR$(187);:GOTO770
960 IFL=J+4THEN980
```

Listing continued

Listing continued

```
970 B=B+14:GOTO740
980 J=J+1:GOTO120
990 REMCOMMANDS
1000 PRINT@64,"Alpha Blank Copy Del Goto Help Load Print Quit Re
pli Save Val =";
1010 I$=INKEY$:IFI$=""THEN1010ELSEIFI$=CHR$(31)THEN740ELSEIFI$="
A"THEN1030ELSEIFI$="B"THEN1050ELSEIFI$="C"THEN1070ELSEIFI$="G"TH
EN1470ELSEIFI$="H"THEN1600ELSEIFI$="L"THEN1670ELSEIFI$="P"THEN18
50ELSEIFI$="Q"THEN2020ELSE1020
1020 IFI$="D"THEN1420ELSEIFI$="R"THEN2240ELSEIFI$="S"THEN2050ELS
EIFI$="V"THEN2520ELSEIFI$="="THEN2580ELSE1010
1030 REMALPHA
1040 T$="(a)":PRINT@0,N$;N$;:PRINT@0,"Type entry then depress <E
NTER>";:PRINT@64,L$;T$;:GOTO2540
1050 REMBLANK
1060 B$(R,C)="":GOTO200
1070 REMCOPY
1080 PRINT@0,N$;N$;:PRINT@0,"Link cells?  <Y>es or <N>o";
1090 I$=INKEY$:IFI$=""THEN1090ELSEIFI$=CHR$(31)THEN190ELSEIFI$="
Y"THENE=0ELSEIFI$="N"THENE=2ELSE1090
1100 PRINT@0,STRING$(128,32);:PRINT@0,"Copy <C>ell, <D>own, or <
R>ight?";
1110 I$=INKEY$:IFI$=""THEN1110ELSEIFI$="C"THEN1320ELSEIFI$="D"TH
EN1120ELSEIFI$="R"THEN1220ELSEIFI$=CHR$(31)THEN190ELSE1110
1120 PRINT@64,"Type number of rows to copy down then <ENTER> ";:
T$="":F$=LEFT$(L$,3):S$="C":F$=F$+S$:IFB$(R,C)=""THENF$=""
1130 I$=INKEY$:IFI$=""THEN1130ELSEIFI$=CHR$(31)THEN190ELSEIFI$=>
"0"ANDI$=<"9"THEN1140ELSEIFI$=CHR$(13)THEN1150ELSE1130
1140 PRINTI$;:T$=T$+I$:GOTO1130
1150 F=R+1:S=VAL(T$):T=R+S:IFT>UTHENT=U
1160 IFLEFT$(B$(R,C),1)="("THEN1170ELSE1180
1170 E=0:FORD=FTOT:B$(D,C)=B$(R,C):NEXT:GOTO190
1180 IFE=0THEN1210
1190 E=0:F#=VAL(D$(R,C)):F$=STR$(F#):IFF#<0THEN1210
1200 F$=MID$(F$,2)
1210 FORD=FTOT:B$(D,C)=F$:NEXT:GOTO190
1220 PRINT@64,"Type number of columns to copy right then <ENTER>
 ";:T$="":F$=LEFT$(L$,3):S$="C":F$=F$+S$:IFB$(R,C)=""THENF$=""
1230 I$=INKEY$:IFI$=""THEN1230ELSEIFI$=CHR$(31)THEN190ELSEIFI$=>
"0"ANDI$=<"9"THEN1240ELSEIFI$=CHR$(13)THEN1250ELSE1230
1240 PRINTI$;:T$=T$+I$:GOTO1230
1250 F=C+1:S=VAL(T$):T=S+C:IFT>HTHENT=H
1260 IFLEFT$(B$(R,C),1)="("THEN1270ELSE1280
1270 E=0:FORD=FTOT:B$(R,D)=B$(R,C):NEXT:GOTO190
1280 IFE=0THEN1310
1290 E=0:F#=VAL(D$(R,C)):F$=STR$(F#):IFF#<0THEN1310
1300 F$=MID$(F$,2)
1310 FORD=FTOT:B$(R,D)=F$:NEXT:GOTO190
1320 S$=LEFT$(L$,3):F$="C":F$=S$+F$:PRINT@64,"Copy ";S$;" to ";:
S$=""
1330 I$=INKEY$:IFI$=""THEN1330ELSEIFI$=CHR$(31)THEN190ELSEIFI$<"
A"ORI$>CHR$(H+64)THEN1330ELSET$=I$
1340 PRINTI$;
1350 I$=INKEY$:IFI$=""THEN1350ELSEIFI$=CHR$(31)THEN190ELSEIFI$<"
0"ORI$>"9"THEN1350
1360 PRINTI$;:S$=S$+I$:T=LEN(S$):IFT<2THEN1350
1370 F=ASC(T$)-64:S=VAL(S$)
1380 IFLEFT$(B$(R,C),1)="("THEN1410
1390 IFE=2THEN1410
1400 B$(S,F)=F$:GOTO190
1410 E=0:B$(S,F)=B$(R,C):GOTO190
1420 REMDELETE
1430 F$=LEFT$(B$(R,C),1):IFF$="("THEN190
1440 F#=VAL(D$(R,C)):F$=STR$(F#):IFF#<0THEN1460
1450 F$=MID$(F$,2)
1460 B$(R,C)=F$:GOTO190
1470 REMGOTO
1480 PRINT@0,N$;N$;:PRINT@0,"Go to which cell? ";
1490 I$=INKEY$:IFI$=""THEN1490ELSEIFI$=CHR$(31)THEN190ELSEIFI$<"
A"ORI$>CHR$(H+64)THEN1490ELSEF$=I$:S$=""
1500 PRINTI$;
1510 I$=INKEY$:IFI$=""THEN1510ELSEIFI$=CHR$(31)THEN190ELSEIFI$<"
0"ORI$>"9"THEN1510
1520 PRINTI$;:S$=S$+I$:T=LEN(S$):IFT<2THEN1510
1530 S=VAL(S$):IFS>UTHEN2740
1540 J=ASC(F$)-64:L=J:C=L:F=0:IFJ>H-3THEN1550ELSE1560
1550 J=H-3:F=H-L:F=F*14:F=42-F
1560 R=VAL(S$):IFR>UTHEN2740
1570 IFR>U-12THEN1590
1580 I=R:K=R:B=197:B=B+F:GOTO120
1590 I=R/13+.95:I=FIX(I):I=I*13-12:T=(R-I)*64:B=T+197+F:K=R:GOTO
120
1600 REMHELP
```

Listing continued

Listing continued

```
1610 CLS:PRINT"* * * HELP ON COMMANDS * * *":PRINT"<A>lpha allow
s cell to contain a non-numeric/formula value":PRINT"<B>lank cle
ars cell and deletes any formula used to create it":PRINT"<C>opy
 makes cell equal to value of a specified cell"
1620 PRINT"<D>el(ete) formula used to calculate cell but retain
value"
1630 PRINT"<G>oto positions cell pointer (cursor) at specified c
ell":PRINT"<L>oads a previously saved file (designated files 1 t
hru 9)":PRINT"<P>rints a copy of all rows of displayed columns"
1640 PRINT"<Q>uit ends session giving option to save data to one
 of 9 files<R>epli(cate) projects/modifies a formula over a seri
es of cells<S>aves data to one of 9 files":PRINT"<V>al(ue) estab
lishes contents of a cell through keyboard input"
1650 PRINT"<CLEAR> cancels command/action and returns to spreads
heet":PRINT"<=> Accesses built-in five function calculator":PRIN
T:PRINT"       * * * DEPRESS <CLEAR> TO RETURN TO WORKSHEET * * *
";
1660 I$=INKEY$:IFI$=""THEN1660ELSEIFI$=CHR$(31)THEN110ELSE1660
1670 REMLOAD
1680 PRINT@0,N$;N$;:PRINT@0,"LOAD WHICH FILE?  (1-9) ";
1690 I$=INKEY$:IFI$=""THEN1690ELSEIFI$=CHR$(31)THEN190ELSEIFI$<"
0"ORI$>"9"THEN1690
1700 C=1:R=1:A$=LEFT$(A$,5):A$=A$+I$
1710 PRINT@64,"LOADING ";A$;" -- PLEASE WAIT";:IFI$="1"THEN1720E
LSEIFI$="2"THEN1730ELSEIFI$="3"THEN1740ELSEIFI$="4"THEN1750ELSEI
FI$="5"THEN1760ELSEIFI$="6"THEN1770ELSEIFI$="7"THEN1780ELSEIFI$=
"8"THEN1790ELSEIFI$="9"THEN1800
1720 CLOSE:OPEN"I",2,"FILE1":GOTO1810
1730 CLOSE:OPEN"I",2,"FILE2":GOTO1810
1740 CLOSE:OPEN"I",2,"FILE3":GOTO1810
1750 CLOSE:OPEN"I",2,"FILE4":GOTO1810
1760 CLOSE:OPEN"I",2,"FILE5":GOTO1810
1770 CLOSE:OPEN"I",2,"FILE6":GOTO1810
1780 CLOSE:OPEN"I",2,"FILE7":GOTO1810
1790 CLOSE:OPEN"I",2,"FILE8":GOTO1810
1800 CLOSE:OPEN"I",2,"FILE9":GOTO1810
1810 INPUT#2,B$(R,C)
1820 C=C+1:IFC=<HTHEN1810
1830 C=1:R=R+1:IFR=<UTHEN1810
1840 CLOSE:GOTO100
1850 REMPRINT
1860 PRINT@0,N$;N$;:PRINT@0,"READY PRINTER -- Press <ENTER> or <
CLEAR> to cancel";
1870 I$=INKEY$:IFI$=""THEN1870ELSEIFI$=CHR$(31)THEN190ELSEIFI$=C
HR$(13)THEN1880ELSE1870
1880 PRINT@64,"Print <D>isplayed values or <F>ormulas?";:I$=INKE
Y$:IFI$=""THEN1880ELSEIFI$="D"THEN1890ELSEIFI$="F"THEN1920ELSEIF
I$=CHR$(31)THEN190ELSE1880
1890 D=1:G=U/13:M=I:I=1:E=1
1900 IFD>GTHEN1910ELSE120
1910 E=0:I=M
1920 PRINT@64,N$;:PRINT@64,"PRINTING -- PLEASE WAIT";
1930 S$="%            %":R=1:C=J:LPRINTSTRING$(10,35);" ";CHR$(C+6
4);" ";STRING$(12,35);" ";CHR$(C+65);" ";STRING$(12,35);" ";CHR$
(C+66);" ";STRING$(12,35);" ";CHR$(C+67);" ";STRING$(7,35)
1940 R$=STR$(R):T=LEN(R$):T=3-T
1950 LPRINTSTRING$(T,32);R$;" * ";
1960 IFI$="D"THEN1980
1970 LPRINTUSINGS$;B$(R,C);:LPRINT" * ";:GOTO1990
1980 LPRINTUSINGS$;D$(R,C);:LPRINT" * ";
1990 C=C+1:IFC<J+4THEN1960
2000 LPRINT:R=R+1:C=J:IFR=<UTHEN1940
2010 LPRINTSTRING$(65,35):R=K:C=L:GOTO120
2020 REMQUIT
2030 PRINT@0,N$;N$;:PRINT@0,"SAVE CONTENTS -- Depress <Y>es to s
ave or <N>o";
2040 I$=INKEY$:IFI$=""THEN2040ELSEIFI$=CHR$(31)THEN190ELSEIFI$="
Y"THEN2050ELSEIFI$="N"THEN2230ELSE2040
2050 PRINT@0,N$;N$;:PRINT@0,"Save as which file?  (1-9)";
2060 I$=INKEY$:IFI$=""THEN2060ELSEIFI$=CHR$(31)THEN190ELSEIFI$>"
0"ANDI$=<"9"THEN2070ELSE2060
2070 C=1:R=1:A$=LEFT$(A$,5):A$=A$+I$
2080 PRINT@64,"SAVING AS ";A$;" -- PLEASE WAIT";:IFI$="1"THEN209
0ELSEIFI$="2"THEN2100ELSEIFI$="3"THEN2110ELSEIFI$="4"THEN2120ELS
EIFI$="5"THEN2130ELSEIFI$="6"THEN2140ELSEIFI$="7"THEN2150ELSEIFI
$="8"THEN2160ELSEIFI$="9"THEN2170
2090 CLOSE:OPEN"O",1,"FILE1":GOTO2180
2100 CLOSE:OPEN"O",1,"FILE2":GOTO2180
2110 CLOSE:OPEN"O",1,"FILE3":GOTO2180
2120 CLOSE:OPEN"O",1,"FILE4":GOTO2180
2130 CLOSE:OPEN"O",1,"FILE5":GOTO2180
2140 CLOSE:OPEN"O",1,"FILE6":GOTO2180
2150 CLOSE:OPEN"O",1,"FILE7":GOTO2180
```

Listing continued

✓ See List of Advertisers on Page 145

Listing continued

```
2160 CLOSE:OPEN"O",1,"FILE8":GOTO2180
2170 CLOSE:OPEN"O",1,"FILE9"
2180 PRINT#1,CHR$(34);B$(R,C);CHR$(34);
2190 C=C+1:IFC=<HTHEN2180
2200 C=1:R=R+1:IFR=<UTHEN2180
2210 PRINT@64,"<R>eturn to spreadsheet or <N>ew spreadsheet or <
E>xit program?";
2220 I$=INKEY$:IFI$=""THEN2220ELSEIFI$="R"THEN190ELSEIFI$="N"THE
N10ELSEIFI$="E"THEN2230ELSE2220
2230 CLOSE:CMD"T":CLS:END
2240 REMREPLICATE
2250 V$=LEFT$(B$(R,C),1):IFV$<"A"ORV$>CHR$(U+64)THEN2740
2260 PRINT@0,N$;N$;:PRINT@0,"Replicate formula <D>own or <R>ight
? ";
2270 I$=INKEY$:IFI$=""THEN2270ELSEIFI$=CHR$(31)THEN190ELSEIFI$="
D"THEN2280ELSEIFI$="R"THEN2420ELSE2270
2280 PRINTI$;:PRINT@64,"Type number of rows to replicate down th
en <ENTER> ";:T$=""
2290 I$=INKEY$:IFI$=""THEN2290ELSEIFI$=CHR$(31)THEN190ELSEIFI$=C
HR$(13)THEN2310ELSEIFI$=>"0"ANDI$=<"9"THEN2300ELSE2290
2300 PRINTI$;:T$=T$+I$:GOTO2290
2310 S$="0":F=R:S=VAL(T$):S=S+R:IFS>UTHENS=U
2320 V$=LEFT$(B$(F,C),1):W$=MID$(B$(F,C),2,2):W=VAL(W$):X$=MID$(
B$(F,C),4,1):W=W+1:IFW>UTHEN2740
2330 W$=STR$(W):IFW>9THEN2350
2340 W$=RIGHT$(W$,1):W$=S$+W$
2350 W$=RIGHT$(W$,2):F$=V$+W$+X$:Y$=MID$(B$(F,C),5,1):IFY$=>"A"A
NDY$<CHR$(U+65)THEN2370ELSEIFY$=""THEN2410
2360 Y$=MID$(B$(F,C),5):F$=F$+Y$:GOTO2410
2370 Z$=MID$(B$(F,C),6,2):Z=VAL(Z$):Z=Z+1:IFZ>UTHEN2740
2380 Z$=STR$(Z):IFZ>9THEN2400
2390 Z$=RIGHT$(Z$,1):Z$=S$+Z$
2400 Z$=RIGHT$(Z$,2):F$=F$+Y$+Z$
2410 F=F+1:B$(F,C)=F$:IFF=STHEN190ELSE2320
2420 PRINTI$;:PRINT@64,"Type number of columns to replicate righ
t then <ENTER> ";:T$=""
2430 I$=INKEY$:IFI$=""THEN2430ELSEIFI$=CHR$(31)THEN190ELSEIFI$=C
HR$(13)THEN2450ELSEIFI$=>"0"ANDI$=<"9"THEN2440ELSE2430
2440 PRINTI$;:T$=T$+I$:GOTO2430
2450 S$="0":F=C:S=VAL(T$):S=S+C:IFS>HTHENS=H
2460 V$=LEFT$(B$(R,F),1):V=ASC(V$)+1:W$=MID$(B$(R,F),2,2):X$=MID
$(B$(R,F),4,1):IFV-64>HTHEN2740
2470 F$=CHR$(V):F$=F$+W$+X$:Y$=MID$(B$(R,F),5,1):IFY$=>"A"ANDY$=
<CHR$(H+65)THEN2490ELSEIFY$=""THEN2510
2480 Y$=MID$(B$(R,F),5):F$=F$+Y$:GOTO2510
2490 Z$=MID$(B$(R,F),6,2):Y=ASC(Y$)+1:IFY-64>HTHEN2740
2500 Y$=CHR$(Y):F$=F$+Y$+Z$
2510 F=F+1:B$(R,F)=F$:IFF=STHEN190ELSE2460
2520 REMVALUE
2530 PRINT@0,N$;N$;:PRINT@0,"Type value or formula then depress
<ENTER>";:PRINT@64,L$;:T$=""
2540 I$=INKEY$:IFI$=""THEN2540ELSEIFI$=CHR$(13)THEN2560ELSEIFI$=
CHR$(31)THEN190ELSEIFI$=CHR$(34)THEN2540ELSEIFI$=CHR$(91)ORI$>CH
R$(7)ANDI$<CHR$(11)THEN2730
2550 T$=T$+I$:PRINTI$;:GOTO2540
2560 B$(R,C)=T$:GOTO190
2570 REM(=)CALCULATOR
2580 T#=VAL(B$(R,C)):PRINT@0,N$;N$;:PRINT@0,"Start with <C>ell v
alue (";T#;") or <Z>ero?"
2590 I$=INKEY$:IFI$=""THEN2590ELSEIFI$="C"THEN2620ELSEIFI$="Z"TH
EN2600ELSEIFI$=CHR$(31)THEN190ELSE2590
2600 T#=0:F#=0
2610 F$=""
2620 PRINT@0,N$;N$;:PRINT@0,"SUBTOTAL = ";T#;:PRINT@64,"ENTER VA
LUE:  ";
2630 I$=INKEY$:IFI$=""THEN2630ELSEIFI$=CHR$(31)THEN190ELSEIFI$=>
"0"ANDI$=<"9"THEN2640ELSEIFI$="+"THEN2650ELSEIFI$="-"THEN2660ELS
EIFI$="*"THEN2670ELSEIFI$="/"THEN2680ELSEIFI$="="THEN2700ELSE263
0
2640 PRINTI$;:F$=F$+I$:GOTO2630
2650 F#=VAL(F$):T#=T#+F#:GOTO2610
2660 F#=VAL(F$):T#=T#-F#:GOTO2610
2670 F#=VAL(F$):T#=T#*F#:GOTO2610
2680 F#=VAL(F$):IFF#=0THEN2630
2690 T#=T#/F#:GOTO2610
2700 IFT#<0THEN2720
2710 F$=STR$(T#):B$(R,C)=MID$(F$,2):GOTO190
2720 B$(R,C)=STR$(T#):GOTO190
2730 B$(R,C)=T$:PRINT@0,"RAPID ENTRY (Press </> for commands <CL
EAR> updates)";:PRINT@B+1,STRING$(13,32);:PRINT@B+2,LEFT$(B$(R,C
),12);:GOTO780
2740 PRINT@64,N$;:PRINT@64,"* * * ENTRY ERROR * * *";:FORD=1TO15
00:NEXT:GOTO190
```

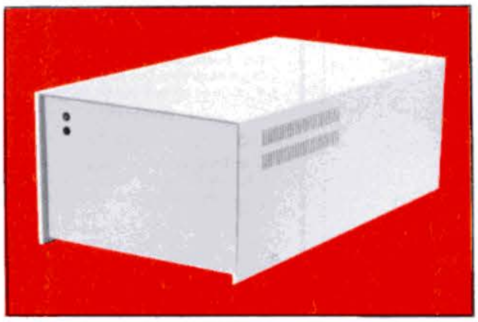
End

✓ See List of Advertisers on Page 145

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Hard Drives



Warranty - One Full Year

Introducing

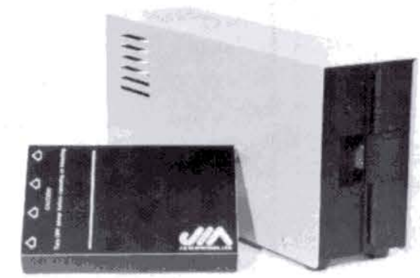
MEGADISK™

Winchester Hard Drive ready to run on the TRS 80 Model I/III/IV/4P, Color Computer, I.B.M.-PC, Apple, Franklin, Max/80.
Software Drivers: LDOS, NewDos/80, Dosplus, TRSDOS 6.0, CP/M

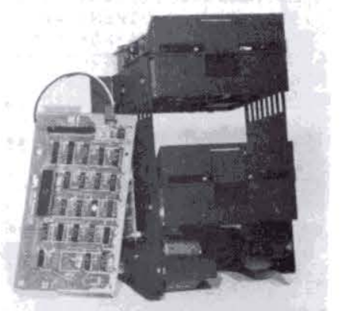
CALL FOR NEW LOW PRICE

DRIVE A HARD BARGAIN™ Complete Systems Starting at ~~$999.95~~

Call Toll Free Ordering 1-800-343-8841



High Quality Lowest Price
Drive 0, 1, 2, 3
for the
Color Computer
Starting at ~~$199.95~~ CALL FOR NEW LOW PRICE



Disk Drive Upgrade
for Model III/IV easy to install system
Starting at ~~$309.95~~ CALL FOR NEW LOW PRICE

SOFTWARE SUPPORT, INC.

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425
Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10:00 am to 3:30 pm

DEALER INQUIRIES INVITED.

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.
Prices subject to change.

CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular Tel. (514) 287-1563
Canadian Toll Free 800-361-5155

Service! Service!
All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. School and D&B corporate P.O.s accepted.

TRS/80 Registered Trademark Tandy Corp. Apple Registered Trademark Apple Computer Corp
IBM-PC Registered IBM Corp. Franklin Registered Trademark Franklin Corp. Max/80 Registered Trademark Lobo Int.
LDOS Reg. Logical System Inc. Dosplus - Micro Systems Software Newdos/80 - Apparat Inc.

See List of Advertisers on page 80

✓ See List of Advertisers on Page 145

by Glen E. Sparks

Picture Perfect

Here's a complete graphics production package—you get the worksheet on which to design your graphics, an easy-to-use coding system to program a dot-matrix print head, and a number of printout options.

While you're probably not disappointed with the LP/DMP printers from Radio Shack, you might be a little jealous of the graphics capability of the Epson printers. It seems like all the good printer graphics programs run on the Epsons. But now you can get better and more interesting graphics from your Radio Shack printer by using the graphics technique I've developed.

Basic Concepts

Program Listing 1 prints out worksheets that let you design and map graphics characters of varying sizes (see Fig. 1). First type in, save, and run Listing 1. Once you start experimenting with your own graphics, you'll need a lot of these worksheets.

The seven numbers (1, 2, 4, 8, 16, 32, and 64) in the worksheet's left-hand column represent the seven controllable print-head wires that, when activated in a specific pattern, create a variety of graphics characters. The print-head pins are numbered this way so that you get a unique sum, in the range of 1–127, for any combination of pins. For example, while the sum of all seven numbers is 127, every combination of the numbers yields a different sum.

The following program prints each of the available TRS-80 graphics characters in a line without gaps. Because of this, you can develop custom graphics characters, made up of individual standard graphics characters, of any length. The CHR$(30) in line 50 of the program below resets the printer, removing it from the graphics mode. Now, type in and run this program:

```
10 LPRINT CHR$(18) 'set printer to graphics
   mode
20 FORI = 128 TO 255
30 LPRINT CHR$(I);
40 NEXTI
50 LPRINT CHR$(30) 'set printer to non-graph-
   ics mode
```

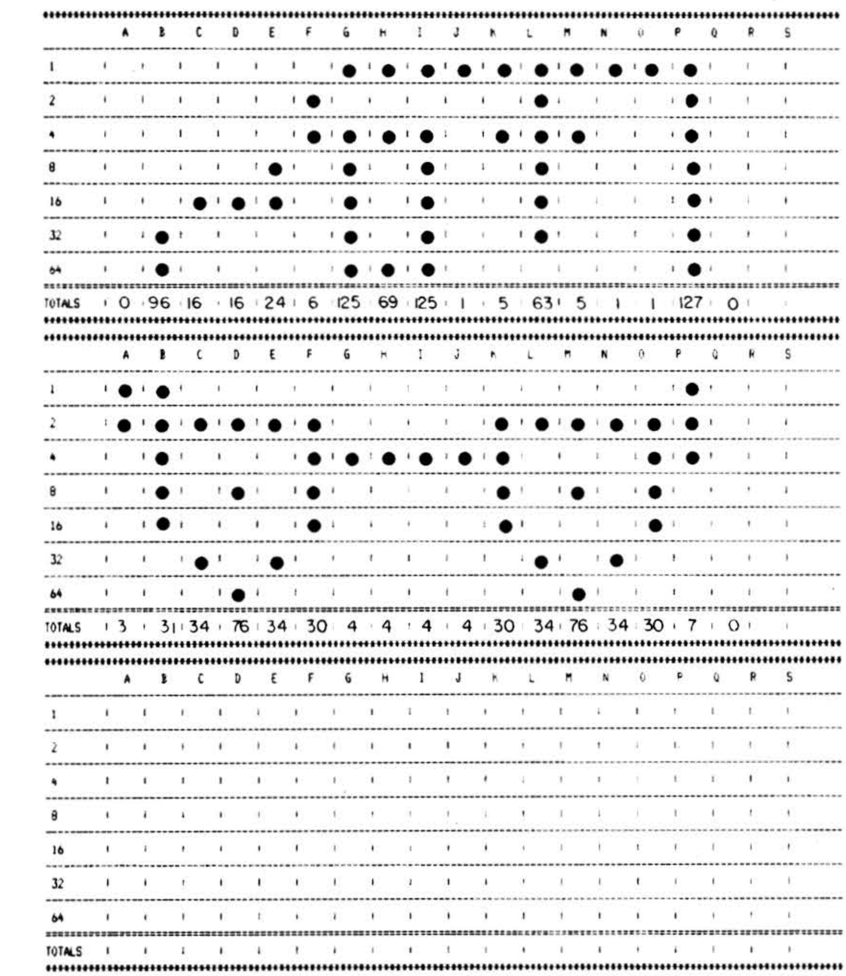


Figure 1. Worksheet and drawing.

Custom Printing

The graphics characters printed here probably look familiar, especially if you've ever shopped for a printer. Be more creative now and try your skill at customized graphics.

Using a soft lead pencil, put dots in each box of column A of the worksheet you created with Program Listing 1. In

The Key Box

LOAD 80

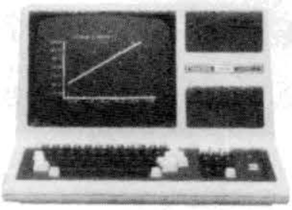
Models I, III and 4
16K RAM
Basic
Radio Shack LP or DMP printer

From Computer Plus to YOU . . .

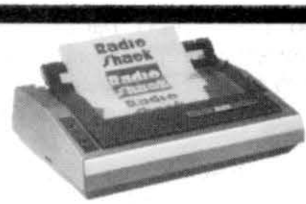
PLUS after PLUS after PLUS

Model 100 8K $495
Model 100 24K $625

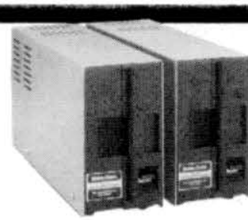
Color Computer II
w/16K Ext. Basic $135
w/64K Ext. Basic $195



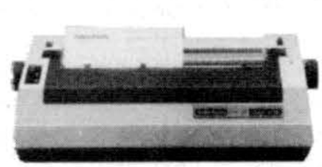
Model 4 16K $629
Model 4 64K
2 Disk & RS232 $1020



DMP120 $385



Color Computer Disk Drive
Drive 0 $289 Drive 1 $220



DWP210 $489
DWP510 $1295

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Model 4 Portable	
64K 2 Drives	1020
Pocket Computer 3	89
Pocket Computer 4	59
Model 2000 2 Drive	2299
Model 2000 W/Hard Disk	3599
Model 12 64K 1 Drive	2360
Model 12 64K 2 Drive	2950
Model 16B 1Dr 256K	3375
Model 16B 2Dr 256K	3965
MODEMS	
Hayes Smartmodem II	215
Hayes Smartmodem 1200	505
Novation Smartcat 1200	410
Novation J-Cat	115
AC-3	125
DC Modem I	89
DC Modem II	160
DC Modem 2212	315

PRINTERS	
Silver Reed EXP500 D.W.	365
Silver Reed EXP550 D.W.	430
DWP410	1159
Gemini Powertype	345
CGP115	159
CGP220 Ink Jet	545
DMP110	299
DMP420	735
Toshiba 1340 (24 wire head)	779
Gemini 10X	265
Gemini 15X	409
Panasonic P1091	315
Smith Corona Fastext	190
Prowriter 8510	345
Okidata	CALL
Epson	CALL
DISK DRIVES	
Model III/4 Drive 0	465
Model III/4 Drive 1	220

Model III/4 Drive 2	259
Model III/4 Drive 3	220
Hard Disk M2000	1525
ETC.	
CCR-81 Recorder	52
Cassette Tapes (10 Pk)	9.95
16K Ram Chips	25
64K Ram Chips	62.95
Model 4 64K Upgrade Kit	79.95
M100 Bar Code Reader	89
64K Printer Buffer	199
Printer Stand	25
Printer Cables	
Printer Ribbons	
Printer Paper	
Computer Books	

Radio Shack software 10% off.
Send for complete listing of brand name software and hardware.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

18

IN MASSACHUSETTS CALL (617) 486-3193

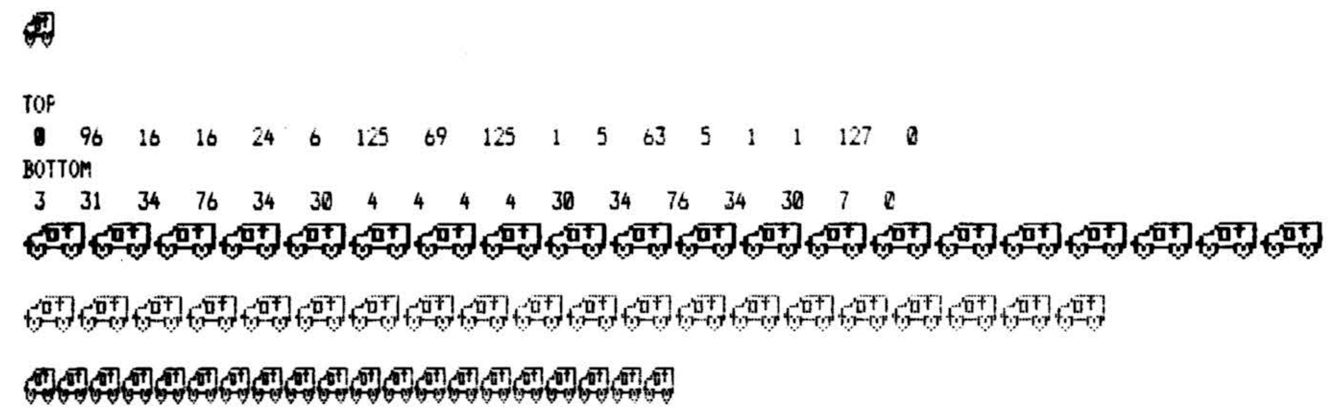


Figure 2. Printout of drawing in Figure 1.

Program Listing 1. Basic program to create work sheets.

```
100 ' GRAPH PAPER FOR CHARGEN/GS1 AND GS2
110 '    SAVED AT GRAPAPER/FRM
120 CLEAR1000
130 LPRINTCHR$(27);CHR$(20)'set to condensed mode
140 CLS
145 PRINT"SET PAPER IN PRINTER AT TOP OF NEW SHEET"
147 PRINT:PRINT"HIT ANY KEY TO START PRINTING"
150 IF INKEY$=""THEN 150
160 FOR FS=1TO3 ' loop to set number of forms
170 LPRINTSTRING$(130,"*")
175 GOSUB1000
180 B=0
190 FORI=1TO8
200 LPRINT" "
210 IFI<8THENGOSUB290  ELSE GOSUB440
220 GOSUB390  'subroutine to put in pin numbers
230 GOSUB310  'lprint column seperating bar
240 NEXTI
250 LPRINT" "
260 LPRINTSTRING$(130,"*")
270 NEXT FS
280 END
290 LPRINTSTRING$(130,"-")
300 RETURN
310 FORI3=1TO19
320 LPRINTTAB(10);
330 LPRINTCHR$(18); 'go into graphics mode
340 LPRINTCHR$(191); 'lprint graphic bar
350 LPRINTCHR$(30); 'come out of graphics mode
360 LPRINT"      "; ' six blank spaces
370 NEXTI3
380 RETURN
390 'SUBROUTINE TO PUT IN NUMBERS
400 A=2[B
410 B=B+1
420 IFA>64 THENLPRINT"TOTALS";:ELSELPRINTA;
430 RETURN
440 'SUBROUTINE TO PRINT TOTAL LINE
450 LPRINTSTRING$(130,"=")
460 RETURN
1000 CL=65
1005 LPRINTTAB(13);
1010 LPRINTCHR$(CL);
1015 LPRINT"      ";
1020 IF CL=83 THEN 1050
1030 CL=CL+1
1040 GOTO1010
1050 RETURN
```

End

Program Listing 2. Basic program to enter one-line graphics.

```
100 '*****************************************************
110 '   CHARGEN/GS1
120 '*****************************************************
130 CLS
140 PRINT"USE GRAPAPER/FRM PROGRAM TO PLOT A SEQUENCE OF PIN FIRINGS"
150 PRINT"THEN ENTER THE TOTAL OF EACH COLUMN IN RESPONSE TO THE P
ROMPTS."
160 PRINT:PRINT"THE PROGRAM WILL ADD 128 TO EACH TOTAL TO GENERATE
 PRINTER"
```

Listing 2 continued

columns B–F put a dot in the top and bottom row only (pin numbers 1 and 64). In column G, put dots from top to bottom again. Now add the values of each column: 127 for A, 65 for B, 65 for C, 65 for D, 65 for E, 65 for F, and 127 for G.

When you run Program Listing 2, it prompts you to type in the totals for each column. After you enter the numbers, type in 999 to terminate data entry. Your printer should be on, and it will print the following: a tiny box, the pin firing codes (the column values), and a three-line series of the character in different font sizes (elongated, normal, and condensed).

Notice that all the boxes are connected; to print a series of individual boxes, enter a zero before typing in 999. Keep in mind that the program automatically adds 128 to the pin codes because the printer requires that all character codes be in a range of 128 to 255.

Try experimenting with different drawings or simply enter a series of numbers; for example, entering 10 numbers at intervals of 5 (don't forget to type 999) makes a nice line of graphics. This type of graphic can be used in a subroutine to separate reports, or as a more interesting alternative to asterisks.

Larger Graphics

Single line graphics are nice, but somewhat limited. Program Listing 3, while operating on the same concept as Program Listing 2, lets you print graphics that are twice as high. When mapping out the larger characters, split the character between two worksheets and calculate column totals for each. After tabulating the totals, run Program Listing 3, which, in turn, prompts you to enter these values.

Remember *to enter zeros in the blank* columns to guarantee alignment. The drawing in Fig. 1 and the corresponding

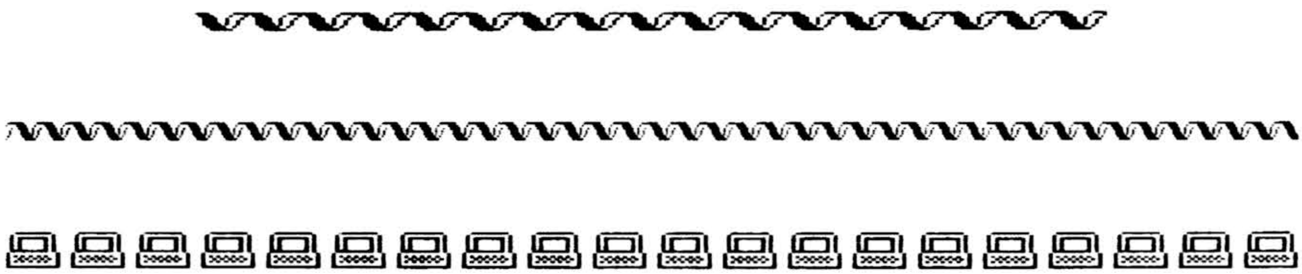


Figure 3. Graphics lines created by Program Listings 4 (top line), 5 and 6 (middle line), and 7 (bottom line).

You can expand this technique to print graphics for all purposes.

printout in Fig. 2 are examples of this size character. (The subroutine in Program Listing 7 is also geared towards the larger graphics characters.)

The subroutines in Program Listings 4–7 illustrate the variety of ways to store and print graphics codes. Program Listing 4, for example, uses CHR$ code links stored in strings to print a line of graphics characters. I used A$ and A1$ only because I didn't want to print off the page when listing the program. Your strings can be much longer if you choose.

Because this particular character is good only when printed in a line, I used a For...Next loop to fill the page from left to right. The string method used here is quite fast, especially when compared with the subroutines in Program Listings 5 and 6.

Program Listings 5 and 6 both use Read Data statements. While one is quicker than the other, they're both slow. I've included these because it's easier to change data statements in each graphics subroutine than to retype the CHR$ codes. If time isn't critical, this method makes it easy to build a library of subroutines.

You can easily expand the technique discussed here to print graphics of all sizes for all purposes. Playing with combinations of column values is both fun and productive. Happy printing! ■

Write to Glen E. Sparks at 6186 Custer, S. Rockwood, MI 48179.

Listing 2 continued

```
170 PRINT"GRAPHICS IN A LINE USING EACH TO THE FONTS AVAILABLE TO
GRAPHICS"
180 PRINT"IT WILL ALSO PRINT OUT THE COLUMN TOTALS FOR REFERENCE."
185 PRINT:PRINT"TYPE 999 TO END INPUT"
190 PRINT:PRINT"HIT ANY KEY TO BEGIN"
200 PRINT:PRINT:PRINT
210 IF INKEY$=""THEN210
220 CLS
230 CLEAR500
240 DIMN(50)
250 Z=1
260 '****************************************
270 '  INPUT CODES
280 '****************************************
290 CLS
300 I=1
305 CL=65
310 PRINT"PIN COLUMN ";CHR$(CL);" ";
315 INPUT N(I)
320 IFN(I)=999 THEN 370
330 IFN(I)>127 THENPRINT"TOO HIGH-MUST BE UNDER 127":GOTO310
350 I=I+1
355 CL=CL+1
360 GOTO310
370 '****************************************
380 '  SEND TO PRINTER
390 '****************************************
400 T=I
410 FORI=1TOT
415 IFN(I)=999 THEN N(I)=0
420 A$=CHR$(N(I)+128)
430 LPRINT CHR$(18);A$; ' put into graphics mode and lprint CHR$
440 NEXTI
450 LPRINTCHR$(30) 'set to non-graphic text
460 LPRINTCHR$(27);CHR$(20) 'set to condensed for code number prin
tout
470 FORI=1TO(T-1):LPRINTN(I);:LPRINT" ";
480 NEXTI
490 '****************************************
500 '  SEND LINE OF CUSTOM CHARACTERS TO PRINTER
530 '   SET CHARACTER FONT FOR LINE OF CUST. CHAR
540 '************************************************************
560 CLS
570 IFZ>1THENLPRINTCHR$(27);CHR$(15) 'get out of Expanded mode
580 ONZ GOTO 590  ,610  ,630  ,650
590 LPRINT CHR$(27);CHR$(14) 'expanded mode
600 GOTO670
610 LPRINTCHR$(27);CHR$(19) 'normal characters
620 GOTO670
630 LPRINTCHR$(27);CHR$(20) 'condensed mode
640 GOTO670
650 RUN240
669 '********************************************
670 ' PRINTOUT OF CHAR REPEATED 20 TIMES
671 '********************************************
680 FORI1=1TO20
690 FORI=1TO(T-1)
700 A$=CHR$(N(I)+128) 'add 128 to pin number to produce CHR$ code
710 LPRINTCHR$(18);A$; 'go into graphics mode and print char
720 NEXTI:NEXTI1
730 LPRINTCHR$(30) 'get out of graphics mode
740 Z=Z+1
750 GOTO 570
```

End

Program Listing 3. Basic program to enter two-line graphics.

```
100 '*********************************************************
110 '   CHARGEN/GS2
120 '*********************************************************
130 CLS
140 PRINT"USE GRAPAPER/FRM PROGRAM TO PLOT A SEQUENCE OF PIN FIRIN
GS"
150 PRINT"THEN ENTER THE TOTAL OF EACH COLUMN IN RESPONSE TO THE P
ROMPTS."
155 PRINT"TYPE 999 TO TERMINATE DATA INPUT"
160 PRINT:PRINT"THE PROGRAM WILL ADD 128 TO EACH TOTAL TO GENERATE
 PRINTER"
170 PRINT"GRAPHICS IN A LINE USING EACH TO THE FONTS AVAILABLE TO
GRAPHICS"
180 PRINT"IT WILL ALSO PRINT OUT THE COLUMN TOTALS FOR REFERENCE."
190 PRINT:PRINT"HIT ANY KEY TO BEGIN"
200 PRINT:PRINT:PRINT
210 IF INKEY$=""THEN210
220 CLS
230 CLEAR500
240 DIMN(50)
241 DIMN1(50)
250 Z=1
260 '**************************************
270 '   INPUT CODES
280 '**************************************
290 CLS
300 I=1
305 CL=65
310 PRINT"PIN COLUMN ";CHR$(CL);" ";
315 INPUT N(I)
320 IFN(I)=999 THEN 370
330 IFN(I)>127PRINT"TOO HIGH-MUST BE UNDER 127":GOTO310
331 INPUT"BOTTOM HALF ";N1(I)
332 IFN1(I)>127THEN PRINT"TOO HIGH-MUST BE UNDER 127":GOTO331
350 I=I+1
355 CL=CL+1
360 GOTO310
370 '**************************************
380 '  SEND TO PRINTER
390 '**************************************
400 T=I
410 FORI=1TOT
415 IFN(I)=999 THEN N(I)=0
420 A$=CHR$(N(I)+128)
430 LPRINT CHR$(18);A$; ' put into graphics mode and lprint CHR$
440 NEXTI
441 LPRINT""
442 FORI=1TOT
443 B$=CHR$(N1(I)+128)
444 LPRINTCHR$(18); B$;
445 NEXTI
450 LPRINTCHR$(30) 'set to non-graphic text
460 LPRINTCHR$(27);CHR$(20) 'set to condensed for code number prin
tout
465 LPRINT"TOP "
470 FORI=1TO(T-1):LPRINTN(I);:LPRINT" ";
480 NEXTI
481 LPRINT""
483 LPRINT"BOTTOM "
485 FORI=1TO(T-1):LPRINTN1(I);:LPRINT" ";:NEXTI
490 '**************************************
500 '  SEND LINE OF CUSTOM CHARACTERS TO PRINTER
530 '    SET CHARACTER FONT FOR LINE OF CUST. CHAR
540 '*********************************************************
560 CLS
570 IFZ>1THENLPRINTCHR$(27);CHR$(15) 'get out of Expanded mode
580 ONZ GOTO 590  ,610  ,630  ,650
590 LPRINT CHR$(27);CHR$(14) 'expanded mode
600 GOTO670
610 LPRINTCHR$(27);CHR$(19) 'normal characters
620 GOTO670
630 LPRINTCHR$(27);CHR$(20) 'condensed mode
640 GOTO670
650 RUN240
669 '******************************************
670 ' PRINTOUT OF CHAR REPEATED 20 TIMES
671 '******************************************
680 FORI1=1TO20
690 FORI=1TO(T-1)
700 A$=CHR$(N(I)+128) 'add 128 to pin number to produce CHR$ code
710 LPRINTCHR$(18);A$; 'go into graphics mode and print char
720 NEXTI:NEXTI1
721 LPRINT""
722 FORI1=1TO20
723 FORI=1TO(T-1)
724 B$=CHR$(N1(I)+128)
725 LPRINTCHR$(18);B$;
726 NEXTI:NEXTI1
```

Listing 3 continued

Listing 3 continued

```
727 LPRINT""
730 LPRINTCHR$(30) 'get out of graphics mode
740 Z=Z+1
750 GOTO 570
```

End

Program Listing 4. Subroutine using string method.

```
2000 'STRINGS/GS3
2010 '2 seconds to load buffer and printout line
2020 'add 128 to each pin column total
2030 A$=CHR$(129)+CHR$(131)+CHR$(159)+CHR$(191)+CHR$(254)+CHR$(252)
2040 A1$=CHR$(240)+CHR$(224)+CHR$(220)+CHR$(194)+CHR$(161)+CHR$(15
3)+CHR$(133)+CHR$(131)
2050 LPRINTCHR$(27);CHR$(14)'set printer to expanded mode
2060 LPRINTCHR$(18)'set to graphics mode
2070 FORI=1TO17
2080 LPRINTA$+A1$;
2090 NEXTI
2100 LPRINTCHR$(27);CHR$(15)'reset printer out of expanded mode
2110 LPRINTCHR$(30)'reset printer to non-graphics mode
2120 RETURN
```

End

Program Listing 5. Subroutine that prints graphics line.

```
3000 'SUBGRAF/GS4
3010 'Subroutine to print graphic line (rolling ribbon graphic)
3020 '17 second delay to fill buffer and print
3030 LPRINTCHR$(27);CHR$(20) 'set to condensed mode
3040 LPRINTCHR$(18) ' set to graphics mode
3050 FORI=1TO50
3060 READN:IFN=999THEN3080
3070 IFN>=0 THEN LPRINTCHR$(128+N);:GOTO3060
3080 LPRINT;
3090 RESTORE 'set to read data again
3100 NEXTI
3110 LPRINTCHR$(30)' reset printer to non-graphics mode
3120 DATA1,3,7,31,63,126,124,112,96,92,66,33,25,5,1,999
3130 RETURN
```

End

Program Listing 6. Subroutine that prints graphics line.

```
3000 'SUBGRAF/GS5
3005 DIMM(20)
3010 'Subroutine to print graphic line (rolling ribbon graphic)
3020 'ONLY 9 second delay to fill buffer and print
3030 LPRINTCHR$(27);CHR$(20) 'set to condensed mode
3040 LPRINTCHR$(18) ' set to graphics mode
3050 I=1
3060 READN:M(I)=N:IFN>255THEN3085
3080 I=I+1:GOTO3060
3085 T=I:FORJ=1TO50
3090 FORI=1TO(T-1):LPRINTCHR$(M(I)+128);:NEXT
3100 NEXTJ
3110 LPRINTCHR$(30)' reset printer to non-graphics mode
3120 DATA1,3,7,31,63,126,124,112,96,92,66,33,25,5,1,999
3130 RETURN
```

End

Program Listing 7. Subroutine that prints two-line characters.

```
3000 'SUBGRAF/GS6  double high graphic
3010 DIMM(20),M1(20)
3020 LPRINTCHR$(27);CHR$(14) 'set printer to expanded mode
3030 LPRINTCHR$(18)  'set printer to graphic mode
3040 ' Read and printout top portion graphic
3050 I=1
3060 READN:M(I)=N:IFN>255THEN3080
3070 I=I+1:GOTO3060
3080 T=I:FORJ=1TO20
3090 FORI=1TO(T-1):LPRINTCHR$(M(I)+128);:NEXT
3100 NEXTJ
3110 ' Read and printout bottom portion graphic
3120 I=1:LPRINT
3130 READN1:M1(I)=N1:IFN1>255THEN3150
3140 I=I+1:GOTO3130
3150 T=I:FORJ=1TO20
3160 PRINT:FORI=1TO(T-1):LPRINTCHR$(M1(I)+128);:NEXT
3170 NEXTJ
3180 LPRINTCHR$(30)  'reset printer to non-graphics mode
3190 LPRINTCHR$(27);CHR$(15) 'reset printer out of expanded mode
3200 DATA 0,0,126,1,125,5,5,5,5,5,5,125,1,126,0,0,0,999
3210 DATA 0,126,65,85,73,85,73,85,73,85,73,85,72,65,126,0,0,999
3220 RETURN
```

End

✓ See List of Advertisers on Page 145

by Jane Goodale

ON THE RECORD

Sequential random-access files each have their own advantages and disadvantages for data storage and retrieval. Here's a way to combine the best features of both.

Sequential files use disk space efficiently, but they have their disadvantages. To update records, a program must read and write an entire file. And searching for a specific record can require extensive disk input/output or main memory, depending on the technique you use. Random-access files (also called direct-access files) don't have these drawbacks and, with good planning, they can be reasonably efficient users of disk space.

I'll introduce three Model I/III Basic programs that create and sort indexed random-access files, letting you easily manipulate data stored on disk. First, however, I'll describe how random-access files work.

Records and Subrecords

Table 1 shows the Disk Basic commands you use to create and access random files. Random-access files store data in buffers. A buffer is one 255-character record; you can define subrecords (i.e., fields) within the buffer using the Field command. The Get command accesses the full record; you need a file-handling program to access the subrecords.

For example, assume you want a file of names and phone numbers that sets aside 25 characters for a name and 12 characters for a phone number. Positions 1–25 in the file buffer contain name characters and positions 26–37 contain the phone number. Each 255-character main record could hold six of these 37-character subrecords, with 33 unused characters left over.

Here's how you'd open this file and define its fields:

```
10 OPEN 1, "R", "NAMEPHON/RND:1"
20 FOR I% = 0 TO 5
30 FIELD 1,(I%*37) AS DM$, 25 AS NM$(I%),
12 AS PH$(I%)
40 NEXT I%
```

Line 10 opens the file as a random file. If the file already exists, the system finds it. If it doesn't, the system creates it. Line 20 determines that the program creates six subrecords. In line 30, the statement (I%*37) AS DM$ sets up a pointer to the beginning of each subrecord; the rest of the line describes each field within the record.

Program Listing 1 creates the random-access file described above to hold your data. Program Listing 2 reads the file sequentially. Whether you access the file randomly or sequentially, you always define fields as shown. You access the record using Get to call the full record. Then you use a file-handling program to locate the subrecord you want.

To write a record, you always fill the buffer using LSET (for left-justified data) or RSET (for right-justified data) with a specified subrecord position. These commands pad data that doesn't quite fill the buffer. Then you write the file to disk using Put with the main record number.

Indexing Your Files

As you can see, reading and writing a random file sequentially isn't too complicated. However, to read a file randomly, your program must have a way to determine the main record number (which I call MR%) and the subrecord number (SR%) of the file in which you're interested. In other words, you need an index of the records stored in the random-access file. I use a sequential index file to point to the random records. It's easy to use and control, and has the advantage of letting you search the index by more than one key word.

One important aspect of random-access files to remember is that fields in random files are absolute in size; you use LSET or RSET to fill the fields to the right or left, with spaces added to fill the buffer. (A sequential file's fields generally vary in size with the data and use a delimiter, like a comma, to define the fields.)

Program Listing 3 demonstrates my indexing technique with a program that catalogues my applications programs. Listing 3 stores data in the fields shown in Table 2. It includes entries for the program name, author, application, and the disk on which it's stored. You could use the same procedure to create a file of names and phone numbers, candidates and votes received, and so on. I've listed all variables at the end of the

The Key Box

Models I and III
32K RAM
Disk Basic
NEWDOS80 (for sort)

Command	Description
Field	Defines buffer
LSET/RSET	Positions data in buffer
Put	Writes the buffer
Get	Reads the buffer
LOF	Sets pointer to last main record of file

Table 1. Commands for manipulating random files.

Position	Length (in bytes)	Description
1–12	12	Program name
13–16	4	Application
17–24	8	Disk name
25	1	Side
26–27	2	Grans
28–37	10	Author
38–57	20	Description

Table 2. Fields for program reference file.

The index file controls access to the random file using the program's name as the key.

program; they're for reference and shouldn't be typed in.

When you run the program, it displays a menu of three options: adding a new file, changing or deleting a file, or exiting the program. As you update records, the program creates a sequential index file in memory using arrays. The program writes out this index file at the end of the job; the file controls access to the random file and uses the program name as the key field.

Each index file record has the format XXXXXXXX/XXX, MR%, SR%; that is, the program name, the main record number (file or buffer number) and the subrecord number. The variables MR% and SR% of the index file point straight to the random record for any specific program name. Table 3 describes the main routines of Listing 3.

Program Listing 4 creates, sorts, and writes two index files from the main random file in Program Listing 3, one using the program name as the key word, the other using the application. Don't sort or change the index file; it controls the update procedure. The sort in lines 240 and 270 is a system sort available under NEWDOS80 2.0. Alternatively, you can insert a sort for your own DOS (for a TRSDOS sort, see "Proper Arrangements," June 1984, p. 96).

Program Listing 5 searches the random file in Program Listing 3 using the two index files created in Listing 4. By creating an index file that can have any field as the key, you can sort the file or access records in any sequence you want.

This technique has saved me the trouble of finding a good algorithm for randomizing the key field (which would have to be unique) into a main record and a subrecord. I feel that I have better control, and the pointers are so simple that they're less likely to get out of sync and send all my data into never-never land. This method should help programmers manipulate random files more easily and confidently. ■

You can reach Jane Goodale at 828 N. 121st St., Omaha, NE 68154.

Lines	Description
10–20	Initialize variables.
30	Read index file. If it doesn't exist, On Error routine takes care of it.
40–50	Initialize and field random file buffer. .
70–90	Menu.
100–190	Add to file. Last record added will be displayed first. To duplicate a field from previous record press the enter key. If a deleted record exists (* in program name) the record is put there. Otherwise it's added at the end.
200–260	Change or delete. Asks for program name, displays the record, and positions cursor under each field in turn. Change field by entering new data, or leave as is by pressing enter. Delete record by typing asterisk (*) in the program name field. Line 260 returns to line 230 where I% is stepped up to find the next occurrence of the same program name, which handles the duplicate key.
400–420	Locate the first deleted record.
500–570	Set variables for input routine.
600	Get data from the random file buffer based on subrecord number XS%.
700–720	Fill random file buffer.
750–760	Fill unused subrecords with x's.
800–810	On Error routine.
900–920	End of job. Writes index file. At this point, I run Listing 4 to create two more index files.
1000–1140	William Barden's Universal Gee Whiz Input routine. See *Programming Techniques for Level II Basic*, p. 57, for full discussion.

Table 3. Main routines for Program Listing 3.

Program Listing 1. Creates a random-access disk file. Add lines 10–40 from p. 106.

```
100  MR%=1 ' initialize main record number
110  FOR SR%=0 TO 5 ' six subrecords to fill
120  INPUT"ENTER NAME ( '*' TO EXIT );N$
130  IF N$="*" THEN GOTO 210
140  INPUT"ENTER PHONE";P$
150  LSET NM$(SR%)=N$ ' fill buffer
160  LSET PH$(SR%)=P$
170  NEXT SR% ' step up subrecord number
180  PUT 1,MR% ' when six are filled, then Put
190  MR%=MR%+1 ' now step up main record number
200  GOTO 110 'and loop back
210  IF SR%=0 THEN 230 'last record
220  PUT 1,MR%' otherwise write it out
230  CLOSE 1:END
```

End

Program Listing 2. Reads a random file squentially. Add lines 10–40 from p. 106.

```
100  FOR MR%=1 TO LOF
120  GET 1,MR%
130  FOR SR%=0 TO 5
140  PRINT NM$(SR%),PH$(SR%) 'subscripted fields
150  NEXT SR%
160  NEXT MR%
170  CLOSE 1: END
```

End

Program Listing 3. Creates indexed random files.

```
1 REM RANDISKU/BAS                    JN GOODALE
2 REM       CREATE AND/OR UPDATE A RANDOM FILE OF PROGRAM
3 REM       AND DATA INFORMATION ON DISKS
8 REM
9 REM     INITIALIZE ----------------------------------------
10 CLEAR25000:CLS:PRINTTAB(15)"UPDATE PROGRAM INFORMATION FILE":PR
INT@340,"INITIALIZING":DEFINTA-Z:DIMIP$(500),IM%(500),IS%(500):E$=
"##":H2$="PROGRAM      APPL DISK     S GR AUTHOR    DESCRIPTION":
H3$=STRING$(63,"=")
20 MR%=1:SR%=0:N%=0:FI$="DISKINDX/INX:1":FR$="DISKINDX/RND:1":ONER
RORGOTO800:OPEN"I",1,FI$
29 REM     READ INDEX -------------------------------------
```

Listing 3 continued

Listing 3 continued

```
30 INPUT#1,IP$(N%),IM%(N%),IS%(N%):IFEOF(1)THEN40ELSEN%=N%+1:GOTO3
0
39 REM    INITIALIZE RANDOM FILE BUFFERS ------------------
40 CLOSE1:MR%=IM%(N%):SR%=IS%(N%):GOSUB420
50 OPEN"R",1,FR$:FIELD1,255 AS DX$:LSET DX$=STRING$(255,"X"):FORI%
=0TO3:FIELD1,(I%*57)AS DM$,12 AS PG$(I%),4 AS AP$(I%),8 AS DN$(I%)
,1 AS SD$(I%),2 AS GR$(I%),10 AS AU$(I%),20 AS DS$(I%):NEXTI%:IFN$
="Y"THEN100
69 REM   MAIN MENU -------------------------------------------------
70 CLS:PRINTTAB(15)"UPDATE PROGRAM INFORMATION FILE":PRINT:PRINTTA
B(20)"1)  ADD TO FILE":PRINTTAB(20)"2)  CHANGE/DELETE":PRINTTAB(20
)"3)  END PROGRAM":PRINT:PRINTTAB(10)"     .... CHOICE (DO NOT HIT
<ENTER>) ";
80 K$=INKEY$:IFK$=""THEN80ELSEPRINTK$:ONVAL(K$)GOTO100,200,900
90 GOTO70
99 REM   ADD TO FILE ---------------------------------------------
100 IFN$="Y"THEN120ELSEIFD$="Y"THEN120
110 GET1,MR%:XS%=SR%:GOSUB600:N%=N%+1:IFSR%=3THENSR%=0:MR%=MR%+1EL
SESR%=SR%+1
120 CLS:PRINTTAB(03)"ADD TO FILE : ('*' TO ESCAPE,    <ENTER> TO R
EPEAT)":PRINTH2$:PRINTH3$
130 IFD$="Y"THENGOSUB400:GOTO100ELSEIFN$="Y"THEN145
140 PRINTPG$;TAB(13)AP$;TAB(18)DN$;TAB(27)SD$;TAB(29)USINGE$;GR%;:
PRINTTAB(32)AU$;TAB(43)DS$
145 FORI%=2TO13:ZX=I%*64+128:GOSUB500:IFZA$="*"THEN180
150 XS%=SR%:XM%=MR%:X%=N%:GOSUB700:IFSR%=3THENPUT1,MR%:MR%=MR%+1:S
R%=0ELSESR%=SR%+1
160 N$="N":N%=N%+1:IFN%>500THENN%=500:GOTO900
170 NEXTI%:GOTO120
180 N%=N%-1:IFSR%>0THENGOSUB750:SR%=SR%-1ELSEMR%=MR%-1:SR%=3
190 GOTO70
199 REM  CHANGE/DELETE ----------------------------------------
200 CLS:PRINTTAB(24)"CHANGE/DELETE":PRINTTAB(12)"'*' TO DELETE";TA
B(31)"<ENTER> TO LEAVE AS IS":PRINTTAB(18)"PROGRAM NAME :";:F$=" N
OT FOUND"
210 ZZ=12:ZC=161:GOSUB1000:IFZA$="*"THEN70ELSEPG$=ZA$:PRINT" ":PG$
=PG$+STRING$(12-LEN(PG$)," ")
220 FORI%=0TON%:IFPG$=IP$(I%)THEN240
230 NEXTI%:PRINT" ":PRINTTAB(18)"PROGRAM ";PG$;F$:INPUT"......<ENT
ER> TO CONTINUE";K$:GOTO70
240 XM%=IM%(I%):XS%=IS%(I%):GET1,XM%:X%=I%:GOSUB600:PRINT@192,H2$:
PRINTH3$:PRINTPG$;TAB(13)AP$;TAB(18)DN$;TAB(27)SD$;TAB(29)USINGE$;
GR%;:PRINTTAB(32)AU$;TAB(43)DS$
250 ZX=384:GOSUB500:IFZA$="*"THENPG$="*":D$="Y":DM%=XM%:DS%=XS%:DR
%=I%
260 GOSUB700:PUT1,XM%:F$=" FINISHED":GOTO230
399 REM  LOCATE DELETED RECORD --------------------------------
400 D$="N":ZX=192:GOSUB500:IFZA$="*"THEND$="Y":GOTO440
410 XS%=DS%:XM%=DM%:X%=DR%:GET1,DM%:GOSUB700:PUT1,DM%
420 FORI%=0TON%:IFIP$(I%)="*"THEND$="Y":DM%=IM%(I%):DS%=IS%(I%):DR
%=I%:GOTO440
430 NEXTI%
440 RETURN
499 REM  SCREEN LAYOUT AND INPUT ------------------------------
500 ZZ=12:ZC=ZX:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENPG$=ZA$
510 ZZ=4:ZC=ZX+13:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENAP$=Z
A$
520 ZZ=8:ZC=ZX+18:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENDN$=Z
A$
530 ZZ=1:ZC=ZX+27:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENSD$=Z
A$
540 ZZ=2:ZC=ZX+29:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENGR%=V
AL(ZA$)
550 ZZ=10:ZC=ZX+32:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENAU$=
ZA$
560 ZZ=20:ZC=ZX+43:GOSUB1000:IFZA$="*"THEN570ELSEIFZA$<>""THENDS$=
ZA$
570 RETURN
599 REM  DEFINE DATA ------------------------------------------
600 PG$=PG$(XS%):AP$=AP$(XS%):DN$=DN$(XS%):SD$=SD$(XS%):GR%=CVI(GR
$(XS%)):AU$=AU$(XS%):DS$=DS$(XS%):RETURN
699 REM  FILL BUFFER ------------------------------------------
700 LSETPG$(XS%)=PG$:LSETAP$(XS%)=AP$:LSETDN$(XS%)=DN$:LSETSD$(XS%
)=SD$:LSETGR$(XS%)=MKI$(GR%):LSETAU$(XS%)=AU$:LSETDS$(XS%)=DS$
710 IP$(X%)=PG$(XS%):IM%(X%)=XM%:IS%(X%)=XS%
720 RETURN
749 REM   FILL UNUSED SUB RECORDS --------------------------
750 X%=N%+1:XM%=MR%:PG$="X":AP$="X":DN$="X":SD$="X":GR$="0":AU$="X
":DS$="X"
760 FORXS%=SR%TO3:GOSUB700:NEXTXS%:PUT1,MR%:RETURN
799 REM  ON ERROR ROUTINE ------------------------------------
800 IFERL=20THENN$="Y":RESUME50
810 PRINT"ERROR ";ERR/2+1;" IN LINE ";ERL:END
899 REM  END -------------------------------------------------
900 CLOSE1:OPEN"O",1,FI$:PRINT"CREATING RANDOM INDEX FILE"
910 FORI%=0TON%:PRINT#1,IP$(I%);",";IM%(I%);",";IS%(I%):NEXTI%:CLO
SE1
```

Listing 3 continued

Listing 3 continued

```
920 CLS:PRINT"END OF UPDATE .... NOW RUNNING PROGRAM INDEX":RUN"RA
NDISK1/BAS"
999 REM  BARDEN'S INPUT ROUTINE -----------------------------
1000 ZA$=STRING$(ZZ,"."):PRINT@ZC,ZA$;CHR$(14);:ZA=LEN(ZA$):FORZF=
1TOZA:PRINTCHR$(24);:NEXTZF:ZA$=STRING$(ZA,32):ZG=1
1010 ZH$=INKEY$:IFZH$=""THEN1010
1020 IFZH$=CHR$(8)ORZH$=CHR$(9)ORZH$=CHR$(13)GOTO1040
1030 IFZH$<CHR$(32)GOTO1010
1040 IFZH$<>CHR$(8)GOTO1070
1050 IFZG=1GOTO1010ELSEPRINTCHR$(24);
1060 ZG=ZG-1:GOTO1010
1070 IFZH$<>CHR$(9)GOTO1090
1080 IFZG>=ZAGOTO1010ELSEPRINTCHR$(25);:ZG=ZG+1:GOTO1010
1090 IFZH$<>CHR$(13)GOTO1100ELSEPRINT@ZC,CHR$(15);STRING$(ZA+1,32)
;:PRINT@ZC,ZA$;:ZA$=LEFT$(ZA$,ZG-1):RETURN
1100 ZG=ZG+1
1110 IFZG>ZA+1PRINTCHR$(24);
1120 IFZG>ZA+1ZG=ZG-1
1130 PRINTZH$;:IFZH$=","THENZH$=CHR$(129)ELSEIFZH$=":"THENZH$=CHR$
(130)
1140 ZA$=LEFT$(ZA$,(ZG-2))+ZH$+RIGHT$(ZA$,ZA-ZG+1):GOTO1010
1200 REM
1210 REM                     VARIABLE LIST
1220 REM   VARIABLE          DESCRIPTION
1230 REM   --------          ----------------------------------
1240 REM     AP$()           APPLICATION IN SUB RECORD
1250 REM     AU$()           AUTHOR IN SUB RECORD
1260 REM     D$              DELETE SWITCH
1270 REM     DM$             DUMMY RECORD FOR BUFFER
1280 REM     DM%             MAIN RECORD NUMBER OF DELETED RECORD
1290 REM     DN$()           DISK NAME IN SUB RECORD
1300 REM     DS$()           DESCRIPTION IN SUB RECORD
1310 REM     DS%             SUB RECORD NUMBER OF DELETED RECORD
1320 REM     DX$             DUMMY STRING FOR START-UP
1330 REM     E$              EDIT STRING
1340 REM     F$              COMMENT STRING ('NOT FOUND' OR 'FINISHED'
1350 REM     FI$             FILE SPEC OF INDEX FILE
1360 REM     FR$             FILE SPEC OF RANDOM FILE
1370 REM     GR$()           GRANS (OBTAINED FROM MKI$(GR%) IN SUB RECO
RD
1380 REM     GR%             GRANS
1390 REM     H2$             HEADING
1400 REM     H3$             HEADING
1410 REM     I%              TEMP - USED IN FOR NEXT LOOPS
1420 REM     IM%()           MAIN RECORD NUMBER IN INDEX FILE
1430 REM     IP$()           PROGRAM NAME IN INDEX FILE
1440 REM     IS%()           SUB RECORD NUMBER IN INDEX FILE
1450 REM     K$              TEMP - INKEY$
1460 REM     MR%             MAIN RECORD NUMBER
1470 REM     N$              NEW FILE SWITCH (="Y" IF NEW FILE)
1480 REM     N%              NUMBER OF RECORDS ON INDEX FILE
1490 REM     PG$()           PROGRAM NAME IN SUB RECORD
1500 REM     SD$()           SIDE IN SUB RECORD
1510 REM     SR%             SUB RECORD NUMBER
1520 REM     X%              TEMP - I% IN SUBROUTINE 700
1530 REM     XM%             TEMP - MR% IN SUBROUTINE 700
1540 REM     XS%             TEMP - SR% IN SUBROUTINE 700
1550 REM     ZA              TEMP - USED IN INPUT ROUTINE 1000
1560 REM     ZA$             TEMP - RESULTS OF INPUT ROUTINE
1570 REM     ZC              TEMP - CURSOR POSITION FOR INPUT ROUTINE
1580 REM     ZF              TEMP - USED IN INPUT ROUTINE
1590 REM     ZG              TEMP - USED IN INPUT ROUTINE
1600 REM     ZH$             TEMP - USED IN INPUT ROUTINE
1610 REM     ZX              TEMP - POINTS TO BEGINNING OF LINE FOR INP
UT
1620 REM     ZZ              TEMP - LENGTH OF FIELD FOR INPUT ROUTINE
```

End

Program Listing 4. Adds and sorts two indexes.

```
1 REM RANDISK1/BAS                  JN GOODALE           03/27/82
3 REM          CREATE KEY INDEX FILES FOR FIELDS
4 REM          OF A RANDOM FILE.
9 REM     INITIALIZE ..........................................
10 CLEAR23000:DEFINTA-Z:DIMIP$(500),IA$(500),IM%(500),IS%(500),S%(
500)
20 FR$="DISKINDX/RND:1":CLS:PRINT"READING PROGRAM INFORMATION FILE
"
50 OPEN"R",1,FR$:FIELD1,255 AS DX$:LSET DX$=STRING$(255,"X"):FORI%
=0TO3:FIELD1,(I%*57)AS DM$,12 AS PG$(I%),4 AS AP$(I%),8 AS DN$(I%)
,1 AS SD$(I%),2 AS GR$(I%),10 AS AU$(I%),20 AS DS$(I%):NEXTI%
100 N%=0:FORMR%=1TOLOF(1):GET1,MR%
110 FORSR%=0TO3
```

Listing 4 continued

Listing 4 continued

```
120 IM%(N%)=MR%:IS%(N%)=SR%:IP$(N%)=PG$(SR%):IA$(N%)=AP$(SR%)
130 N%=N%+1
140 NEXTSR%
150 NEXTMR%:N%=N%-1
199 REM     MAIN ROUTINE ....................................
200 CLOSE1:FI$="DISKINDX/INA:1":GOSUB240:W=1:W$="APPLICATION":GOSU
B300
210 FI$="DISKINDX/INP:1":GOSUB270:W=2:W$="PROGRAM":GOSUB300
230 GOTO900
240 CMD"O",N%+1,*S%(Ø),IA$(Ø):RETURN
270 CMD"O",N%+1,*S%(Ø),IP$(Ø):RETURN
299 REM     DO IT ...................................
300 PRINT:PRINT"WRITING : ";W$:OPEN"O",1,FI$
310 FORI%=ØTON%:S%=S%(I%)
320 ONWGOTO330,340
330 F$=IA$(S%):GOTO360
340 F$=IP$(S%):GOTO360
360 PRINT#1,F$;",";IM%(S%);",";IS%(S%)
370 NEXTI%:CLOSE1:RETURN
899 REM     END OF JOB ...................................
900 PRINT" ":PRINT"END OF JOB"
920 END
```

End

Program Listing 5. Searches random file using indexes created by Listing 4.

```
1 REM LOOKDISK/BAS                    JN GOODALE
3 REM             SEARCH AND DISPLAY RANDOM FILE OF PROGRAM
4 REM             INFORMATION USING A SEQUENTIAL FILE TO
5 REM             REFERENCE THE RECORDS
9 REM     INITIALIZE .........................................
10 CLEAR28000:CLS:PRINTTAB(15)"SEARCH/DISPLAY PROGRAM INFORMATON F
ILE":PRINT@340,"INITIALIZING":DEFINTA-Z:DIMIP$(500),IM%(500),IS%(5
00):E$="##":H2$="PROGRAM       APPL DISK     S GR AUTHOR      DESCRI
PTION":H3$=STRING$(63,"=")
20 FR$="DISKINDX/RND:1"
30 CLS:PRINTTAB(20)"SEARCH/DISPLAY PROGRAM":PRINT:PRINTTAB(15)"USI
NG ONE OF THE FOLLOWING KEYS :":PRINTTAB(20)"1)  PROGRAM":PRINTTAB
(20)"2)  APPLICATION"
40 PRINT:PRINT"ENTER CHOICE (DO NOT HIT <ENTER>) --> ";
50 K$=INKEY$:IFK$=""THEN50ELSEPRINTK$:ONVAL(K$)GOTO80,90
60 GOTO30
80 L=12:C$="PROGRAM NAME":C=1:FI$="DISKINDX/INP:1":GOTO120
90 L=4:C$="APPLICATION":C=2:FI$="DISKINDX/INA:1":GOTO120
120 N%=Ø:OPEN"I",1,FI$
130 INPUT#1,IP$(N%),IM%(N%),IS%(N%):IFEOF(1)THEN140ELSEN%=N%+1:GOT
O130
140 CLOSE1:OPEN"R",1,FR$:FIELD1,255 AS DX$:LSET DX$=STRING$(255,"X
"):FORI%=ØTO3:FIELD1,(I%*57)AS DM$,12 AS PG$(I%),4 AS AP$(I%),8 AS
 DN$(I%),1 AS SD$(I%),2 AS GR$(I%),10 AS AU$(I%),20 AS DS$(I%):NEX
TI%
150 CLS:PRINTTAB(15)"DISPLAY PROGRAM INFORMATION FILE":PRINT:PRINT
TAB(20)"1)  SEARCH":PRINTTAB(20)"2)  DISPLAY ALL":PRINTTAB(20)"3)
 RESTART":PRINTTAB(20)"4)  END":PRINT:PRINT"  .... CHOICE  (DO NOT
HIT <ENTER>) ..>> ";:
160 K$=INKEY$:IFK$=""THEN160ELSEPRINTK$:ONVAL(K$)GOTO180,240,460,4
70
170 GOTO150
180 CLS:PRINTTAB(20)"SEARCH ON : ";C$:PRINT
190 PRINT"ENTER ";C$;:INPUTS$:LC%=1:H1$=S$:LM%=Ø:GOSUB410
200 S$=S$+STRING$(L-LEN(S$)," "):FORI%=ØTON%:IFS$=IP$(I%)THENGOSUB
300
210 NEXTI%:GOSUB360:GOTO150
240 LC%=1:CLS:PRINTTAB(10)"DISPLAY FILE IN SEQUENCE BY :";C$:H1$=C
$
290 GOSUB410:LM%=Ø:FORI%=ØTON%:MR%=IM%(I%):SR%=IS%(I%):GOSUB310:NE
XTI%:PRINT"END OF FILE":GOSUB380:GOTO150
300 MR%=IM%(I%):SR%=IS%(I%)
310 IFMR%<>LM%THENGET1,MR%
320 LM%=MR%:XS%=SR%:GOSUB440
330 PRINTPG$;TAB(13)AP$;TAB(18)DN$;TAB(27)SD$;TAB(29)USINGE$;GR%;:
PRINTTAB(32)AU$;TAB(43)DS$
340 IFINT(LC%/10)=LC%/10THENGOSUB380:GOSUB410
350 LC%=LC%+1:RETURN
360 PRINT"END SEARCH ON ";S$:GOSUB380:RETURN
370 REM     CONTINUE OR QUIT PROMPT ...........................
380 PRINT@960,"..'Q' TO QUIT ........ <ENTER> TO CONTINUE ";
390 K$=INKEY$:IFK$=""THEN390ELSEIFK$="Q"THENGOTO150
400 RETURN
410 CLS:PRINTTAB(10)"PROGRAM INFORMATION FILE BY ";H1$
420 PRINT:PRINTH2$:PRINTH3$:RETURN
430 REM  DEFINE DATA -----------------------------------------
440 PG$=PG$(XS%):AP$=AP$(XS%):DN$=DN$(XS%):SD$=SD$(XS%):GR%=CVI(GR
$(XS%)):AU$=AU$(XS%):DS$=DS$(XS%):RETURN
450 REM     END OF JOB ....................................
460 RUN
470 CLOSE1:PRINT" ":PRINT"END OF JOB";
```

End

LOGICAL SYSTEMS *Catalog*

© 1984

1. The BASIC Answer *

2. BSORT/MOD324 **

3. DiskDisk *

4. FED II *

5. File Manager *

6. LSI HELP System *

7. LS-HOST/TERM **

8. LED *

9. LS-QFB/COMP **

10. LS-UTILITY Disk **

* Versions available for LDOS 5.1 and TRSDOS 6 (Specify when ordering).

** Available for TRSDOS 6 only.

LDOS 5.1 Only **$69.**

smal-LDOS Only **$39.**

TRSDOS is a registered trademark of the Tandy Corporation.

LDOS is a registered trademark of Logical Systems, Inc.

Prices and specifications subject to change without notice.

1. * **TBA** is a text pre-processor that translates structured BASIC "source" into interpreter executable programs. Supports long variable names and named references/procedures, along with conditional translation. **$79.**

2. ** **BSORT/MOD324** is an advanced machine language sort for M4 BASIC. Multiple key/tag arrays, 1 or 2 dimension, string or numeric arrays. MOD324 does most of the restructuring to allow M3 BASIC programs to be run under M4 BASIC. **$49.**

3. * **DiskDISK** allows the partitioning of large volume drives into smaller logical volumes which "look" just like additional floppy drives. This should be considered a must for hard disk or other "large" drive users. **$99.**

4. * **FED II** is an all-purpose File/Disk editor for LDOS or TRSDOS 6. Bytes may be modified in either ASCII or HEX. Location may be done via search string, load address or record number. Shows disassembly of /CMD files. **$49.**

5. * **File Manager** will handle all those COPY, MOVE, PURGE or BACKUP jobs with ease. Faster and more versatile than BACKUP or PURGE, **FM** can also produce a "script" for later execution through JCL. **$49.**

6. * The LSI **HELP System** is a complete series of packages to provide on-line help information to LDOS/TRSDOS 6 users. Packages are available for DOS/BASIC help, Technical help, and the development of additional "HELP" information files. Prices start at **$29.**

7. ** **LS-HOST/TERM** is a complete Mod 4 communications system. Includes ADDS25 terminal emulator, XMODEM file transfer utility for error-free transfer, HOST program with password protection and much, much more. Ideal for use with XENIX. **$199.**

8. * **LED** is the LSI full-screen text EDitor. Uses all LDOS special features (KSM, filters/drivers, ...). Excellent for editing KSM and JCL source files. T6 version includes "block write to disk" and "insert file at cursor". **$49.**

9. ** **LS-QFB/Comp** consists of two utilities. QFB performs a *fast* mirror-image backup of a disk, formatting the destination disk in the process. Comp will compare any two disks or files, with differences to display or printer. **$49.**

10. ** The **LS-Utility Disk** is a collection of the most popular LSI filters and utilities re-written for use under TRSDOS 6 on the Mod 4. Includes KSMPLUS, MAXLATE, CALC, READ40 and much, much more. **$49.**

Now pay only **$69** for the power of the **LDOS** operating system, or just **$39** for the **smal-LDOS** subset. Specify Model 1 or 3 when ordering.

Publications

LDOS Quarterly back issues are available, and are highly recommended for all LDOS users.

"**The Source**" is *The* commented source code for TRSDOS 6.2. A must for all TRSDOS 6 programmers. Complete three volume set is **$249.**

SHIPPING & HANDLING NOT INCLUDED

LOGICAL SYSTEMS INC.

301

TOLL FREE ORDERLINE **(800) 248-3535**

Write or call for our free catalog.

8970 North 55th Street / P.O. Box 23956 / Milwaukee, WI 53223

by Andy Levinson

PATCH WORK

You get a paging directory, pause control for program listings, full error messages, file identification with the Auto command, and more with these 13 patches to TRSDOS 1.3.

Tired of TRSDOS 1.3? If your answer is yes, take comfort in the fact that you're not alone. While Model III TRSDOS is a vast improvement over TRSDOS 2.3, it isn't perfect. I've developed a series of patches, however, that promise to smooth some of TRSDOS 1.3's rough edges.

The 13 patches included here make TRSDOS 1.3 a more efficient operating system. They give you an abridged TRSDOS directory, correct a TRSDOS bug for nonexistent files, display a directory one page at a time, pause a program or directory listing with the same command, eliminate the need for a colon with the Directory command, identify a file you load with the Auto command, list free disk space in detail, provide full error messages, refine the TRSDOS Ready prompt, change the TRSDOS boot-up banner, administer a better formatting test for new disks, and eliminate TRSDOS's "Operation Aborted" message. You don't have to install all the patches; rather, you can pick and choose as you please.

The Patches

You install the patches with the TRSDOS Patch command. It's simple to use. Start at TRSDOS Ready, type in the patches you want, check for typographical errors, and press the enter key to finish the command. If you get an error message, try to figure out what went wrong before continuing. You must copy all the command lines listed in the patch you select. For example, if the patch consists of four patch lines, you must enter all four lines, as four separate TRSDOS commands. Don't enter any other DOS commands until you've completed all patching.

Patch 1, the Catalogue command, gives you an abridged directory from the TRSDOS Ready prompt, eliminating the problems associated with TRSDOS 1.3's Directory command. Patch 1 eliminates TRSDOS's Route command and "Operation Aborted" message, so it doesn't use additional disk, directory, or memory space. The directory listing, however, isn't as complete as the original TRSDOS listing.

To install this patch correctly, first make sure that your TRSDOS 1.3 short directory routine in Basic works and then type in the patch. To call the new command, type CAT or CAT (drive number) from TRSDOS Ready, and press the enter key.

Note that section (a) of Patch 1 contains the two patches issued by Radio Shack (see *TRS-80 Microcomputing News*, October 1981, p. 13). Ignore section (a) if you've already incorporated this change. Type in the patch as listed:

(a) Radio Shack fix:

```
PATCH *10 (ADD=4E2E,FIND=CD3E4B,
  CHG=CD8A50)
PATCH *10 (ADD=508A,FIND=4469736B,
  CHG=4FC33E4B)
```

(b) The Catalogue command:

```
PATCH *1 (ADD=50A9,FIND=4F7065726174,
  CHG=0D7EFE343004)
PATCH *1 (ADD=50AF,FIND=696F6E204162,
  CHG=FE3030023E30)
PATCH *1 (ADD=50B5,FIND=6F727465640D,
  CHG=327142C31944)
PATCH *1 (ADD=51C9,FIND=524F55544520
  A842,CHG=434154202020AA50)
```

Patch 2 fixes a bug in TRSDOS 1.3 that results when TRSDOS tries to list nonexistent files in the disk directory hash index table. The code for Patch 2 is:

```
PATCH *10 (ADD=4E47,FIND=02,CHG=03)
```

Patch 3 modifies the Directory command so that it displays directory files one page at a time. Once you install this patch, type in DIR, press the enter key and the usual long-format directory appears. However, if there are too many entries for one screen, the listing stops and waits for you to press the enter key before displaying the next set of entries. Type in Patch 3 as listed here:

```
PATCH *6 (ADD=616D,FIND=CD8861B7C8
  FE40,CHG=3A803FD620C8CD)
PATCH *6 (ADD=6174,FIND=2806C547AF78
  C1,CHG=7C61CDC901AFC9)
```

Patch 4 applies to TRSDOS 1.3's List command. When you list a file from TRSDOS Ready, the display scrolls unless you pause the listing by pressing the @ key. This patch changes the pause control to the shift-@ key combination. The command is the same as that in Basic, and therefore much easier to remember. The code for Patch 4 is:

```
PATCH *11 (ADD=5265,FIND=40,CHG=60)
```

Patch 5 is similar to Patch 4 except that it applies to TRSDOS 1.3's Directory command. It lets you pause a long-format directory listing by pressing the shift and @ keys at the same time, rather than pressing the @ key alone. The reason for this patch is compatibility: It makes the pause control for both the Basic List and the Directory commands the same. Patch 5 shouldn't be applied if you've installed Patch 3. Here's the code for Patch 5:

```
PATCH *6 (ADD=6173,FIND=40,CHG=60)
```

Patch 6 gives you the option of including (or not including) a colon when naming the disk drive for the Directory command. The code is:

```
PATCH *6 (ADD=59EC,FIND=7EFE3A2015,
  CHG=2BCD781D38)
PATCH *6 (ADD=59F1,FIND=237ED630DA,
  CHG=06FE3A2010)
PATCH *6 (ADD=59F6,FIND=3452FE04D2,
  CHG=237ED630DA)
```

The Key Box

Model III
TRSDOS 1.3

Patch 7 works with the Auto command. When you use the Auto function, your computer displays the name of the file you're automatically loading, rather than just the message "Auto Function Engaged." Type in this patch as:

```
PATCH *0 (ADD=4F60,FIND=6B4F,
  CHG=2542)
```

Patch 8 not only tells you where you have free disk space, but also the amount of space, in grans, available. To make room for this patch, I had to modify the password message of the Prot (PW) command. After you add Patch 8, you're asked "New Master?" instead of "New Master Password?" The code for Patch 8 is:

```
PATCH *6 (ADD=5D52,FIND=FE28D0,
  CHG=CD4555)
PATCH *6 (ADD=5542,FIND=2050617373,
  CHG=3F2003FE28)
PATCH *6 (ADD=5547,FIND=776F72643F,
  CHG=D8F1C3FC5B)
```

Patch 9 is a one-byte patch that makes an entire error message appear rather than the error number. Once this patch is in place, you'll see the message without having to enter another command. The code for this patch is:

```
PATCH *4 (ADD=4E28,FIND=20,CHG=18)
```

Patch 10 eliminates the row of dots that TRSDOS 1.3 displays after the TRSDOS ready prompt. The code for this patch is:

```
PATCH *1 (ADD=4E75,FIND=063E3E,
  CHG=C37E4E)
```

Patch 11 shortens the TRSDOS 1.3 opening banner from a picture of the computer to a single line. It also skips the time and date prompts when you boot-up. A minor problem with this patch is that new files created usually have garbage for the date, although the file itself isn't affected. If this bothers you, install only the first three of the four patch lines for Patch 11. The code is as follows:

```
PATCH *0 (ADD=5039,FIND=20,CHG=0D)
PATCH *0 (ADD=508A,FIND=28,CHG=0D)
PATCH *0 (ADD=516C,FIND=D0,CHG=0D)
PATCH *0 (ADD=4EA9,FIND=CA,CHG=C3)
```

Patch 12 provides a better formatting test for new disks. Once you install Patch 12, you can expect more disks to fail the formatting test, but it's better to lose a marginal disk than to lose your important data. Here's the code for Patch 12:

```
PATCH *7 (ADD=5BEE,FIND=E5,CHG=5B)
```

Patch 13 eliminates the "Operation Aborted" message. Don't apply this patch if you've already installed Patch

Patching is an easy way to change a file; all you have to do is copy and enter a few lines of text.

1. The code for Patch 13 is:

PATCH *1 (ADD = 50A9,FIND = 4F,CHG = 0D)

One final note—the patches won't take effect until you reboot your disk.

Before Patching

When installing a new patch, always use a back-up disk as the Patch command permanently modifies the machine-language system programs stored on disk. Only patch more important disks when you're sure that the patches work as expected; a typographical error could ruin your disk in seconds. If you've applied non-Radio Shack patches to TRSDOS 1.3, the patches given here might not work.

One way to avoid errors is to create a do-file. With the Do and Build commands, you only have to type in the patches once. The do-file is then used to correctly patch as many disks as desired. The procedure for using these commands is explained in the TRSDOS 1.3 manual (also see Douglas Payne's "Brick by Brick," *80 Micro*, April 1984, p. 58).

Another option is to use a word processing program to create and edit do-files. A do-file is an ASCII file and most word processing programs work with ASCII files. All you have to do is write command lines, each with 63 characters (or fewer), and a carriage return at the end. Then save the file in ASCII format. You should select a file name ending with the /BLD extension. With a word processor, you can create a master file containing all of the patches listed here. From this master file you can delete and combine the various lines to create new, customized do-files that contain the specific patches that you want to install.

Understanding the Patch Command

The Patch command changes information at a specific location in a disk file. It's an easy way to change a file since all you have to do is copy and enter a few lines of text. The programmer, on the other hand, must first determine what changes you want made, and then must find available space in the program file to apply those changes. If the file is full, then you must overwrite existing Z80 computer instructions with the new patches. The command format for a patch is as follows: PATCH filename (ADD = XXXX,FIND = YY, CHG = ZZ).

The first item of information is the file name, necessary so that TRSDOS can find the file and apply the patch. Because TRSDOS 1.3 system files don't have file names, use the format, *#:$, to identify the decimal number (#) and disk drive ($) of that system file.

The second item of information in the Patch command indicates where you should make the changes. This information is provided by ADD = XXXX, the next variable on the command line. The "XXXX" is a hexadecimal number representing a memory address. The third item, "FIND = YY", reveals the contents of the disk file at that patch location. Again, the "YY" represents a hexadecimal number; it's possible to have more than one hexadecimal number in a Patch command.

You should know what's on the disk before installing a patch because TRSDOS 1.3 won't complete the patch if the numbers indicated in Find don't match those on the disk file. The advantage of patching with TRSDOS 1.3 is that while it may take longer, it's almost impossible to install a patch in the wrong location.

The final item of the patch format command, "CHG = ZZ", indicates the new information you want on disk. In doing this, TRSDOS 1.3 checks to make sure that the number of Find bytes you need are equal to the number of Find bytes available. If there's no match, the command is aborted with an error message. Otherwise TRSDOS writes the "CHG" bytes to the disk file where the Find bytes had been. The actual switch is made in memory and then written to disk as a modified file sector. For this reason, you can't write-protect your disk when you're patching a file. If you're skilled in hexadecimal math and machine language, you can make minor modifications to a machine-language program without having to re-create the file with an editor/assembler. ■

Write to Andy Levinson at 11575 Sunshine Terrace, Studio City, CA 91604.

HAS YOUR
COMPUTER SUPPLIER
DESERTED
YOU?

**Look no further.
Perry Computers can meet all
your needs for major brand items.**

Perry Computers has been in the computer business for seven years, and will be there when you need us. We don't believe in deserting you.

Shop With us Toll Free – Simply call our Toll Free number to place your order. Our knowledgeable & courteous staff will be glad to take your order and answer your questions.

Choose From Top Quality Merchandise – All merchandise sold by Perry Computer is name brand with manufacturer's warranty. All Items are the same as you would buy locally, 100% original, no upgrades. If a problem does occur, just call us on our **Customer Service Hotline 517-625-4268.**

FOR ORDERS CALL 1-800-248-3823

Watch For Monthly Specials – On a variety of systems and accessories. If you are stepping up to a more powerful system, expanding or a first time user, be sure to call about products and prices.

Name Brands – RADIO SHACK, EPSON, COLUMBIA, SANYO, COMREX, OKIDATA, TRANSTAR, STAR, AMDEK, BOTEK, ELEPHANT, HAYES, VERBATIM, JUKI, SCOTCH, GORILLA, VIDEO PLUS, TELEWRITER, VIP, TOM MIX, C.ITOH, SKC, MANNESMAN TALLY, ANCHOR SIGNALMAN, IDS, KEYTRONICS, MOUSE SYSTEMS, PRINCETON GRAPHICS, ANADEK, CARDCO, TOSHIBA, ORANGE MICRO, MICROBUFFER AND MORE. If you don't see what you're looking for, give us a call. If at all possible we'll get if for you.

PERRY COMPUTERS

13[illegible] N MAIN, P.O. BOX 156, PERRY, MI 48872

by Mike O'Neal

Formula Solutions

Cubic saves you time and frustration by solving complex equations.

Complex mathematical formulas have always intrigued me. However, solving these equations by trial and error, or by the quadratic formula, can become tedious and time-consuming. Why not let your computer do the work? I wrote Cubic to solve real or complex roots of cubic, quadratic, and linear algebraic equations.

Using Cubic

Type in and save Cubic (see Program Listing 1, 2, or 3, depending on your system). When you run the program, it displays the prompt seen in the Figure. Cubic always displays the formula for a cubic equation (where the greatest exponent is three) and prompts you to enter values for the variables A, B, C, and D.

To solve for a quadratic equation (where no exponent exceeds 2), you need to zero out the first term in the equation by entering a zero for the value of A.

For a linear algebraic equation (no exponent), enter a zero for both the coefficients A and B.

As an example, consider the cubic equation $X^3+3X^2-4X-12=0$. Coefficient A equals 1 because there's only one X to the power of three, B equals three because there are three X's squared, C equals negative four because four X's are subtracted, and D equals the constant -12. To solve this equation, you enter the values 1, 3, -4, -12, at the coefficient prompt.

Once you enter the coefficents, Cubic displays the real roots of the equation. There are three solutions to this sample problem: 2, -2, and -3.

Cubic then lets you check the veracity of these numbers by prompting you to substitute a number for X in the equation ("CHECK#?"). You can thus reenter and verify Cubic's solutions. This is handy for rechecking suspicious answers.

Consider this quadratic equation: $-2X^2+11X-15=0$. The coefficients here are $A=0$, $B=-2$, $C=11$, and $D=-15$. The first coefficient is zero because there's no X^3 term in the quadratic equation; any term not present is considered to have a coefficient of 0. The two solutions to this quadratic are $X1=2.5$ and $X=3$.

Complex Roots

Cubic also works with complex roots. Consider this equation: $X^2-6X+25=0$. The coefficients are 0, 1, -6, 25, while the solutions are the two complex conjugates $3+4i$ and $3-4i$, displayed

```
THIS PROGRAM FINDS ROOTS OF CUBIC EQUATIONS
OF THE STANDARD FORM: F(X) = A*X'3 + B*X'2 + C*X + D
PLEASE INPUT THE COEFFICIENTS A,B,C,D ? 1,3,-4,-12

THE FIRST REAL ROOT IS X0 =  2
THE OTHER 2 ROOTS ARE
X1 = -2
X2 = -3
Press <ENTER> to END program. CHECK #? -2
F (-2 ) =  0
Press <ENTER> to END program. CHECK #?
READY
```

Figure. Cubic prompts you to enter equation coefficients, then solves for the equation's roots.

Program Listing 1. Cubic for the Model I.

```
0 ' CUBIC/I VERS 2/15/84 by Mike O'Neal
30 PRINT"THIS PROGRAM FINDS ROOTS OF CUBIC EQUATIONS"
40 PRINT"OF THE STANDARD FORM: F(X) = A*X'3 + B*X'2 + C*X + D"
50 INPUT"PLEASE INPUT THE COEFFICIENTS A,B,C,D ";A3,A2,A1,A0
60 PRINT:IFA3<>0THEN100
70 IFA2<>0THENA=A2:B=A1:C=A0:PRINT"THE";:GOTO190
80 IFA1=0THEN50
90 PRINT"THE ONLY ROOT IS X = ";-A0/A1:END
100 'CUBIC
110 S=1:L=0:P=1
120 S=1-S:L=1-L:GOSUB300:IFYS<>YLTHEN140
130 S=-S:L=-L:GOSUB300:IFYS=YLTHEN120
140 H=(S+L)/2:GOSUB290:IFYS<>YHTHENL=HELSES=H
150 IFABS(L-S)<PTHENP=ABS(L-S):GOTO140
160 PRINT"THE FIRST REAL ROOT IS X0 = ";H:PRINT"THE OTHER";
```

Listing 1 continued

The Key Box

Models I, III, 4, 1000, and 2000
16K RAM Cassette Basic
32K RAM Disk Basic

Using the quadratic formula, Cubic solves the remaining equation.

as $3 \pm 4i$ (i stands for imaginary, meaning the square root of negative one).

To solve $X^2 = -9$, substitute 0, 1, 0, 9 into the program for the result $X = 3i$ or $X = -3i$.

Program Structure

Programmers will notice that Cubic uses the notation X*X*X to cube a number instead of X^3, which calls the computer's built-in exponent capability. My reasoning here is that it's more efficient for Basic to multiply the numbers when the exponent is small. With the built-in exponent function, Basic first must logarithmically convert both the X and the power, then multiply the two numbers together, taking the inverse log to obtain the result.

Cubic first must determine if it's solving a cubic, quadratic, or simple linear equation. Line 100 of all three listings initiates the search for the point where the function crosses the X-axis. Once the first zero is located, Cubic puts the equation into its quadratic form. Using the quadratic formula, Cubic then solves the remaining equation. If the equation is already quadratic, Cubic jumps to line 190; if it's a linear equation, Cubic uses simple algebra to solve for the answer at line 90.

Cubic Versions

I've included three versions of Cubic: The first version (Program Listing 1) is for the Model I Level II tape or disk system, the second version (Program Listing 2) is for Model III Disk Basic, the third version (Program Listing 3) applies to the Models 4, 1000, and 2000. Model 1000 and 2000 users must change the CHR$(127) in line 260 of Program Listing 3 to CHR$(241).

If you have a Model III tape system simply use Program Listing 1 and substitute line 260 of Program Listing 2 for line 260 of Program Listing 1. This lets you take advantage of the special algebraic symbol, the plus/minus. In addition, Cubic runs on minimum memory system as the Program Listings are only about 1K in length. ■

Write to Mike O'Neal at 220 N. College Avenue, #18, College Place, WA 99324.

Listing 1 continued

```
170 'SYNTHETIC DIVISION
180 A=A3:B=H*A+A2:C=H*B+A1
190 'QUADRATIC
200 PRINT" 2 ROOTS ARE ";
210 D=B*B-4*A*C:IFD<0THEN240
220 X1=(-B+SQR(D))/(2*A):X2=(-B-SQR(D))/(2*A)
230 PRINT:PRINT"X1 = ";X1:PRINT"X2 = ";X2:GOTO270
240 PRINT"THE COMPLEX CONJUGATES:":D=-D
250 XR=-B/(2*A):XI=SQR(D)/ABS(2*A)
260 PRINTXR"+"XI"I":PRINTXR"-"XI"I"
270 S=999999:INPUT"PRESS <ENTER> TO END PROGRAM. CHECK #";S
280 IFS=999999ENDELSEGOSUB310:PRINT"F ("S") = ";Y:GOTO270
290 YH=SGN(A3*H*H*H+A2*H*H+A1*H+A0)
300 YL=SGN(A3*L*L*L+A2*L*L+A1*L+A0)
310 Y=A3*S*S*S+A2*S*S+A1*S+A0:YS=SGN(Y):RETURN
```

End

Program Listing 2. Cubic for the Model III.

```
0 ' CUBIC/III VERS 2/15/84 by Mike O'Neal
10 DEFFNA(X)=A3*X*X*X+A2*X*X+A1*X+A0
20 DEFFNB(X)=SGN(FNA(X)):CC=32
30 PRINT"THIS PROGRAM FINDS ROOTS OF CUBIC EQUATIONS"
40 PRINT"OF THE STANDARD FORM: F(X) = A*X'3 + B*X'2 + C*X + D"
50 INPUT"PLEASE INPUT THE COEFFICIENTS A,B,C,D ";A3,A2,A1,A0
60 PRINT:IFA3<>0,100
70 IFA2<>0,A=A2:B=A1:C=A0:PRINT"THE";:GOTO190
80 IFA1=0,50
90 PRINT"THE ONLY ROOT IS X = ";-A0/A1:END
100 'CUBIC
110 S=1:L=0:P=1
120 S=1-S:L=1-L:IFFNB(S)<>FNB(L),140
130 S=-S:L=-L:IFFNB(S)=FNB(L),120
140 H=(S+L)/2:IFFNB(S)<>FNB(H),L=HELSES=H
150 IFABS(L-S)<P,P=ABS(L-S):GOTO140
160 PRINT"THE FIRST REAL ROOT IS X0 = ";H:PRINT"THE OTHER";
170 'SYNTHETIC DIVISION
180 A=A3:B=H*A+A2:C=H*B+A1
190 'QUADRATIC
200 PRINT" 2 ROOTS ARE "
210 D=B*B-4*A*C:IFD<0,240
220 X1=(-B+SQR(D))/(2*A):X2=(-B-SQR(D))/(2*A)
230 PRINT"X1 = ";X1:PRINT"X2 = ";X2:GOTO270
240 PRINT"THE COMPLEX CONJUGATES: ";:D=-D
250 XR=-B/(2*A):XI=SQR(D)/ABS(2*A)
260 PRINTXR;CHR$(127);STR$(XI);CHR$(105)
270 S=999999:INPUT"Press <ENTER> to END program. CHECK #";S
280 IFS=999999ENDELSEPRINT"F ("S") = ";FNA(S):GOTO270
```

End

Program Listing 3. Cubic for the Models 4, 1000, 2000.

```
0 ' CUBIC/4
10 DEF FN A(X)=A3*X*X*X+A2*X*X+A1*X+A0
20 DEF FN B(X)=SGN( FN A(X)):CC=32
30 PRINT "THIS PROGRAM FINDS ROOTS OF CUBIC EQUATIONS"
40 PRINT "OF THE STANDARD FORM: F(X) = A*X'3 + B*X'2 + C*X + D"
50 INPUT "PLEASE INPUT THE COEFFICIENTS A,B,C,D ";A3,A2,A1,A0
60 PRINT:IF A3<>0 THEN 100
70 IF A2<>0 THEN A=A2:B=A1:C=A0:PRINT "THE";:GOTO 190
80 IF A1=0 THEN 50
90 PRINT "THE ONLY ROOT IS X = ";-A0/A1:END
100 'CUBIC
110 S=1:L=0:P=1
120 S=1-S:L=1-L:IF FN B(S)<> FN B(L) THEN 140
130 S=-S:L=-L:IF FN B(S)= FN B(L) THEN 120
140 H=(S+L)/2:IF FN B(S)<> FN B(H) THEN L=H ELSE S=H
150 IF ABS(L-S)<P THEN P=ABS(L-S):GOTO 140
160 PRINT "THE FIRST REAL ROOT IS X0 = ";H:PRINT "THE OTHER";
170 'SYNTHETIC DIVISION
180 A=A3:B=H*A+A2:C=H*B+A1
190 'QUADRATIC
200 PRINT " 2 ROOTS ARE "
210 D=B*B-4*A*C:IF D<0 THEN 240
220 X1=(-B+SQR(D))/(2*A):X2=(-B-SQR(D))/(2*A)
230 PRINT "X1 = ";X1:PRINT "X2 = ";X2:GOTO 270
240 PRINT "THE COMPLEX CONJUGATES: ";:D=-D
250 XR=-B/(2*A):XI=SQR(D)/ABS(2*A)
260 PRINT XR;CHR$(127);STR$(XI);CHR$(105)
270 INPUT "CHECK #";S
280 PRINT "F ("S") = "; FN A(S):GOTO 270
```

End

NEW!
IN STOCK!!

NO WAITING!!!

$777
5 MEG HARD DRIVE

$10 handling & shipping

BEWARE!!

Would you buy from *this* hard drive specialist?

Comes complete with cable and your choice of software driver * (CP/M, LDOS, TRSDOS)

10 Megabyte Hard Drive............. $995
15 Megabyte Hard Drive............. 1195
30 Megabyte Hard Drive............. 1495

SYSTEM FEATURES

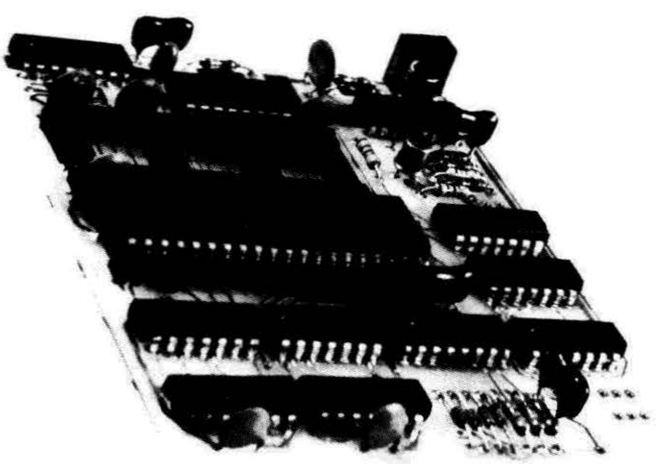
- For TRS-80 Model 3-4-4P
- Size Rated After Format
- Error Checking & Correcting

*Model 3 requires LDOS
Model 4 requires TRSDOS 6.2 or Montezuma Micro CP/M 2.2

Seriously folks, there are firms that offer benefits or products seemingly too good to be true. There's no reason to expose yourself to possible unhappiness. Aerocomp has proven record of thousands of satisfied TRS-80® customers. Look at the back issues of this magazine. You won't find many companies that have been around as long as Aerocomp. We fully support TRS-80® computers and most all operating systems including CP/M® 2.2. Aerocomp leads the way to low hard prices so you can afford to enjoy the benefits of increased storage and faster disk I/O. These units are precision engineered, tested and delivered complete and ready to use, right from stock. Guaranteed for one year parts and labor. You can count on us to be here if you need us. As always, your satisfaction is assured with our 14 day free trial offer. How can you wrong? Specify the software driver of your choice and start enjoying your computer's real capability. Do it today!

See opposite page▸▸▸▸▸▸

"DDC"
Double Density Controller



Add 80% more capacity to your disk system with the Double Density Controller (DDC) from Aerocomp.

The Story

Some products have what it takes to seem to last forever. Our "DDC" is one of those products. What it does is allow you to operate your TRS-80 Model 1 disk system in double density. In this case double density means almost doubling the storage capacity of your diskettes. Single density, thats the way Radio Shack designed your Model I expansion interface, organizes your disk into 10 sectors per track. Each sector contains 256 bytes of data for a total storage capacity of 2,560 bytes or 2.5K per track times the number of tracks your drive is capable of addressing. Double density,on the other hand, writes 18 sectors per track each containing 256 bytes for a total of 4,608 bytes or 4.5K. That 80% more data in the same space. Why didn't Radio Shack do that in the beginning, you ask? Well it costs money to do double density because it is more difficult to do than single density and the data is harder to capture reliably. That means more cost and the Model I was meant to be a low-cost computer for the masses. Therefore, no double density.

The Facts

Other companies introduced double density controllers for the Model I but they were not so good. We waited and waited but, even new models failed to correct problems with data separation that kept cropping up. So we went to work and came up with a new design to cure the old problem. At last! A double density controller for the Model I with a higher probability of data recovery than with any other double density controller on the market then or since. Our analog design phase lock loop data separator has a wider capture window than the digital types the others use. This allows high resolution data centering. Our "DDC" analog circuit allows infin itely variable tuning. The attack and settling times are optimum for 5.25" diskettes. The oft-stated fears of adjustment problems have been proved groundless by thousands of satisfied users the world over. The bottom line here is state-of-the-art performance and reliability.

ORDER YOURS TODAY

TRS-80 Model I disk system owners who are ready for reliable double density operation will get 80% more storage per diskette; single and double density data separation with far fewer disk I/O errors; single density compatibility; simple plug-in operation. You will need a disk operating system that has the necessary double density software. All the popular DOS's (except TRSDOS) have the drivers. We have put together two special packages in the event you don't already have one of the more popular DOS's.

"DDC" by itself........................ $99
"DDC" and LDOS 159
"DDC" and NEWDOS 80 v2.0 189

Please add $4 handling & shipping

- ☞ **40 Track**
- ☞ **80 Track**
- ☞ **Single Head**
- ☞ **Dual Head**
- ☞ **Bare**
- ☞ **Complete**
- ☞ **Full Size**
- ☞ **Half Size**
- ☞ **5-1/4" or 8"**

DISK DRIVES

Aerocomp leads the way to the BEST value in disk drives and related peripheral products on the market today. Sound engineering, high performance, quality construction, no-risk free trial, outstanding warranty service and a reputation for doing the right thing make your decision to buy Aerocomp the correct one. Please look over our offerings and make your selection. When you have made your choice call our toll-free number and place your order. If you need assistance in making your selection, please call our information number. It's listed in the box at the bottom of this ad along with the technical assistance number for those of you who want to get right to the nitty-gritty. Thanks, we all appreciate your business and will continue to do our very best to support you.

COMPLETE DRIVES Add $6 S&H

Fits TRS-80 Models 1,3,4 and the Color Computer plus the others that use standard drives. The drive of your choice is mounted in a sturdy, all-steel cabinet. An external gold plated drive connector allows cabling without disassembly of the cabinet. Half-high drives come in a full-size cabinet that will hold and power our half-high drives. Single drives have a panel covering the unused space allowing a second drive to be added at any time. All are shipped fully assembled ready to use. Specify silver or white enclosure.

40tk Single Side full size (TM100-1)........... $ 169
40tk Double Side full size (TM100-2)............ 209
80tk DS full size (TM101-4)........................... 299
1-40tk SS half-high FD-55A in dual case 179
2-40tk SS half-high FD-55A's in dual case.... 319
1-40tk DS half-high FD-55B in dual case 219
2-40tk DS half-high FD-55B's in dual case .. 399
1-80tk DS half-high FD-55F in dual case...... 249
2-80tk DS half-high FD-55F's in dual case... 449

These 8" Thinline drives work with the Model 2 and 12 plus others that use standard drives. The rugged all-steel cabinet has an extra heavy duty power supply rated for continuous operation. A removeable air filter allows only clean air to circulate. Cabinets with single drives are supplied with a blank panel to cover the unused space. A second drive can be added at any time. Add $12 shipping and handling.

1-8" SS Tandon TM848-1E & case........... $ 389
2-8" SS Tandon TM848-1E's & case........... 649
1-8" DS Tandon TM848-1E & case............. 499
2-8" DS Tandon TM848-2E's & case.......... 799

BARE DRIVES Add $4 S&H

These drives are completely burned-in and tested for your ultimate satisfaction. Add that extra drive today! Please add $5 for shipping and handling for each drive ordered.

40tk SS full size Tandon TM100-1 .... $ 129
40tk DS full size Tandon TM100-2 169
40tk SS half high TEAC FD55-A.................. 129
40tk DS half high TEAC FD55-B.................. 169
80tk DS full size Tandon TM101-4 239
80tk DS half high TEAC FD55-F 239
8" SS half high Tandon TM848-1E............... 290
8" DS half high Tandon TM848-2E............... 360

TRS-80 Model III & 4 DISK CONTROLLER and DRIVE KITS

Convert your cassette Model III or 4 to disk operation with one of our complete kits. You receive our own advanced disk controller board with gold plated edge contacts capable of 4-drive operation; our own power supply; plated steel mounting towers complete with RFI shield plus all the cables and hardware necessary. Detailed instructions are included. All you need is a screwdriver and a pair of pliers. System kits come with 40 track single-side Tandon drives (TM100-1) or just order the basic kit and pick the drives you want from the selection in the next column.

CONTROLLER KIT ....$ 199
(Complete - Everything you need - less drives)
1 DRIVE SYSTEM............... 319
2 DRIVE SYSTEM............... 449
CONTROLLER ONLY........... 110
MOUNTING KIT & POWER SUPPLY.....................95
RS-232 BOARD.......................69

MISCELLANEOUS GOODIES $2 S&H

Model 1 TRSDOS 2.3 disk & manual.........$ 25
Model III TRSDOS 1.3 disk & manual 25
LDOS (specify Model I or III)..........................69
NEWDOS 80 v2.0 (specify Model I or III)....119
10 disks in smoke lib box, Lifetime Guar..... 16
5.25" drive power supply & enclosure...........59
8" drive power supply, encl. & fan, 5V-24V 150
5.25" 2-drive cable..24
5.25" 4-drive cable..34
5.25" Extender cable with gold contacts.......13

FREE TRIAL OFFER

Use your Aerocomp hardware product for up to 14 days. If you are not satisfied for **ANY REASON** (except misuse, damage or improper handling), return it (insured) in the original shipping container for a full purchase price refund, less shipping. **Sorry, this offer does not apply to software. Defective software will be replaced only.** Any hardware/software specials will be prorated and the software will be charged at the regular unbundled price. we have confidence in our products and we know you will be satisfied.

WARRANTY

We offer a one year warranty on parts and labor against defects in materials and workmanship. In the event service becomes necessary for any reason you will find our service department fast, friendly and cooperative. We want to keep you happy. Out of warranty repairs are also available.

OUR FAMOUS MODEL I STARTER PACKAGE

If you have a Model I and an Expansion Interface this is what you need to get started with disks. Included is one 40 track single-side disk drive complete with matching silver case and power supply, a 2-drive cable, a TRSDOS 2.3 disk operating system and TRSDOS manual plus all insurance and delivery charges to your door (lower 48 states).

Yours for only **$239**

You can add our renown "DDC" double density controller to either the Radio Shack or the LNW E/I for 80% more storage capacity on your drive. Go ahead, you deserve increased density. See the opposite page for the latest details.

$99

100% BURN-IN and TEST
All our products are burned-in and fully tested prior to shipment. We want you to receive an item ready-to-go. AEROCOMP means reliability!

ORDER NOW!
Call our toll-free number service and place your order. Have your American Express, Mastercharge or Visa number ready. We will not charge your card until the day we ship your order. You may order by mail using your credit card, check or money order. Personal and company checks are welcome and cause no shipping delay as long as they are bank printed and the signature exactly agrees with the name printed on the check. We will ship surface COD with no deposit but all COD's require cash or a cashier's check on delivery. Texas residents add 6% State Sales Tax. No tax collected on out of state shipments. Canadians add $20 to your orders if over $500 for customs documentation.

CALL TOLL-FREE

800-527-3582 USA
800-442-1310 TEXAS

For inquiries or information or to check on or change an order call 214-339-8324

AEROCOMP
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

✓82

✓ See List of Advertisers on Page 145

Desirable Interruptions: A I/III/4 Interrupt Controller

"Pardon the interruption...." While interruptions are considered impolite, sometimes you have to make them, especially in microcomputer systems. Many computers, particularly those involved with real-time or time-sharing applications, require microprocessor interrupts to make proper operation possible.

An interrupt allows an external device to have the processor stop what it's doing and take time to service the external device. If the processor isn't doing something more important (determined by a combination of hardware and software), the processor complies with the interrupt. If the processor is working on something more important (a higher-priority device) it ignores the interrupt request and continues with what it was doing.

Many systems (like the TRS-80 Models I, III and 4), use polling for input/output (I/O) operations to see when you've pressed the keyboard or when it's OK to send a character to the printer. Polling is simply a check to see if a device is ready or needs servicing. In a polling system, the processor continually loops to check the status of various devices. This is fine as long as the processor has nothing else to do. But it can often be doing useful work while waiting on I/O devices. Interrupts allow the I/O devices to interrupt the processor when they need service, permitting it to do useful work in the meantime.

This month's project is an interrupt controller compatible with the Models I, III and 4. The controller board uses an 8259A interrupt controller integrated circuit (IC), which does most of the work on the board.

The Key Box

Models I, III, and 4
16K RAM
Assembly Language
Editor/Assembler

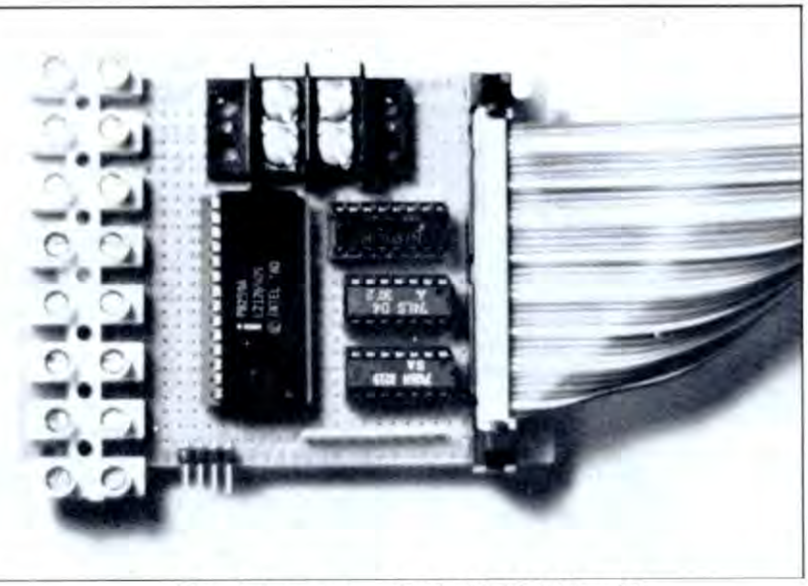


Photo 1. Interrupt controller for Models I, III, and 4.

A Look At Z80 Interrupts

The Z80 has two interrupt input lines (INT/ and NMI/) and uses a combination of two signals to acknowledge an interrupt. The INT/ line is the general-purpose interrupt input, the one most commonly used. The NMI/ line, or nonmaskable interrupt line, is reserved for special emergency situations, such as a power failure.

You can enable or disable (mask) the INT/ line under software control, using the Z80's Enable Interrupt (EI) and Disable Interrupt (DI) instructions. This allows the software to determine when to accept interrupts and when not to. The NMI/ input, on the other hand, is nonmaskable, meaning that you can't disable it with software. Because of this, it's rarely used.

The Z80 has three interrupt modes, mode 0, mode 1 and mode 2, which you can use with the INT/ interrupt input. These modes give the Z80 a considerable amount of interrupt flexibility, allowing effective interrupt use in a variety of system applications. I'll discuss how the Z80 responds to each of the interrupt modes separately, including the best uses for each one.

Z80 Mode 0 Interrupts—8080A-Compatible Mode

Interrupt mode 0 is the default mode on processor reset. The Z80 responds to its interrupt input in the same manner as its predecessor, the 8080A. The Z80 looks at the interrupt input at the end of each instruction. If it's active (low) and interrupts are enabled, the processor stops its execution and begins servicing the interrupt request. The response will vary depending on the interrupt mode you're using.

In the case of mode 0, the Z80 sends an interrupt-acknowledge signal to the interrupting device when it accepts an interrupt. It does this by simultaneously bringing the M1/ and IORQ/ signals low. The interrupt-acknowledge signal acts as a read strobe to the interrupting device, telling it to put the opcode (first byte) of an instruction on the data bus. This is a Call instruction pushing the processor's program counter (PC) onto the stack and branching the execution to the beginning of the interrupt service routine.

The 8080A (like the Z80) provides a set of 1-byte Call instructions ideal for this purpose. These eight instructions, called Restart instructions, make a call to a predetermined location in memory (usually occupied by ROM). Because of their bit makeup, you can easily generate them with hardware. The instructions to do so, along with the binary code, are shown in Fig. 1, along with the memory addresses to which they call. The underlined bits indicate the only bits that differ among the eight different Restart instructions.

While the Restart instructions are easy to generate and are the most common in 8080A systems, you can put any 8080A instructions (or Z80 instructions for the Z80) on the bus. For greater versatility in where the computer puts the Call Table service routine in memory, interrupt controllers (like the 8259A used in this month's project) are often designed to generate actual 3-byte Call instructions. This way, you could specify an actual 16-bit address in it. For instructions like this, where the number of instruction bytes is greater than one, the CPU will send enough interrupt-acknowledge signals to read in all the bytes necessary.

Mode 0 is primarily used in systems when you want to maintain compatibility with 8080A systems, or when you're using peripherals (such as the 8259A in this month's project) for use with 8080A-type processors. The Z80's flow response to an interrupt in mode zero is shown in Fig. 2.

Z80 Mode 1 Interrupts—Restart 38H

This is the simplest and least flexible interrupt mode the Z80 offers. If an interrupt occurs while you're enabling interrupts, the Z80 merely executes a Restart 38H instruction, which is a Call to memory location 0038 hexadecimal (hex). The Z80 produces no interrupt-acknowledge signal, and requires no external hardware. You use this mode in small systems, where the interrupt requirements are minimal and the external hardware required for the other Z80 interrupt modes is unnecessary. This is the one most often used by TRS-80s. The Model I uses it with its 25ms interrupt and disk controller interrupts, and the Models III and 4 don't even provide the interrupt-acknowledge signal on their expansion connectors, necessary for the use of the other interrupt modes.

Figure 3 illustrates the Z80's flow response to a mode 1 interrupt.

Z80 Mode 2 Interrupts—Indirect Call

Mode 2 is the most powerful of the Z80 interrupt modes. The Z80 generates an interrupt-acknowledge signal to the interrupting device, and expects a 1-byte vector in response. This vector is actually the low byte of a memory address which, by convention, must be even. The high byte of the address is found in the Z80's Interrupt (I) register, which you must set up with software. The Z80 puts the 2 bytes together to form a 16-bit address in memory, where it then gets the 16-bit address of the interrupt service routine. The I register points to a vector table page in memory, and the input vector byte provides the offset into that table. Since you can program the I register, you can load the vector table virtually anywhere in memory.

Most Z80 peripheral devices support mode 2 interrupts. You can also support it with a little external hardware, in much the same way as mode 0 interrupts. Mode 2 interrupts are used in most Z80 systems that use Z80 peripherals, or that provide the hardware required for the additional flexibility. The Z80's flow response to mode 2 interrupts is shown in Fig. 4.

The Z80 Nonmaskable Interrupt

You can't disable the nonmaskable interrupt. This allows top-priority response from the Z80 in case of a system emergency. When the Z80 responds to a nonmaskable interrupt, it executes a Restart 66H instruction, performing a call to memory location 0066 hex. The Z80 also saves the enable/disable status of the INT/ interrupt before it disables the interrupts for NMI/ servicing (described below). The Z80's flow response to a nonmaskable interrupt is shown in Fig. 5.

Interrupt Flip-Flops

The Z80 has two internal flip-flops for interrupt enabling and disabling. These flip-flops, designated IFF1 and IFF2, work with each other to provide the interrupt enable/disable flexibility, as well as temporary enable status storage during servicing of nonmaskable interrupts. On reset, the Z80 clears both IFF1 and IFF2, preventing maskable interrupts (INT/) from being accepted. When it executes an EI instruction, the Z80 sets both flip-flops, allowing it to accept maskable interrupts. The two flip-flops follow each other most of the time.

When a nonmaskable interrupt occurs, the Z80 saves the current state of

Restart Instruction	Bit Pattern	Call Address
RST 00H	11000111	0000H
RST 08H	11001111	0008H
RST 10H	11010111	0010H
RST 18H	11011111	0018H
RST 20H	11100111	0020H
RST 28H	11101111	0028H
RST 30H	11110111	0030H
RST 38H	11111111	0038H

Figure 1. Restart Instructions.

1. Interrupt occurs.
2. Disable further interrupts.
3. Assert acknowledge.
4. Read byte op-code on data bus.
5. More bytes to come? If yes, then 4.
6. Execute instruction received from interrupting device.

Figure 2. Mode 0 interrupt response.

1. Interrupt occurs.
2. Disable further interrupts.
3. Call memory location 0038H.

Figure 3. Mode 1 interrupt response.

1. Interrupt occurs.
2. Disable further interrupts.
3. Assert acknowledge.
4. Get vector on data bus.
5. Combine vector with I-register value to form address of service routine.
6. Get service routine address and call the service routine.

Figure 4. Mode 2 interrupt response.

1. Interrupt response.
2. Maskable interrupt status is saved.
3. Maskable interrupts are disabled.
4. Call location 0066H.
5. Restore maskable interrupt enable status when done.

Figure 5. NMI response.

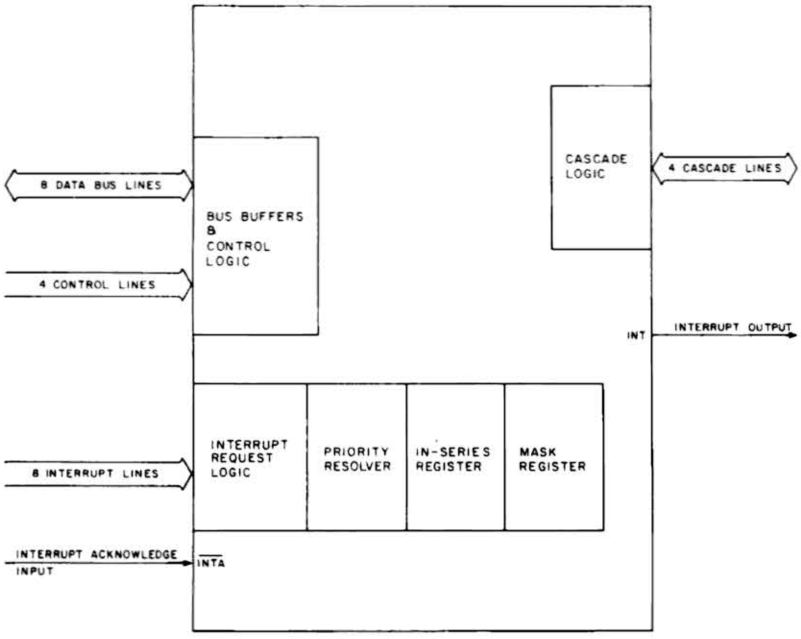


Figure 6. 8259A block diagram.

IFF1 in IFF2, while it clears IFF1 to make sure no maskable interrupts occur while processing the nonmaskable interrupt. When the Z80 executes the RETN (return from nonmaskable interrupt), it transfers the value in IFF2 to IFF1, restoring its original, pre-interrupt value.

Levels and Edges

Different processors have different ways of determining the validity of an incoming interrupt request. The Z80, for instance, looks for a low level on its INT/ line. When the line is low, the Z80 assumes that the connected device is requesting an interrupt. Some processors, like the 8085A, look for a high level. When the line is high, the processor assumes that the connected device is requesting an interrupt. Others look for an edge trigger. They expect to see a rising edge or falling edge of a signal for a valid interrupt request. Some even require a combination of an edge and a level.

Each of these triggering modes has its advantages and disadvantages. The level-triggered mode is the most common, and assumes that whenever a device has its interrupt line at a certain level (either high or low, depending on the processor), it needs servicing. During the service routine, the processor performs whatever operations are necessary for the device to deactivate its interrupt request, before the processor reenables the interrupts.

The edge-triggered mode assumes that a device will activate its interrupt line when it needs service, making the necessary edge trigger, but may not deactivate the line level until some later time. In this case, it is not desirable for the CPU to keep seeing the line as an active interrupt and attempting to service it.

For example, say an interrupt input is positive edge-triggered, meaning that it is triggered by a low-to-high voltage transition. The transition notifies the CPU that a device is requesting an interrupt so the CPU can service the interrupt. At the end of the interrupt service routine (ISR), the CPU reenables interrupts, but the interrupt line is still high. A CPU with a level-triggered input would immediately see the high line as another interrupt request and service the interrupt. The edge-triggered CPU, however, would not recognize another interrupt request until the line returned low, then went high again.

The third type of triggering, a combination of edge- and level-triggering, is often used for nonmaskable interrupt inputs, or in systems with noisy environments. The CPU only checks the interrupt lines between instructions (actually at the end of each instruction). If a low-to-high transition occurs on the line, the CPU sets an internal flip-flop to record the edge. Then, when the time comes to check for an interrupt, the CPU looks for both an active flip-flop, indicating the edge has occurred, and an active (high) input. If both of these conditions are not true, no interrupt service processing will take place.

The advantage of looking for both conditions is two-fold. First, if a glitch (voltage spike) occurs and sets the flip-flop, it will not cause a nonmaskable interrupt, since the line probably won't still be high when the CPU checks it. Second, if a nonmaskable interrupt does occur, the CPU won't recognize another nonmaskable interrupt until the line returns low, then goes high again. The benefits of level-

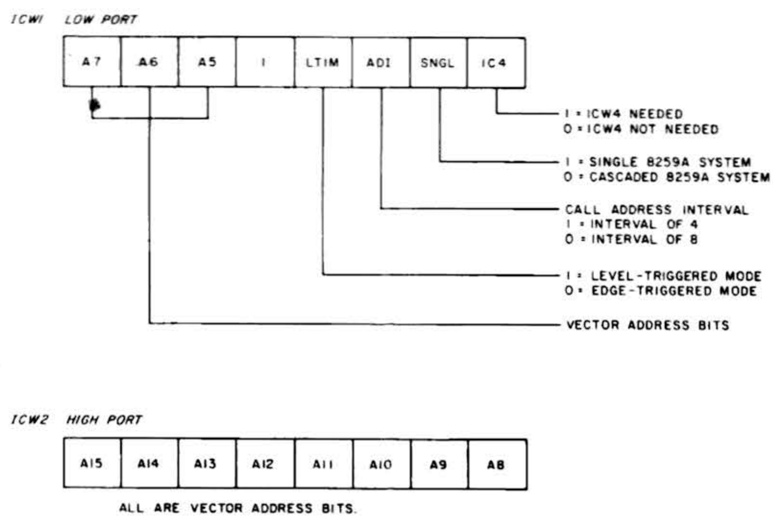


Figure 7. ICW1 and ICW2 of 8259A.

TAX-PREP™ '85

Easy, 1984 tax templates for MULTIPLAN™ or LOTUS 1-2-3™ users

"The best I've seen, excellent support, good instructions, very easy to get used to." Professional Tax Preparer, NJ.

SAVE TIME - SAVE MONEY
USE YOUR MULTIPLAN or LOTUS 1-2-3 SPREADSHEET

Professional, fully documented, linked templates for preparing 1984 personal income tax returns. Enter data once. User friendly, complete manual & tutorial. All tax tables and computations built in, also use for "what if" tax planning. IRS accepted printout.

Includes all of the following forms and schedules: 1040, A, B, C, D, E, F, G, R, SE, W, 2106, 2119, 2210, 2441, 3468, 3903, 4562, 4684, 4797, 5695, 6251.

TO ORDER: specify TRS-80 Models 4, 4p, II, 12, or 16 (TRSDOS), 8" CP/M or Tandy 2000. IBM PC, Apple II and Macintosh also available. MULTIPLAN or 1-2-3 REQUIRED (specify which).

TAX-PREP '85 - $129.95 plus $3.00 shipping (PA add 6% tax). Call or write for more information.

EZ WARE 178

17 BRYN MAWR AVE. S8, Bala Cynwyd, Pa 19004 (215) 667-4064

Electronic Mail.... On A Micro!

As well as Public, Private or Pay BBS.

Our system offers features such as:

- Multi levels of access to a user.
- High security of system and your files.
- Unattended message transfer to compatible system.
- Fully maintainable from a remote location.
- User configurable to include or exclude most features, allowing you to personalize your system.
- Number of users & messages configurable from 1 to 900, allowing those with large storage to use it.
- Configurable to handle different terminals.
- Supports Hayes type of modem.
- and much much more.

ORDER NOW - $AVE $10.00 when you mention this AD.

IBM & Compatibles $249.95
TRS-80 .... $199.95
Mail Label Module $59.95
Terminal for IBM $79.95

32 *TVG Systems* P.O. BOX 94433
(604) 946-7910 Richmond, B.C. V6Y 2A8

Try Our BBS - Call our Data Line - 604-946-0955

MULTI-BASIC

The Compatible BASIC Compiler from Alcor
(Supported Features Chart)

	MBASIC	MBASIC Compiler	CBASIC	CBASIC Compiler	Multi-Basic
IF ELSE/FOR NEXT	•	•	•	•	•
WHILE WEND	•	•	•	•	•
OPEN/CLOSE	•	•	•	•	•
CREATE/DELETE			•	•	•
PRINT/PRINT USING	•	•	•	•	•
LPRINT/LPRINT USING	•	•			•
PRINT @	•	•			•
INPUT/LINE INPUT	•	•	•	•	•
INPUT #/LINE INPUT #	•	•			•
READ #/READ # LINE			•	•	•
LOC/LOF/EOF/ERROR	•				•
FIELD/GET/PUT	•	•			•
RSET/LSET/ERASE	•	•			•
MKD$/MKI$/MKS$	•	•			•
CVI/CVS/CVD/SPC	•	•			•
READ/DATA/RESTORE	•	•	•	•	•
TRON/TROFF	•				•
DEF FN/RANDOM/RND	•	•	•	•	•
DEF USR/SWAP/WAIT	•	•			•
CALL (ASSEMBLY LANG)	•				•
DEFSTR/DBL/SNG/INT	•	•			•
DOUBLE/REAL			•	•	•
INTEGER/STRING			•	•	•
GOTO/GOSUB	•	•	•	•	•
ON ERROR GOTO	•			•	•
RESUME/RESUME NEXT	•				•
ERL/ERR	•			•	•
ON number GOTO/GOSUB	•	•	•	•	•
NAME/RENAME	•	•	•	•	•
PEEK/POKE/INP/OUT	•	•	•	•	•
SYSTEM/SOUND	•				•
SADD/MATCH/UCASE$			•	•	•
VAL/TAB/STR$/VARPTR	•	•	•	•	•
SIN/COS/TAN/ATN	•	•	•	•	•
LOG/EXP/ABS/SQR	•	•	•	•	•
COMMAND$/IF END			•	•	•
INKEY$/INPUT$	•	•			•
TIME$/DATE$/HEX$/OCT$	•	•			•
STRING$/SPACE$	•	•			•
LEFT$/RIGHT$/MID$	•	•	•	•	•
CHR$/ASC/LEN/SGN	•	•	•	•	•
OPTION BASE	•				•
ROW/POS/LPOS	•				•
DIM/MEM/FRE	•	•	•	•	•
MOD/MFRE				•	•
GET/PUT (CHARACTER)			•	•	•
KILL/CLS/CLEAR/INSTR	•	•			•
INT/FIX/CINT	•	•			•
CSNG/CDBL	•	•			•
CHAIN	•		•	•	•
CONSTAT%/CONCHAR%			•	•	•
CONSOLE/LPRINTER			•	•	•
BINARY RANDOM FILES	•	•			•
ASCII RANDOM FILES			•	•	•
255 CHARACTER NAMES				•	•
REDIMENSIONED ARRAYS	•		•	•	•
UNLIMITED STRING SIZE				•	•
MULTI-LINE FUNCTIONS			•	•	•
PROCEDURES/RECURSION					•
FUNCTION TRACING					•
LOCAL VARIABLES			•	•	•
NESTED FUNCTIONS					•
OPTIONAL LINE NUMBERS			•	•	•
DESCRIPTIVE LABELS			•	•	•
SINGLE PRECISION	•	•			•
DOUBLE PRECISION	•	•	•	•	•
LINK TO PASCAL & C					•

For TRS80 models I, II, III, 4, 12, or 2000 using TRSDOS, CP/M, or MSDOS and for IBM PC using PCDOS
Multi-Basic, Pascal, or C $139
Add shipping $6.00 USA, $28 Overseas

ALCOR Systems
13534 Preston Road, Suite 365
Dallas, Texas 75240
(214) 494-1316

215

See List of Advertisers on Page 145

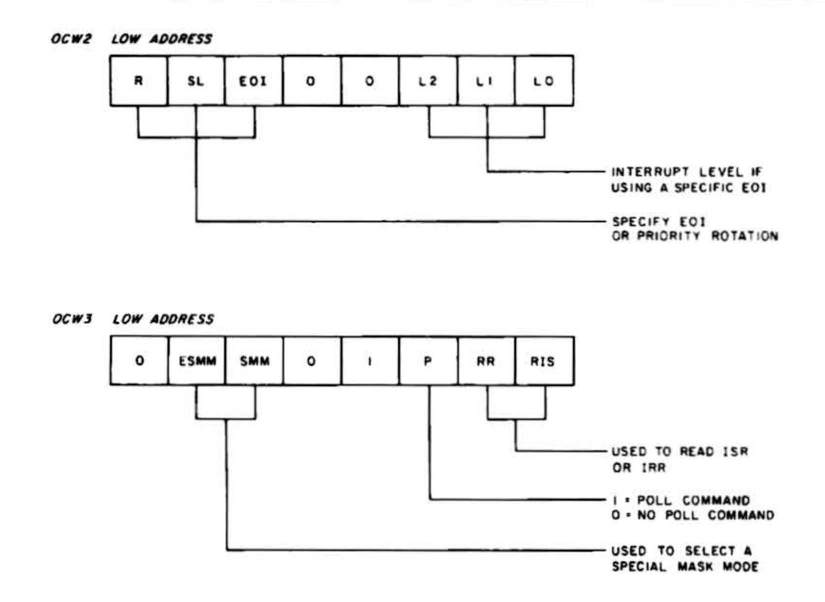


Figure 8. OCW1 and OCW3 of 8259A.

triggering and edge-triggering are combined for a double-safe interrupt input.

The 8259A Programmable Interrupt Controller

The 8259A programmable interrupt controller (PIC), developed by Intel, is a versatile device that provides eight interrupt inputs and a full 3-byte call to the CPU for versatility in locating the interrupt jump table in memory. The basic block diagram of the 8259A is shown in Fig. 6. An 8-bit bus interface handles communication with the CPU and one interrupt output to the CPU. The 8259A takes up two locations in I/O addressing space.

The 8259A provides several features that make it useful for a variety of applications. The eight interrupt inputs are prioritized, allowing higher-priority devices to interrupt lower-priority devices, but not the other way

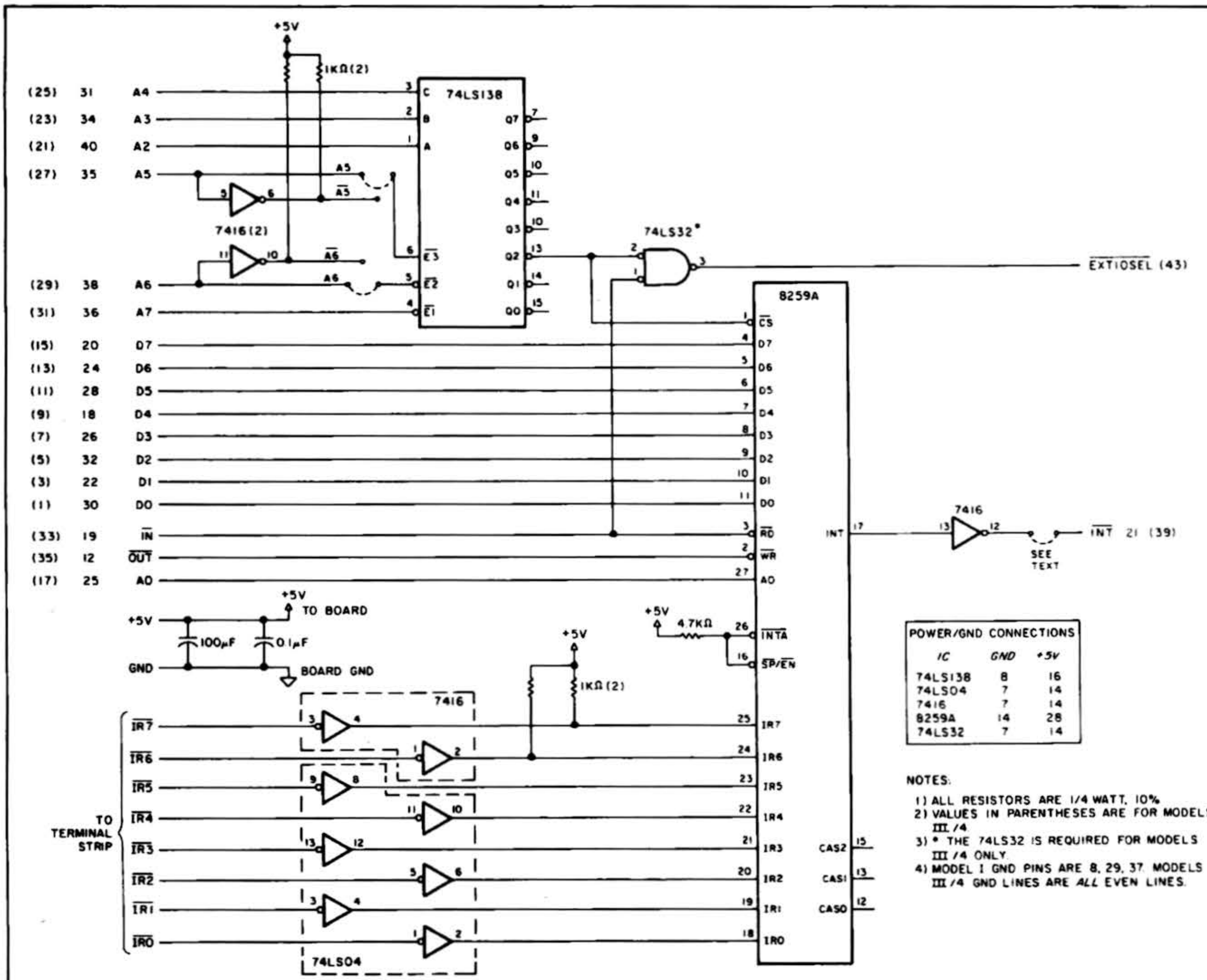


Figure 9. Interrupt controller board.

PRINTER SALE

$189

ML 80 Features

- 80 cps unidirectional printing
- 80 columns standard, 132 condensed
- TRS-80 character set
- Graphics 64 block shapes
- Long-life, 7- pin print head, warranted for one full year
- Friction & pin paper feeds
- Parallel Centronics interface

OKIDATA

SUNLOCK SYSTEMS ✓456
4217 Carolina Ave.
Richmond, Va. 23222

ADDITIONAL PRINTER SPECIALS

Epson		Okidata		Brother 15	$479	CItoh 8510	$359
RX80	$299	ML82	$349	Brother 25	739	CItoh F10	1099
80FT	369	ML83	569	Gemini 10X	279	Sv.Reed 500	429
RX100	529	ML84	975	Man/Tal 80	319	Diablo 620	949
FX80	479	ML92	429	Qume 1140	1349	NEC 2000	849
FX100	649	ML93	699	SCM TP-2+	449	NEC 3510	1499

TO ORDER CALL TOLL FREE 800-368-9191
In Virginia call 804-321-9191
We accept MasterCard, Visa and CODs

DISK TERM

VERY SMART TERMINAL

Terminal Mode: Transmit an ASCII file from memory or disk. Programmable for unattended execution. Receive a file, write to disk when buffer is full, clear buffer and repeat, all unattended. File can even span disks.

Transmit and Receive Protocol: Separate from terminal. Allows transmission of multiple files of any type. Completely programmable for unattended operation at both ends. Even redials if no carrier.

Other Features:

- Menu Driven
- DOS commands from program
- Supports auto-dial modems
- XON & XOFF changeable
- Auto log-on
- Translation Tables
- Printer buffer 25K to 2.5K
- Programmable Macro-keys
- Operates under DOSPLUS, LDOS, NEWDOS/80, TRSDOS
- Prepare files off-line then upload
- Telephone Directory for auto-dial with modem settings
- Store all settings in configuration files
- Model 3 & 4 transmit and receive at up to 9600 baud

Model 4 enhancements: Full 80 x 24 screen. Has split screen option to separate data received and transmitted. Able to use second 64K for buffer.

Model I, III, & 4 .... $69.00
Add $3.00 Postage / Check, Money Order, Visa, & Mastercard
Indiana Residents add 5% Sales Tax

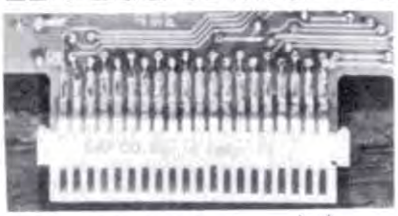
INDIANA SOFTWARE DEVELOPMENT CO.
(Division of DonTho Computer Service, Inc.)
723 Franklin Square, Suite 502 • Michigan City, IN 46360
Phone: (219) 879-2941 ✓425

Gold Plug 80

TRS-80+ MOD I, III, COCO, TIMEX 1000, OSBORNE, TI99/4a others

Eliminate disk reboots and data loss due to oxidized contacts at the card edge connectors.
GOLD PLUG 80 solders to the board edge connector. Use your existing cables. (if gold plated)

GOLD PLUG 80 Mod I Mod III



new — SPECIAL PRICE

GOLD PLUG 80 Mod I (6)	$44.95
Keyboard/El (mod I)	15.95
Individual connectors	$7.95
GOLD PLUG 80 Mod III (6)	48.95
Internal 2 Drive Cable	29.95
Mod III Expansion port	9.95

Available at your favorite dealer or order direct from **E.A.P. CO.**

Dealer Inquiries Invited

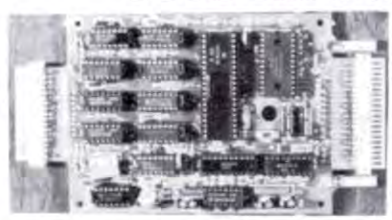
USA shipping $1.45 — Can/Mex $4
Foreign $7 — TEXAS 5% TAX

VISA

+ trademark Tandy Corp

COCO MODULE INSTALLATION AVAILABLE

COCO Disk Module



Ground tab extensions

new — SPECIAL PRICE

COCO Disk Module (2)	$16.95
Ground tab extensions	INCL
Disk Drives (all R.S.)	$7.95
Gold Disk Cable 2 Drive	29.95
Four Drive Cable	39.95

Model I, III/4

PERCOM HARD DISK
5 Mb Hard Disk .... $ 995.00
10 Mb Hard Disk $1450.00
SYNCOM DISKETTES $17.95/10
(SINGLE SIDED DOUBLE DENSITY)

GOLD PLUG 80-E.A.P. COMPANY ✓216
P.O. Box 14 Keller, TX 76248 (817) 498-4242

MasterCard

✓ See List of Advertisers on Page 145

around. You can program the inputs as level-triggered or edge-triggered, and you can selectively mask (enable or disable) them. You can cascade up to nine 8259A's together, providing up to 64 prioritized interrupt levels.

The CPU can send two kinds of command words to the 8259A: Initialization command words (ICWs) and operation command words (OCWs). The CPU sends the ICWs only once to initialize the 8259A after power-up. It sends the OCWs at various times as required to perform different operations. OCWs can alter interrupt masks, change priority levels, acknowledge interrupts, and more.

While you have four defined ICWs available, the interrupt controller board will use only two, described here. You'll use all three of the defined OCWs, since they provide the real flexibility in operation.

The interrupt controller is primarily designed to generate a 3-byte call instruction to the CPU when a device requests an interrupt, and the CPU responds with an interrupt acknowledge. Since TRS-80s aren't well-designed for this type of interrupt response (particularly the Models III and 4) you won't use that feature; instead, you'll use its polled mode operation capability.

When you apply power to the 8259A, the state of the device, including its interrupt output line, is unknown, since no reset input is available. Before you use the device, you must initialize it with the ICWs. The two ICWs you'll use are shown in Fig. 7. While two are shown, you'll use only the first, although you must still send both to the 8259A to meet initialization requirements.

The default mode setting for the 8259A is for 8080A mode, which is what you want. You can also program it to work with the 8086/8088 family of processors. Bit 0 of ICW1 specifies whether or not ICW4 will or won't be sent. Since ICW4 specifies optional parameters you won't use, this bit is programmed with a zero. Bit 1 distinguishes whether the chip is by itself or if other PICs are cascaded with it. This bit is programmed as a 1 in our system, since it's by itself. Bit 2 relates to the spacing of service routine call addresses in normal operation, and doesn't apply to our system.

Bit 3 of ICW1 determines whether the inputs will look for level-triggered interrupts (1) or edge-triggered interrupts (0). I will use level-triggered interrupts in my examples, but use whatever meets your needs. Bit 4 must be a 1 to indicate that you're sending ICW1. Bits 5–7 of ICW1 and all of ICW2 provide address bits for the call addresses generated during normal operation. These don't apply to your system either.

Once you initialize the 8259A, you can selectively enable or disable the various interrupt inputs by writing a mask byte (OCW1) to the higher 8259A port (port 29 hex on your board). Any bit that is set masks (disables) the corresponding interrupt input, and cleared bits enable the corresponding inputs.

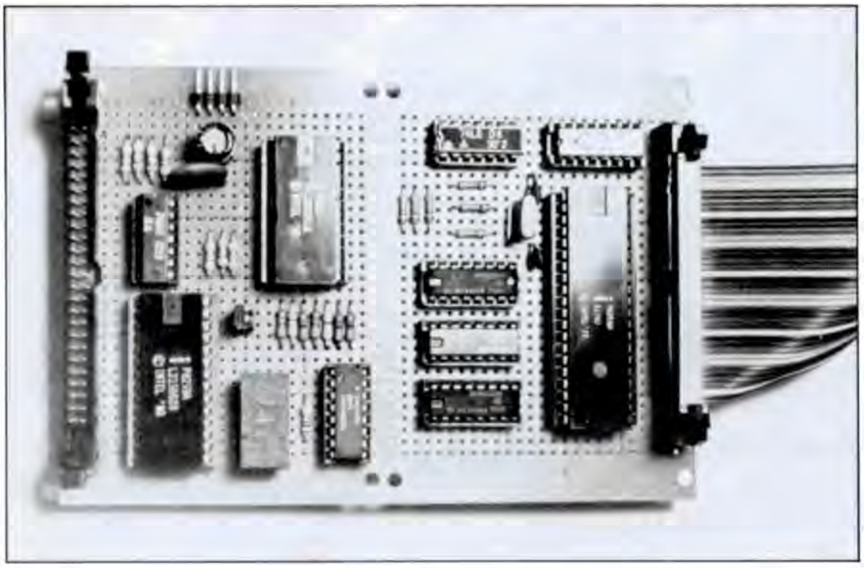


Photo 2. General-purpose I/O board with interrupt controller.

The formats for OCW2 and OCW3 are shown in Fig. 8. I'll describe the functions of these operation command words later, when I discuss the operation of the interrupt controller board.

Building the Interrupt Controller Board

The parts list for the interrupt controller board (as well as the general-purpose I/O board addition) is shown in Table 1, and the schematic is shown in Fig. 9. Some of the parts are optional, depending on what you actually want to build (described in the remainder of this article). The interrupt controller board is shown in Photo 1. The photo does not include the recommended additional reset circuitry described below.

Building the controller board is simple and straightforward. You need only a few easily obtained ICs, along with a handful of passive components. The addressing for the 8259A is jumper selectable, near the 74LS138. The jumper settings I used are indicated by dotted lines on the schematic (the programs described later will assume this addressing). The possible addressing for the device is as follows:

E2	E3	Addressing Range
A6/	A5/	48–4B hex
A6/	A5	68–6B hex
A6	A5/	08–0B hex
A6	A5	28–2B hex (my choice)

Although the hardware reserves four port locations for each jumper setting, the 8259A requires only two. It is double addressed within the port addressing range. That is, the device will appear in the lower two or upper two locations of the chosen addressing range. The convention for such a situation is to use the lower addresses of the addressing range. I will use addresses 28 hex and 29 hex in the programs described later.

Because the 8259A doesn't have internal reset circuitry, the state of its interrupt output line (pin 17) is unknown at power-up. If this is connected (via the 7416 gate) to the TRS-80 interrupt input line at system power-up, a problem will exist if the interrupt is active. This may not be a problem with the Models III and 4 since they have an internal register to enable external interrupts. I presume (though I am not certain) the initialization software in the TRS-80 disables external interrupts, making it necessary for applications software to enable external interrupts before the CPU accepts interrupts. If this is true, the optional reset circuitry described below is unnecessary. In any case, the optional circuitry can't hurt, and I recommend it at least for all Model I users.

The reason the jumper is shown on the schematic (on the interrupt line going to the TRS-80) is in case you don't want to add the optional reset circuitry, or for some reason don't want the interrupt on the TRS-80's interrupt line. You may not need the jumper in your application. You'll also need a +5V @150mA (milliamps).

The interrupt inputs to the 8259A have inverters on them. The inputs to these inverters can then be connected to a terminal strip to provide you with eight active low interrupt inputs.

Building the General-Purpose I/O Board Addition

In December's Project 80, I described the construction of a general-purpose I/O board. An interrupt controller would be a valuable addition, so I've provided the schematic for it (see Fig. 10). The parts list is in Table 1, as it is for the stand-alone interrupt controller board described above. The general-purpose I/O board with the interrupt circuitry is shown in Photo 2.

The construction of this addition is straightforward. You address (chip select) the 8259A from an unused pin on the 74LS138 (pin 13). This gives you the same addressing range as for the stand-alone controller board described above, 28–2B hex. The general-purpose I/O board already has two major peripheral devices, and they're designed to provide interrupts to the CPU during various modes of operation. The five possible interrupt pins (two from the 8255A and three from the 8253/4) are connected to the first five interrupt inputs on the 8259A. This allows selective interrupt inputs under software control.

You could use this, for example, to generate an interrupt to the CPU at predetermined points in time, and update something during the interrupt service routine. I'll illustrate this in the programs described later. This also allows the 8255A to generate interrupts to the Z80 after certain handshaking operations. (I used the same concept in the printer buffer project described in the September/October 1984 Project 80, p. 102 and p. 146.)

I used five inputs to the 8259A, with three unused. I put inverters at these inputs, to make available three general-purpose active low interrupts.

The Optional Reset Circuitry

The schematic for the optional reset circuitry is shown in Fig. 11. The parts are listed in Table 1, along with the other project parts. The reset circuitry provides a function available on many peripheral devices, and would probably have been put on the 8259A if an extra pin were available.

The reset circuitry is very simple, and requires the addition of only a 74LS74 dual D-type flip-flop and a 74LS32 quad OR-gate (Models III and 4 users may use spare 74LS32 gates from the EXTIOSEL/ circuitry), and a couple of resistors. The interrupt output that was to go to the TRS-80 INT/ line (pin 12 of the 7416) is routed instead to pin 9 of the 74LS32. The TRS-80 RESET/ output line then goes to pin 1 of the 74LS74, the reset line on the first flip-flop. For this to work, you must turn on the interrupt controller before or simultaneous with the TRS-80, so that it can see the TRS-80's power-up RESET/ signal.

Pins 13 and 12 of the 74LS32 connect to the chip select (CS/) and write (WR/) lines on the 8259A, respectively. This creates an active (low) output signal when both input signals are active (low). This creates the necessary clock input to the 74LS74 when the 8259A is written to.

Look at how this circuit works. When you reset the TRS-80, the Q/ (Q not) output of the 74LS74 (pin 6) goes high. This makes the output (pin 8) of the connected OR-gate high, regardless of the other gate input. The second 74LS74 flip-flop is then used as an inverter, to create a low level on its Q (pin 9) output. This then goes to the 7416 gate to get re-inverted to a high level, an inactive interrupt level.

When the computer first writes to the 8259A, however, the clock input (pin 3) to the first flip-flop makes the D-input value (high) appear on the Q output, and the inversion of that to appear on the Q/ output. The low level on Q/ enables the connected 74LS32,

See List of Advertisers on Page 145

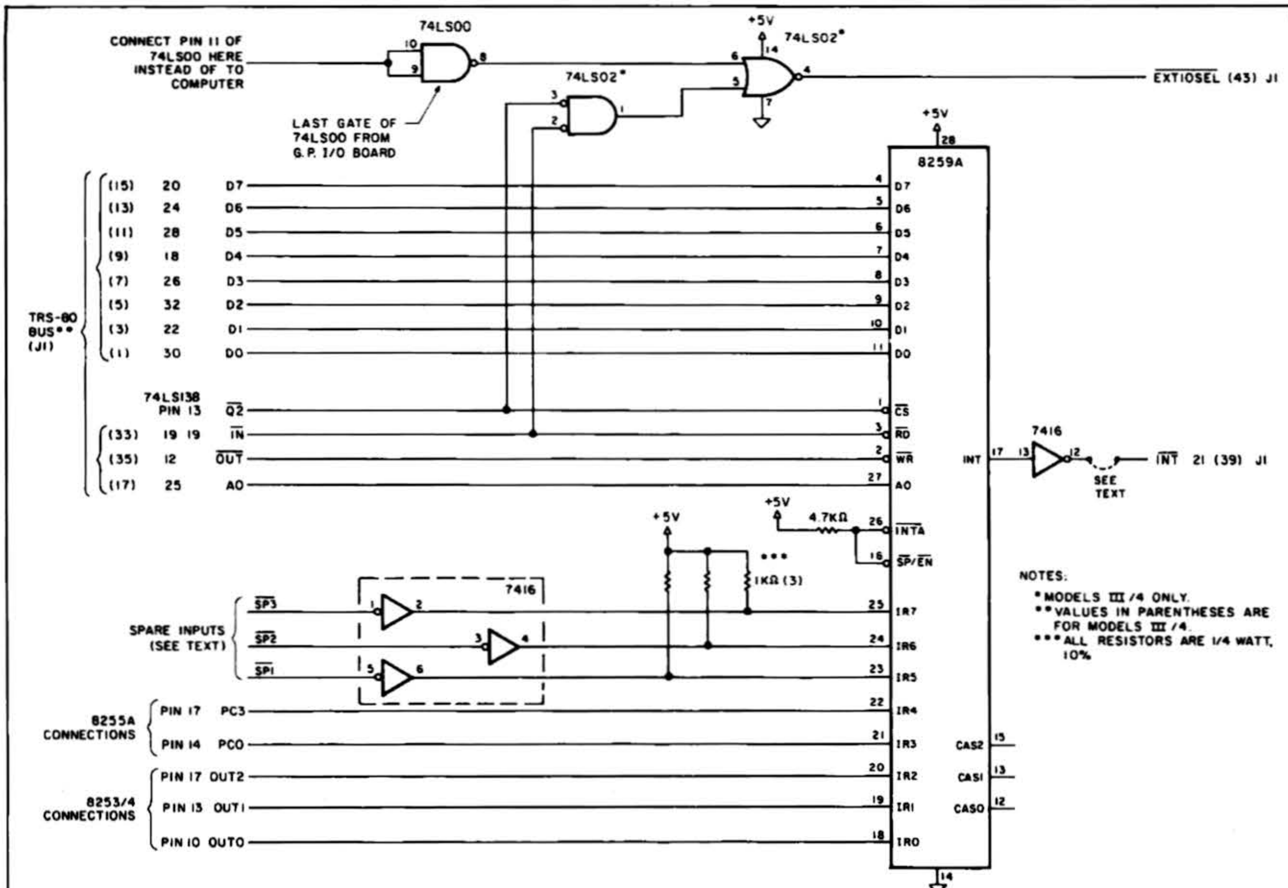


Figure 10. Interrupt controller addition to the I/O board.

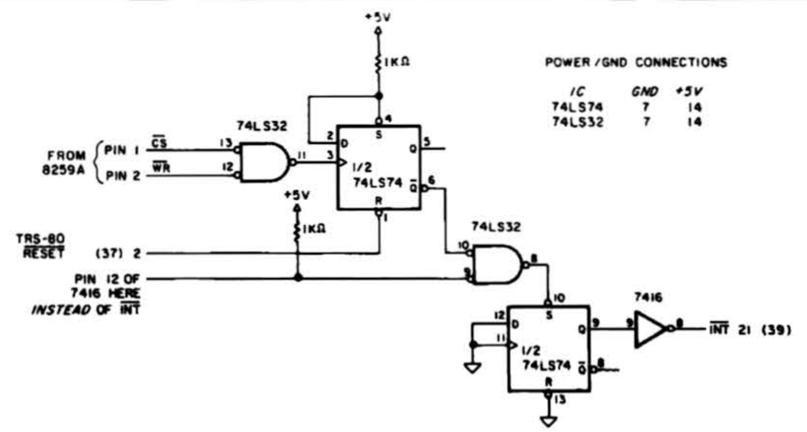


Figure 11. Optional reset circuitry for interrupt controller.

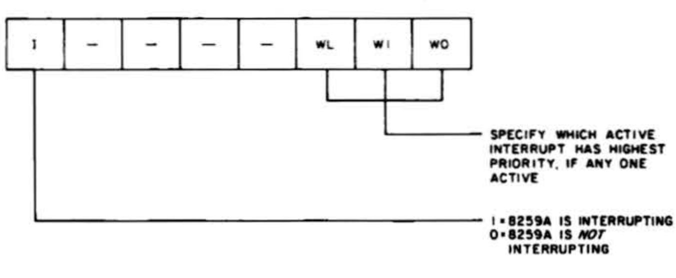


Figure 12. Interrupt status byte from 8259A.

which transfers the interrupt level on its pin 9 input to its pin 8 output, and, after two further inversions, is finally placed on the TRS-80's interrupt line. The interrupt output from the 8259A to the TRS-80's interrupt line will, then, always be enabled until it receives another reset signal. The reason for the double inversion (using the second flip-flop and the 7416) is because of the need to use the open collector 7416 gate, described below.

Operating the Interrupt Controller

This section concerns both the stand-alone controller board, as well as the general-purpose I/O board addition, since both operate identically. One of the essential requirements of the interface to the Model I bus is to use an open collector driver. Open collector drivers, like the 7416 inverting drivers, don't have the collectors on their output transistors pulled up, as do standard transistor-to-transistor logic ICs. This lets you tie any open-collector outputs together with a common pull-up resistor. The net effect is that, when all the outputs are high, the line is high. When any one of them goes

✓ See List of Advertisers on Page 145

Program Listing. Demonstration for Model III/4 to illustrate the use of the 8259A interrupt controller with the December Project 80 general-purpose I/O board.

```
             ;*******************************************************
             ; This is a demonstration program for the Model III/4 to*
             ; illustrate the use of the 8259A interrupt controller  *
             ; with the Project 80 G.P. I/O board. It sets up the    *
             ; various board devices to increment the LED display    *
             ; every second. See column text for a more-indepth      *
             ; description.                                          *
             ;                                                       *
             ; Written by Roger C. Alford                            *
             ;*******************************************************
             ;
             ;
0023         PPICMD  EQU     23H     ;8255A COMMAND PORT
0021         PPIB    EQU     21H     ;8255A PORT B
0028         PICP0   EQU     28H     ;LOWER PORT OF 8259A
0029         PICP1   EQU     29H     ;UPPER PORT OF 8259A
0027         PITCMD  EQU     27H     ;8253 COMMAND PORT
0024         PITT0   EQU     24H     ;TIMER 0 PORT ON 8253
00EC         TRSEXE  EQU     0ECH    ;TRS-80 EXTERNAL I/O ACCESS PORT
00E0         TRSINE  EQU     0E0H    ;TRS-80 INT. ENABLE PORT
00E0         TRSRDI  EQU     0E0H    ;TRS-80 READ INT. STATUS PORT
             ;
7000                 ORG     7000H   ;SAFE ADDR FOR ALL MEM SIZES
             ;
             ;VARIABLES USED:
7000 9770    INTTBL  DEFW    TIMER0  ;TIMER 0 INTERRUPT
7002 B870            DEFW    BADINT  ;INVALID INTERRUPT
7004 B870            DEFW    BADINT  ;INVALID INTERRUPT
7006 B870            DEFW    BADINT  ;INVALID INTERRUPT
7008 B870            DEFW    BADINT  ;INVALID INTERRUPT
700A B870            DEFW    BADINT  ;INVALID INTERRUPT
700C B870            DEFW    BADINT  ;INVALID INTERRUPT
700E B870            DEFW    BADINT  ;INVALID INTERRUPT
             ;
0001         PPIBYT  DEFS    1       ;STORAGE FOR 8255 PORT B BYTE
0001         TIMCNT  DEFS    1       ;STORAGE FOR INTERRUPT COUNTER
0002         VECTOR  DEFS    2       ;STORAGE FOR TRS-80 ISR ADDRESS
             ;
             ;
7014 F3      INTRPT  DI                          ;DISABLE Z80 INTERRUPTS
                                                 ; FOR INITIALIZATION.
7015 3E10            LD      A.10H               ;WRITE BYTE TO ENABLE
7017 D3EC            OUT     (TRSEXE),A          ; EXTERNAL I/O ACCESS
7019 3E1E            LD      A,1EH               ;GET 8259A ICW1 -- SINGLE
                                                 ; CONTROLLER SYSTEM,
                                                 ; LEVEL-TRIGGERED.
701B D328            OUT     (PICP0),A           ;SEND TO 8259A
701D AF              XOR     A                   ;ZERO THE ACCUMULATOR
701E D329            OUT     (PICP1),A           ;WRITE ICW2 -- NOT REALLY
                                                 ; USED HERE.
7020 3EFE            LD      A.0FEH              ;GET INT. MASK BYTE
7022 D329            OUT     (PICP1),A           ;ENABLE TIMER 0 INT.
7024 3E99            LD      A.99H               ;SET 8255A PORT B FOR
                                                 ; OUTPUT MODE 0.
7026 D323            OUT     (PPICMD),A          ;SEND TO 8255A COMAND PRT
7028 3E03            LD      A.03H               ;SET LED TO '0' AND
                                                 ; TIMER 0 FREQ.=250 KHZ.
702A D321            OUT     (PPIB),A            ;SEND TO 8255A PORT B
702C 321070          LD      (PPIBYT),A          ;SAVE VALUE ALSO IN MEM.
702F 3E30            LD      A,30H               ;GET 8253 COMMAND BYTE --
                                                 ; SET TIMER 0 TO MODE 0
7031 D327            OUT     (PITCMD),A          ;SEND TO 8253 COMAND PRT
7033 3E24            LD      A,24H               ;GET LOW COUNT BYTE
7035 D324            OUT     (PITT0),A           ;SEND TO TIMER 0
7037 3EF4            LD      A.0F4H              ;GET HIGH COUNT BYTE
7039 D324            OUT     (PITT0),A           ;SEND TO TIMER 0
             ;
703B AF              XOR     A                   ;CLEAR THE ACCUMULATOR
703C 321170          LD      (TIMCNT),A          ;CLEAR THE INT. COUNTER
703F 3A1240          LD      A.(4012H)           ;SEE IF ISR ALREADY THERE
7042 FEC9            CP      0C9H                ;LOOK FOR A RETURN
7044 2808            JR      Z,NOISR             ;IF YES, NO ISR IN SYS.
7046 2A1340          LD      HL,(4013H)          ;ELSE, GET ISR ADDRESS
7049 221270          LD      (VECTOR),HL         ;SAVE IN VECTOR LOCATION
704C 1806            JR      INITDN              ;FINISH INIT.
704E 21C470  NOISR   LD      HL,NOPROC           ;GET ADDR FOR NO PROCESS
7051 221270          LD      (VECTOR).HL         ;PUT IT INTO VECTOR
7054 3EC3    INITDN  LD      A.0C3H              ;GET JUMP OPCODE BYTE
7056 321240          LD      (4012H),A           ;WRITE TO TRS-80 VECTOR
                                                 ; TO INSURE IT'S THERE
7059 216670          LD      HL,SERVIC           ;GET ADDRESS OF OUR ISR
705C 221340          LD      (4013H).HL          ;SAVE IN TRS-80 VECTOR
705F 3E08            LD      A,08H               ;WRITE BYTE TO ENABLE
7061 D3E0            OUT     (TRSINE),A          ; EXTERNAL I/O INTS.
7063 FB              EI                          ;ENABLE INTERRUPTS NOW
7064 18FE    LOOP    JR      LOOP                ;LOOP FOREVER -- JUST
                                                 ; WAIT FOR THINGS TO
                                                 ; HAPPEN.
             ;
             ;*******************************************************
             ; This is the interrupt service routine (ISR)           *
             ; processing code. If the interrupt belongs to the      *
             ; 8259A, it will be processed here. Otherwise, it will  *
             ; be forwarded to the TRS-80.                           *
             ;*******************************************************
             ;
7066 F5      SERVIC  PUSH    AF                  ;SAVE USED REGISTERS ON
```

Listing continued on p. 162

low, the line goes low. This is called a wired-OR configuration.

Open collector drivers allow many devices to share the common interrupt line to the CPU. The CPU must, however, have some way to determine what is requesting any given interrupt. While Model I's require the use of open collector drivers, Models III and 4 users could get by with using a 74LS04 instead of the 7416 chip described above, since open-collector drivers aren't required on the external I/O bus. It is, however, recommended that you use the 7416 because it's a better driver and it lets you parallel other devices that use the interrupt line on the bus in the future. The pull-up for the 7416 is internal to the Models III and 4, as well as to the Model I.

As mentioned during the discussion of the 8259A, the operation command words let you set up most of the various options the 8259A allows. I've developed a program for the Models III and 4 that illustrates the use of the 8259A (see the Program Listing). The program uses the functions of the general-purpose I/O board, but should give you an understanding of the operation of the interrupt controller even if you only build the stand-alone interrupt controller board.

You first set up the 8259A by sending the two ICWs, as well as the interrupt mask (the port addressing is important to note). The interrupt mask sent to the 8259A is 0FEH, only enabling interrupt input 0, which is the output of Timer 0 on the 8253. The 8255A is then set up to permit control of the programmable divided clock to the 8253, as well as the LED display. The display is initialized with a 0. The original 8255A value is also stored in a memory location (PPIBYT) for later reference (during the ISR). Another memory location (TIMCNT) is also cleared, to be used as a count register.

Timer 0 of the 8253 is now set up to operate in mode 0 (interrupt on terminal count), which causes the output of the timer to initially go low, but then go high when the count reaches zero. The frequency input to Timer 0, as established by the 8255A, is 250 KHz, the slowest possible frequency. Assuming we would like to update the LED display once each second, the interrupt to the CPU must occur at a frequency that will easily multiply to

Continued on page 162

Getting the Message Across

Illustration by Roy Lewando

One of the most important functions of a BBS is its ability to search and find information. Successful BBSes can locate information accurately and efficiently. This month's BBS Express concentrates on finding messages both in memory and on disk.

While it's faster to locate information in memory than on disk, memory space is more limited, which in turn limits the amount of information you can keep in memory. A message header, for example, consists of the message number, subject, date, time, section number, name of sender, and destination. Ideally, all this information should be in memory so you can locate a message quickly. This information requires a total of 92 bytes for each header, or 9.2K of memory if there are 100 messages on board. Unfortunately, however, that's a lot of memory.

To complicate matters further, this header information is in string form, which can easily trigger string garbage collection. The message number, however, isn't in string form. An acceptable solution, then, is to create a fast, in-memory search for the message number, leaving all other searches as disk operations. Because the message numbers are stored economically, memory isn't heavily taxed.

Storing the Message Number

Your computer's memory stores integers in Z80 least significant byte/most significant byte (LSB/MSB) format. Each integer requires 2 bytes: The first byte contains the least significant byte, while the second byte contains the most significant byte. In message number 256, for example, the least significant byte is 0, and the most significant byte is 1. The VARPTR command returns the address where the BBS stores the least significant byte; it stores the most significant byte in the next position.

Using this information, you build an in-memory index of the message numbers that you access with a short machine-language code (see Program Listing 1). The index is simply a string that contains the message numbers in sequential order, making it easy to search for any given message number. The VARPTR (string variable) returns the memory address of a pointer that contains both the length of the string and the memory address of the string's first character. Because this information is all part of the same string, it's easily written to disk.

Now you need to convert the message numbers to and from the Z80 LSB/MSB format. Disk Basic does this: It contains MKI$, which returns a 2-byte string in LSB/MSB format, and CVI, which takes that string and returns a number. In the case of message number 256, MKI$(256) = CHR$(0) + CHR$(1); CVI(CHR$ (0) + CHR$ (1)) = 256. Each time a user leaves a message, the BBS writes MKI$ (message number) to the in-memory string. The BBS stores the header for the Nth message in record N, and finds it in the index string at position 2*N − 1. If P is the position in the string, and RN is the record number of the header for the message, then

$$RN = (P + 1)/2$$
$$P = (2*RN) - 1$$

Program Listing 1, in Assembly language, searches the index for the message number. With this message number, you have access to the header, which gives you the secret file name of the text of the message. Label FSRCH calls the PARAM routine which places the length of the string passed from Basic in the B register, and the address of the string's first character in the HL register pair. Basic POKEs the message number that you're looking for in SEARCH + 1.

The Key Box

LOAD 80

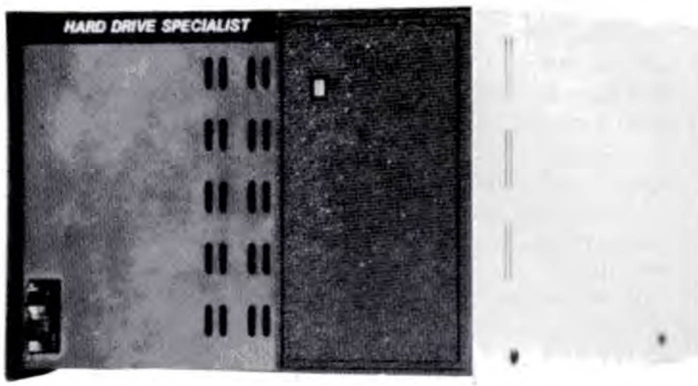
Model III
48K RAM
Disk Basic
Assembly Language
Two Disk Drives
Editor/Assembler

COMPLETE HARD DRIVE $795.

AS SEEN AT THE RADIO SHACK COMPUTER SHOWCASES

TRS-80 Model 1/3/4 and 4P Primary	Secondary
5 Megabyte - 795.00	595.00
10 Megabyte - 1095.00	895.00
15 Megabyte - 1295.00	1095.00
30 Megabyte - 1895.00	1695.00

(Model 1 add 50.00)



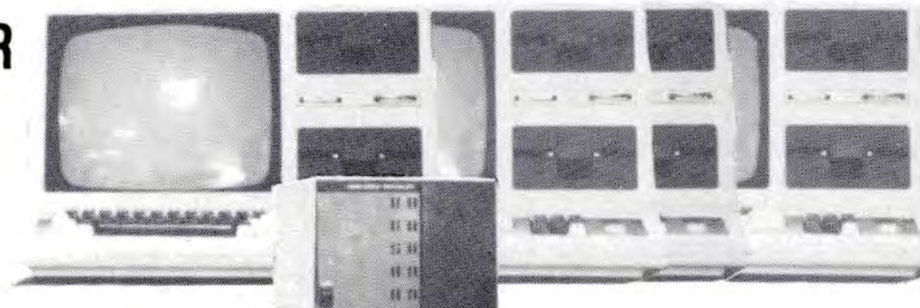
TRS - 80 Hard Disk Subsystem Features

- All sizes rated after formatting.
- TRS-80 Models 1, 3, 4, and 4P full supported.
- Your choice of DOSPLUS, LDOS, or TRSDOS 6.x Software Drivers included.
- Newdos 80/Version 2.5 Systems available.
- Up to two secondary drives may be added.
- Error checking and correcting controller.
- Buffered seek drives for improved access times.
- Built-in power up diagnostics.
- Drives use plated media for long disk life.
- Heavy duty power supply.
- Fan cooled.
- Gold connectors used throughout for high reliability.
- 30 day money back guarantee.
- 1 year warranty.

Shop and compare. Hard Drive Specialist has been building hard drive systems for years and have sold thousands of subsystems to satisfied TRS-80 customers. We fully support TRS-80 Models 1/3/4 and 4P computers on all available hard drive operating systems. We use the latest state-of-the-art drives and controllers. Our drives all use buffered seek logic and plated media to result in almost one-fourth the average access times found on our competitors drives. Plated media results in longer platter life and high resistance to head crashes not found on coated media drives. We utilize high reliability construction throughout including, heavy duty power supply, cooling fan, and gold connectors. We back up our drives with a one year parts and labor warranty and we're so sure you'll like our systems we offer a thirty day money back, no questions asked, guarantee. So when it comes time to buy, look beyond the price and choose the best drive. After all, we are the Hard Drive Specialist!

HDS MULTIPLEXER

$1295.



ALLOWS USEAGE OF UP TO 4 MODEL III OR IV SYSTEMS USING DOSPLUS OR LDOS ON A HDS HARD DRIVE. INCLUDES MASTER CONTROL UNIT, HOST ADAPTERS FOR 2 COMPUTERS, AND 40 FEET OF CABLE. EACH ADDITIONAL HOST ADAPTER ADD $150. EACH ADDITIONAL FOOT OF CABLE ADD $1. UP TO 100 FEET PER COMPUTER.

NEW!

Tandy 1000, Tandy 1200, IBM, Zenith, and Compaq Fixed Disk Drives

10 Megabyte - Internal — $ 895.
10 Megabyte - External — $1195.
15 Megabyte - External — $1695.
30 Megabyte - External — $2195.
10 Megabyte - Secondary — $ 895.
15 Megabyte - Secondary — $1395.
30 Megabyte - Secondary — $1895.

MODEL 3/4 BOOT ROM $39.95

ALLOWS YOU TO BOOT DIRECTLY FROM YOUR HARD DRIVE FOR DOSPLUS

Hard Drive Software

Dosplus 3.4f (Model 1 only)	$99.95
Dosplus 3.5e (Model 1 drivers for 3.5)	$59.95
Dosplus 3.5f (Model 3/4 drivers for 3.5)	$59.95
Dosplus IVf (Model 4 drivers for IV)	$59.95
Dosplus 4.0a (Model 3/4 only)	$99.95
LDOS (tm) 5.1.x Drivers (Model 1/3/4)	$59.95
TRSDOS 6.0 (tm) Drivers (Model 4 only)	$59.95
Newdos/80 Version 2.5 (Model 1/3/4)	$199.00

HARD DRIVE SPECIALIST

Ordering Information

Use our Watts Line to place your order via Visa, MasterCard, or Wire Transfer. Or Mail your payment directly to us. Any non-certified funds are held until proper clearance is made. C.O.D. orders are accepted, as well as purchase orders from government agencies. Most items are shipped off the shelf, with the exception of Hard Drive products which are custom built. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request.

✓455 1-713-480-6000 Order Line 1-800-231-6671
16206D Hickory Knoll, Houston, Texas 77059

✓ See List of Advertisers on Page 145

The BBS then loads this value into the DE pair, transfers the address of the first character in the string to the IX register with the PUSH HL, POP IX instructions, and loads C with 255 to act as a counter.

The BBS loads HL with the value that's located in the first two positions of the string. C is bumped twice, changing its value to 1. Because deleted messages are marked by being set to a negative value (your computer uses 2's complement notation), if bit 7 of the value in the H register is a 1, it's negative and thus ignored as a killed record.

If the number in this position isn't negative, it's compared with the value that we're searching for by the RST 18H instructions. In effect, RST 18H, in 1 byte, subtracts the contents of the DE register pair from the contents of the HL register pair, ignoring the result yet saving the flag status.

If the message numbers exceed the limit, the BBS removes all deleted messages and text files.

If the RST 18H command sets the zero flag, the BBS has found the message number. But if the Carry flag isn't set, you've gone too far. In either event, the BBS increments the C counter by 1, then divides it by 2 by shifting it one position to the right (SRL C), placing zero in the B register, and moving BC to HL to pass to Basic.

If the message number is still unknown, the BBS bumps IX, to point to the next pair of characters, decrements B, and initiates DJNZ, looping to FSR010 until done. The JP BASIC instruction passes the value in the HL register (the record number that we're seeking) to Basic.

Program Listing 2 shows how Basic interfaces with the code in Program Listing 1. Basic defines FD as the address of SEARCH + 1. Line 430 POKEs the LSB of the number that you're searching for into SEARCH + 1, and the MSB into SEARCH + 2. A call to USR3 returns S equal to either the record number you want or to the next-highest record number.

Lines 2730–2840 make up the close-out routine. This is done first by compressing the number of messages on the board so they don't exceed the maximum set by the sysop. If you're under the limit, write the index back to disk. If, however, the message numbers exceed the limit, the BBS removes all deleted messages and text files. Any excess messages are removed with NW$, a new index string. This new index starts removing messages at the high end of MN$ until NW$ is equal to 2*MX, the maximum number of messages the sysop allows. Everything below that point gets killed, and NW$ is written to disk. ■

Program Listing 1. BBS code that locates message numbers.

```
;          SEARCH FOREWARD FOR SPECIFIED 2-BYTE STRING
;
FSRCH   CALL    PARAM           ;GET VARPTR(MN$)
SEARCH  LD      DE,0000         ;SEARCH STRING
        PUSH    HL
        POP     IX
        LD      C,255           ;STR. POS. COUNTER
FSR010  LD      L,(IX+0)
        LD      H,(IX+1)        ;GET FIRST PAIR FOR
CMPR
        INC     C
        INC     C               ;BUMP STRNG POINTER
        BIT     7,H             ;CHECK FOR NEGETIVE
        JR      NZ,REVR         ;NEGETIVE - DEAD FILE
        RST     18H             ;COMPARE HL/DE
        JR      Z,FSR100        ;FOUND
        JR      NC,FSR100       ;TARGET>SOURCE
REVR    INC     IX
        INC     IX
        DEC     B
        DJNZ    FSR010          ;LOOP TIL FOUND
FSR100  INC     C               ;STRPOS+1
        SRL     C               ;(STRPOS+1)/2
        LD      B,0
        PUSH    BC
        POP     HL
        JP      BASIC           ;PASS TO BASIC
```

End

Program Listing 2. BBS code to interface Program Listing 1 with Basic.

```
430 POKE FD+1,INT(MN/256):POKE FD,MN-(INT(MN/256)*256):
S=USR3(VARPTR(MN$)):RETURN
2730 IF (SN=0) OR (LEN(MN$)=0) THEN SL=0:SH=0:SN=0:NW$="":
GOTO 2820
2740 FOR S=LEN(MN$)-1 TO 1 STEP-2:MN=CVI(MID$(MN$,S,2))
2750 IF (LEN(NW$)<2*MX) AND (MN>0) THEN
NW$=MID$(MN$,S,2)+NW$:P$=CHR$(INT((S+1)/2))+P$:GOTO2770
2760 A$="MSG0000/BBS"+DD$:GOSUB860:KILL A$:SN=SN-1
2770 NEXT:IF LEN(NW$)>1 THENSL=CVI(LEFT$(NW$,2)):
SH=CVI(RIGHT$(NW$,2)):GOSUB220 ELSE SL=0:SH=0:GOTO 2820
2780 FOR P=1 TO SN
2790 GET 1,ASC(MID$(P$,P,1))
2800 PUT 1,P
2810 NEXT:CLOSE
2820 GOSUB190:GET 3,1:LSET SL$=MKI$(SL):LSET
SH$=MKI$(SH):LSET SN$=MKI$(SN):LSET SC$=MKI$(SC+1):LSET
ND$=MKI$(ND):LSET NM$=MKI$(NM):LSETDS$=MKI$(DS):PUT 3,1
2830 FIELD 3,255 AS ZM$:LSET ZM$=NW$:PUT 3,3:CLOSE
2840 RUN
```

End

The BBS Express, 80 Micro's bulletin board system, is open 24 hours a day. Call us at 603-924-6985 to see the finished product. UART parameters are 300 baud, 7-bit words, one stop bit, and even parity.

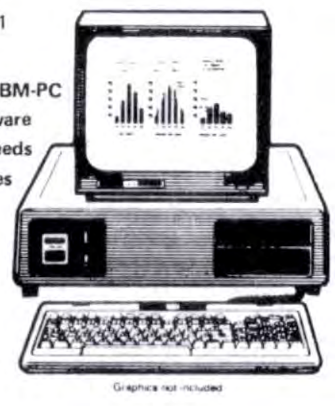
You can reach J. Stewart Schneider and Charles E. Bowen either through their bulletin board at 606-739-6088 or c/o Saturday Software, P.O. Box 404, Catlettsburg, KY 41129.

Stevens' Computer Discount

SAVE

THE ULTRA-HIGH PERFORMANCE MS-DOS PERSONAL COMPUTER

$2295.00 Tandy 2000 26-5103-5111


Graphics not included

- More speed and storage than IBM-PC
- Choose from the hottest software
- User-Expandable for future needs
- 128K, two D/S D/D slim drives
- VM-1 green screen monitor

We also have similar savings on all Radio Shack, Epson, Brother, C'itoh, 3M, Cardco, Hayes, Sanyo and much more

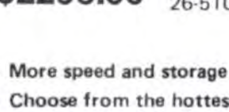
ORDER LINE
215-935-2006

SCD

P.O. Box 525
Phoenixville, PA 19460

SCD

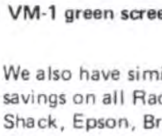
✓442



Power Problems?

SURGES!
SPIKES!
RFI/EMI!
DIPS!
SAGS!
BLACKOUTS!
BROWNOUTS!

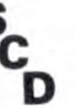
AEGIS® . . . Power Conditioning Equipment . . . THE SOLUTION
Protects From Damaging Voltage Surges, Lost Data, & Costly Down Time



SPIKE-SPIKER®
Transient Voltage Suppressors & Noise Filters
Eleven Models — All Models Rated 120V, 15A

Deluxe Power Console—2-stage transient absorber; dual 5-stage filter; common & differential mode protection; nano seconds response; clamping at 150V; 8 individually switched sockets; fused; main switch; 7' cord and status lite. **$89.95.**

Quad Power Console—6-stage transient absorber; dual 5-stage filter; common & differential mode protection; pico second response; clamping at 131V; four outlets; fused, master switch; 7' cord and status light. **$75.95**



Mini II—*Direct AC Plug-In*; 2-stage transient absorber; dual 3-stage filter; common & differential mode protection; nano second response; clamping at 150V; two outlets and status lite. **$44.95**

LINE-SAVER™
Standby Uninterruptible Power System
—Clean Reliable Power System—

Model LS-240—240 watts—VA capacity; increased back-up time; 11 min. full load, 27 min. ½ load, 43 min. ⅓ load; 4-AC outlets; 3-staged transient protection; dual 4-staged RFI/EMI filter; sealed rechargeable internal battery; master control switch; test switch; external fuses; detachable 6' cord; external DC connectors for mobil use and extended hold-up time; many more exclusive features. **$485.00**

✓485

Call or write for free literature.
Dealer inquires invited.

INSTANT ORDER LINE
800-524-0400 TWX 501-651-2101
IN PENNA. 215-837-0700

KALGLO®
6584 Ruch Rd., Dept 80
Bethlehem, PA 18107

PA Res. add 6% sales tax; for COD add $3.00 + shipping & handling. All pre-paid SPIKE-SPIKER orders, freight allowed. All LINE-SAVER orders add $10.00 shipping & handling.

PROGRAMMING TOOLS FOR YOUR TRS-80™

INSTANT ASSEMBLER

The INSTANT ASSEMBLER is a powerful machine language development system for the TRS-80. One program contains the editor, assembler, and debugger. It allows you to assemble directly to memory and immediately debug your programs 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 the beginner and a super fast development system for the experienced programmer. INSTANT ASSEMBLER has many other unique features. It detects syntax errors as source is entered and stores its source in a compressed format that uses only 40% as much memory as other assemblers. It is incredibly fast and will assemble 10K of object code in just 8 seconds. It will also create relocatable code modules that can be saved on disk or tape and linked together for very large or modular assemblies. INSTANT ASSEMBLER comes with a comprehensive 65 page instruction manual and is available in several versions for different machines. The CP/M version does not run on all machines so inquire before ordering.

TAPE INTASM 2.1 for Model 1 and 3 .... **$39.95**
DISK INTASM 2.1 for Model 1 and 3 .... **$49.95**
DISK INTASM 3.0 for Model 4 .... **$69.95**
DISK INTASM 3.0 for CP/M **$69.95**

INSIDE LEVEL II

This book is a comprehensive reference guide to the Model 1 and Model 3 ROMs which allows the machine language 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. In addition, the 18 chapters include a large body of other information useful to the programmer. 80 Micro said "The book has no flaws; it is a perfect gem." Byte Magazine said "I recommend this book to serious machine language programmers."

INSIDE LEVEL II **$15.95**

DEMON DEBUGGER

DEMON (for DEbugger and MONitor) is a sophisticated tool with which you can explore and debug machine language programs. The STEP mode has 19 different commands and will step through machine language programs one instruction at a time, showing you the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The MONITOR mode has 26 different commands including a labeling disassembler. DEMON is available on tape or disk and includes a comprehensive 40 page manual with many examples.

DEMON 1.0 for Model 1 and 3 .... **$39.95**
DEMON 2.0 for Model 4 .... **$39.95**

DISK INDEX

DISK INDEX will assemble an alphabetized index of every disk in your collection. Simply put each disk in the computer one after another, and DISK INDEX will quickly read the program names and free space left. It will recognize any DOS and disk density (except CP/M) and catalog disks for the Model 1, 3, and 4. The whole index or any selected part may be displayed on the screen or printed on paper in several different formats. Display modes include programs grouped by disk name, by program name, disks in order of decreasing free space, and by program showing the number of backup copies of each one. The index itself may be stored on disk for future access and update. A 48K machine will hold up to 255 disks or over 2000 programs in each file, and you can create as many files as you need. DISK INDEX comes with a comprehensive 28 page instruction manual.

DISK INDEX 3.2 for Model 1, 3, or 4 .... **$39.95**

TELCOM

Mumford Micro offers two telecommunications programs. **TELCOM I** has most of the features needed to communicate with bulletin boards, time share systems, or for file transfers between two disk-based micros over modems or direct wire. It is menu driven and extremely simple to use. **TELCOM II** is an enhanced version of this program, which also supports the LYNC protocol which will automatically detect and correct errors in transmission. Files can be sent to or fetched from an unattended computer. The instruction manuals for both programs are available separately for $5 to help you decide which program is best suited to your needs.

TELCOM I for Model 1, 3, or 4 .... **$39.95**
TELCOM II for Model 1 or 3 .... **$69.95**

MODEL I SPEED UP

The SK-2 is a hardware modification for the Model 1 that allows CPU speeds to be increased by 50% or 100%. Speeds may be changed with a toggle switch (not included) or on software command. It can also be configured to return to normal speed any time a disk is active. It mounts inside the keyboard unit with only 4 necessary connections and is easily removed if the computer ever needs service. The SK-2 has been field proven by 3 years of use, and comes fully assembled with socketed IC's.

SK-2 .... **$29.95**

ORDERING: Ten day money back guarantee. Include $2.50 for postage and handling. California residents add 6% sales tax. Visa, MasterCard and COD orders accepted. SPECIFY MODEL NUMBER. Dealer inquiries invited.

MUMFORD MICRO SYSTEMS

Box 400-E Summerland, California 93067 (805) 969-4557

✓ See List of Advertisers on Page 145

Arrays Make Data Input An Easy Deal

Illustration by John Pirman

A Basic array is a list of related data items, either numbers or strings. You store related data under one variable name, and you usually delimit the number of items the array can accept. For instance, the array A(10) holds 10 data items in the array named A. A primary advantage of arrays is that they let you input large amounts of related data quickly.

Numeric vs. String Arrays

Program Listing 1, Numeric Array, demonstrates how to set an array of numeric variables. It starts with a data statement (line 110) that contains the data the program loads into the array variables. Note the form: I've separated each number in the data statement with a comma.

The dimension statement (DIM) in line 120 creates an array with space for five variables. The program dimensions A to 5 to accommodate the five numbers in line 110.

Line 140 reads the data into array variables A(1)–A(5). To see this, run the program, then print A(1), A(2), and so on.

In Program Listing 2, String Array, the data statement in line 110 contains string data (text) rather than numbers. The strings in a data statement don't need quotation marks, as is necessary when you type in Z$ = "PLUTO".

Line 115 clears 200 bytes for string storage. The Clear statement must come before the dimension statement. Always clear enough bytes to allow for the string space your program needs for arrays, string variables, and literals.

A string array variable must take the form A$(X) unless you first establish it as a string with the DEFSTR statement.

Rules to Remember

A data statement can be as long as your computer accepts (approximately 255 characters). If you run out of space, you can continue a data list on more data statement lines. Basic reads data sequentially, from left to right, from the lowest to the highest line numbers.

You can put data statements anywhere in a program, but you should put them at the beginning of the program for easy reference.

It's not essential that you dimension arrays containing fewer than 11 items, but I recommend including dimension statements for two reasons: You save memory by defining the limits of arrays and it's good programming practice.

The statement DIM A(10) sets up an 11-item array because arrays start with zero. In Listings 1 and 2, Basic sets numeric array A(0) to a zero value and string array A$(0) to a null value.

In some programs you might not start with A(0) for reasons of consistency. For example, in the String Array program in Listing 2, it makes sense to have the array subscript correspond to the month it represents. January is the first month, so you should start with A$(1).

You can assign an array value with subscripts of zero to 10 without dimensioning an array or giving values to other subscripts in the array. For example, the statement A(10) = 10.5 is legal, even if the program doesn't dimension, read, or assign values to A(0)–A(9).

You can dimension several arrays with one DIM statement by using the form DIM A(50), B$(Y), Z1(5). To redimension an array you must precede it with a Clear statement. Failure to do so results in a Redimensioned Array (DD) error.

A Bad Subscript (BS) error results from using subscripts outside the limits of the array. An Out of Data (OD) error occurs when Basic reads a value into an array without sufficient data. (For more information on error messages, refer to Basic Takes, November 1984, p. 158.)

Some programs read data lines more than once. This requires a Re-

---

The Key Box

Models I, III, and 4
Basic

★FREE SHIPPING★
WITHIN CONTINENTAL 48 STATES
ON PREPAID CASH ORDERS ONLY

MICRO IMAGES
INDUSTRIES INC.

MICRO IMAGES
INDUSTRIES INC.

TRY US!

We've Made Retail Prices A Thing of the . . . PAST . . .

Order Toll Free

1-800-223-1182

*THIS IS AN ORDER LINE **ONLY**!*
DIRECT ALL INQUIRIES AND
N.Y. ORDERS TO:
(718) 445-7124

DISK SALE!
NASHUA 5¼ S/S D/D Box 10 17.95
Verbatim MD525 01 S/S D/D Box 10 21.95
Maxell MD-1 S/S D/D Box 10 21.95

$69.95
OMNITERM
by Lindberg Systems
Mdl I/III or IV

$109.95
LAZYWRITER
MDL I, III or IV
SEE LAZY OPTIONS BELOW

DOSPLUS
Mdl I or III 3.5 . . . . 84.95
Mdl IVA 129.95
Mdl II 199.95

ELECTRIC WEBSTER
W/Corr
$118.95

DOTWRITER
4.0 W/Letterset
Mdl I/III or IV
$84.95

NEWSCRIPT
Ver. 7.1
$99.95
W/Label — $109.95

MAXI
CRAS 74.95
MGR II . . . 149.95
STAT . . . 159.95

LE SCRIPT
Mdl I/III or IV
$108.95

DATAGRAPH
Mdl I, III or IV
$69.95
Pie Option—$29.95

COMPUSERVE STARTER KIT
with 5 Free Hours
$33.95

TALLY MASTER
Mdl I or III
$59.95

MULTIDOS
Specify Mdl I or III
Single or Double Density
$49.95

MZAL
REL III
$74.95

FLIP N' FILE
Original-50 Disk
$19.95

COPYART II
Mdl I or III
$84.95
SPECIFY PRINTER

EPSON PRINTERS
CALL FOR BEST PRICES

LETTER QUALITY PRINTERS
DYNAX & TTX
CALL FOR BEST PRICES

PROSOFT UTILITIES
TRASHMAN - Model I or III $34.95
FASTER - Model I or III 26.95
RPM - Model I or III 21.95

LAZY OPTIONS
LAZY FONT (Epson) 44.95
LAZY MERGE 39.95
LAZY DOC 49.95
LAZY Draw, Do, Tab, Calc 34.95

MAIL LIST PROGRAMS
POWER MAIL PLUS-Mdl 2/12/16 . . . . . . 129.95
POWER MAIL PLUS-Mdl I/III 129.95
POSTMAN MASS MAIL-Mdl I/III 74.95
MAXI MAIL-Model III 64.95

Z BASIC - MDL I/III 79.95

MICROTERM - Model I or III 69.95

TZAL — Model III Tape 39.95
BASIC EDITOR - Model I/III Tape 24.95

MAS 80 Acctg System
Integrated G/L, A/R, A/P & Ck Reg.
Retail $599.00 *SPECIAL $399.00*
Overview — $5.00 Credited on Purchase

EPSON PRINTERS . . . . CALL FOR BEST PRICES!
TABCO PAR. DUAL PRINTER SWITCH . . . 69.95
MICRO BUFFER by Practical 16K Parallel or 8K Serial . . . 139.95
NETWORX 4 PLUG SURGE PROTECTOR . . . 59.95

PRINTER STANDS

SPACE AGE	NO FRILLS	
MX 80 No Frills	Clear 14.95	Bronze 19.95
MX80 Space Age	Clear 27.50	Bronze 29.95
Microline 82A Space Age	Clear 27.50	Bronze 29.95
Microline 83A Space Age	Clear 34.95	Bronze 39.95
NEC 8023A Space Age	Clear 27.50	Bronze 29.95
MX100 Space Age	Clear 34.95	Bronze 39.95
Multiuse 18x12x6x¼	Clear 24.95	Bronze N/A
No Frills 132 Col. w/slot	Clear N/A	Bronze 24.95
Gorilla Banana	Clear N/A	Bronze 19.95
FX80 Space Age	Clear N/A	Bronze 34.95
FX100 Space Age	Clear N/A	Bronze 44.95

WE HAVE A STAND FOR MOST PRINTERS

Visit Our Retail Location when in the New York Area

Write or ✓ reader service for your free catalog

RIBBONS
ZIP BOX RELOADS

	½ Doz.	Doz.
Epson MX 70/80-20 Yds	24.00	42.00
Epson MX 100-30 Yds.	30.00	51.00
NEC/Prowriter-14 Yds.	21.00	36.00
Centronics 730/737/739/779 or LP-I/II/IV-16 Yds.	18.00	33.00

All ZIP BOXES are individually sealed black nylon and require no rewinding. Epson Reloads also available in red, blue, brown, green & purple. Any mix allowed.

CARTRIDGES

	Each	Doz.
Epson MX70/80	7.00	60.00
Epson MX100	10.00	108.00
Prowriter 8510 & Nec 8023A	7.50	75.00
RS LP III/V	6.50	66.00
RS LP VI/VIII	6.50	66.00
RS DSY WH II or DWP 410 Multi	6.00	66.00
RS DSY WH II - Nylon	6.50	69.00
RS DMP 2100 - Toshiba 1350	9.00	96.00
MICROLINE 80/82A/83A/92	N/A	30.00
MICROLINE 84 ½ x 40 Yds	5.50	60.00
ANADEX - 9000 Series	10.00	108.00
Diablo Hytype II-Multi Strike	5.50	54.00
Diablo Hytype II - Nylon	6.00	60.00
Qume-Multi Strike	N/A	36.00
NEC—5500/7700-Multi Strike	6.00	60.00
Mannesman Tally 1000	6.50	66.00
TTX 1014	8.00	90.00
Brother HR I (Film)	5.50	60.00
Silver Reed Ex55/77 (Film)	6.50	66.00
Wang System 5	5.00	48.00
Xerox 6/0/620 Memorywriter-Multi	7.00	75.00
IBM DisplayWriter-Multi	8.00	84.00

Note: All cartridges black only.
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 manufacturers in the country.

—QUALITY GUARANTEED—

Micro Images Industries Inc.
164-06 Crocheron Ave. Dept. L
Flushing, N.Y. 11358
(718) 445-7124
HOURS: Mon.-Thurs. 10AM-6PM; Fri. & Sat. 10AM-5PM

VISA ✓157 MasterCard

ORDERING INFORMATION
COD Orders add $2.00. Minimum credit card order $25.00 FREE shipping on all prepaid cash orders within the continental 48 states via UPS Ground. Actual shipping and insurance charges apply on all COD orders, credit card orders, approved P/O's and shipments outside the continental 48 states. Certified Ck., M/O, COD and credit cards shipped immediately. Please allow 2 weeks for personal checks. Credit card orders shipped to card address only. Prices subject to change without notice. New York State residents please add appropriate sales tax.

✓ See List of Advertisers on Page 145

Program Listing 1. Numeric Array program.

```
100 REM * NUMERIC ARRAY * LISTING NO. 1 *
110 DATA 10,6,9,23,15
120 DIM A(5)
130 FOR X=1 TO 5
140 READ A(X)
150 NEXT X
160 END
```

End

Program Listing 2. String Array program.

```
100 REM * STRING ARRAY * LISTING NO. 2 *
110 DATA JAN,FEB,MAR,APR,MAY
120 DIM A$(5)
130 FOR X=1 TO 5
140 READ A$(X)
150 NEXT X
160 END
```

End

Program Listing 3. Template program.

```
100 REM * TEMPLATE * LISTING NO. 3 *
110 DIM A(5)
120 FOR X=1 TO 5
130 REM * THIS LINE MUST BE REPLACED
135 A(X)=X
140 NEXT
150 END
```

End

Program Listing 4. Dimension by Variable program.

```
100 REM* DIMENSIONING BY VARIABLE * LISTING NO. 4
110 CLS
120 INPUT "HOW MANY ITEMS IN ARRAY LIST";X
130 DIM A(X)
140 FOR C=1 TO X
150 PRINT "VALUE NUMBER"C
160 INPUT Z
170 A(C)=Z
180 NEXT C
190 END
```

End

Program Listing 5. Coin Toss program.

```
100 REM * COIN TOSS * LISTING NO. 5 *
110 DIM P(2)
120 FOR X=1 TO 10
130 T=RND(2)
140 REM * MODEL 100, MAKE ABOVE LINE: 130 T=INT(1+RND(1)*2)
150 P(T)=P(T)+1
160 NEXT X
170 PRINT
180 PRINT "SCORE: HEADS:"P(1)" TAILS:"P(2)
190 END
```

End

Program Listing 6. Card Dealer Program.

```
100 REM * CARD DEALER * LISTING NO. 6 *
110 CLEAR 500
120 DIM A$(52)
130 RANDOM
140 REM * MODEL 100, MAKE ABOVE LINE: 130 V=VAL(RIGHT$(TIME$,2)):
FOR T=1 TO V: G=RND(1): NEXT T
150 B=1
160 S$="CDHS"
170 R$="A23456789TJQK"
```

Listing 6 continued

Card Dealer uses arrays to simulate dealing a deck of cards.

store statement to set the data pointer back to the first item in the data list and prevent an OD error.

At the location in the program where you want the pointer to move to the start of the first data line, type RESTORE.

In a program with data on lines 110 and 120, you can type in RESTORE 120 to set the data pointer to the start of line 120 (Model 4 only).

Programming Techniques

Template, in Program Listing 3, demonstrates another way to establish a numeric array. Type in the listing, then type in 130 A(X)=X. This picks up the incremented value of X and passes it to array variable A(X).

Now substitute INPUT "VALUE"; A(X) for line 130. The program waits for you to input a numeric value and press the enter key.

Here's a change for the Models I and III: 130 A(X)=RND(100). This line provides random input, and is useful in programs that check varied outcomes using the same set of random values.

Dimensioning by Variable, in Program Listing 4, lets you input the dimension of the program array. Line 120 prompts you to set the number of array subscripts. Input 10, and line 130 becomes DIM A(10). The For...Next loop in lines 140–180 lets you set values for A(1)–A(10).

Coin Toss, in Program Listing 5, tests probability against results and shows how array values can change during a program run. P(1) is player 1, who wants heads, and P(2) is player 2, who wants tails.

The For...Next loop in lines 120–160 tosses the coin 10 times. Line 130 generates random output. If it outputs a 1, line 150 increments the P(1) score by 1. If it produces a 2, the program adds 1 to P(2). Line 180 prints the ending scores.

Program Listing 6, Card Dealer, uses arrays to simulate dealing a deck of cards. Line 110 clears 500 bytes. Line 120 reseeds the random number generator, and in line 150 B = 1 represents the first card in the unshuffled deck.

Line 160 contains the suits: clubs (C), diamonds (D), hearts (H), and spades (S), and line 170 contains the 13 card ranks.

The suit loop (S) starts in line 180 and the rank loop (R) starts in line 190. The four suits and 13 ranks give a total of 52 cards. Line 200 makes the card into the appropriate midstring values of rank (R$) plus suit (S$). Line 210 increments B by 1 to set up creation of the next card.

The random dealing routine is in the For...Next loop in lines 230–290. Line 230 starts the loop from 1–52 to ensure that all cards are dealt. Line 260 checks the array string to see if it's already dealt (equal to a dash). If so, the program returns to line 240 for another random number.

Line 270 prints the card chosen, and line 280 changes the array variable to a dash so the program doesn't pick it up again. The program displays the cards in the order they're dealt. Run Card Dealer again and they appear in a different order. This program is a good foundation for creating a card game program.

Listing 6 continued

```
180 FOR S=1 TO 4
190 FOR R=1 TO 13
200 A$(B)=MID$(R$,R,1)+MID$(S$,S,1)
210 B=B+1
220 NEXT R,S
230 FOR X=1 TO 52
240 C=RND(52)
250 REM * MODEL 100, CHANGE LINE 240 TO 240 C=INT(1+RND(1)*52)
260 IF A$(C)="-" THEN 240
270 PRINT A$(C);" ";
280 A$(C)="-"
290 NEXT X
300 END
```

End

Multiple-Dimension Arrays

So far I've discussed only single-dimension arrays. Array variable A(4,4,4) is an example of a three-dimensional array representing width, height, and depth. Such an array might be useful in representing Rubic's cube (although it assumes the puzzle has internal cubes).

As a demonstration of multiple arrays, Annual Profit/Loss in Program Listing 7 uses a double-dimensioned array to figure year-end profits or losses.

The months given as data in lines 110–120 become the values of B$(1)—B$(12) in line 160. The DIM statement in line 140 establishes a numeric array, A(12,2), 26 variables, and a string array, B$(12).

Note the mix of string and numeric arrays. In A(12,2), the 12 establishes a row for each month, and the 2 establishes columns for monthly income and expense figures.

The For...Next loop in lines 150–170 reads the months into the B$(X) array. The loop in lines 180–260 prints the name of the month and leads into a loop within a loop in lines 210–240. If the C equals 1, the program inputs income; if it equals 2, the program inputs expenses.

Program Listing 7. Annual Profit/Loss program.

```
100 REM* ANNUAL PROFIT/LOSS * LISTING NO. 7 *
110 DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
120 DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
130 CLS
140 DIM A(12,2), B$(12)
150 FOR B=1 TO 12
160 READ B$(B)
170 NEXT
180 FOR B=1 TO 12
190 PRINT B$(B)" REPORT"
200 PRINT
210 FOR C=1 TO 2
220 IF C=1 THEN INPUT "INCOME";M ELSE INPUT "EXPENSES";M
230 A(B,C)=M
240 NEXT C
250 CLS
260 NEXT B
270 CLS
280 PRINT "PRESS A KEY TO SEE NEXT ENTRY."
290 PRINT "ANNUAL RESULT"
300 PRINT "MONTH";TAB(10);"INCOME";TAB(20);"EXPENSES";TAB(30);
"PROFIT"
310 PRINT STRING$(39,"-")
320 FOR B=1 TO 12
330 T=A(B,1)-A(B,2)
340 PRINT B$(B)TAB(10);A(B,1);TAB(20);A(B,2);TAB(30);T
350 V=V+T
360 Z$=INKEY$
370 IF Z$="" THEN 360
380 NEXT B
390 PRINT TAB(25);"TOTAL";V
400 END
```

End

January's income variable is A(1,1) and its expense variable is A(1,2). The subcripts continue to A(12,1) for income and A(12,2) for expenses for December.

Line 230 makes the array variable worth the value you input for M. (You can combine lines 220–230 by substituting A(B,C) for M in line 230 and deleting line 230.)

The second part of the program prints the results, month by month, as well as the month's profit/loss total. The program obtains the results by subtracting each month's expenses from profits (line 330).

Line 340 displays the information, line 350 increments V by the month's total, and line 390 displays the yearly total.

You can use triple- or even quadruple-dimensional arrays. Multiple-dimensional arrays use up memory, however. A 4-by 4- by 4-dimension array sets aside space for 64 variables.

Next Month

Next month's topic is streamlining programs. Call it packing, call it crunching, it's all about how to make programs shorter and faster. See you then. ■

You can reach Richard Ramella at 1493 Mt. View Ave., Chico, CA 95926.

Teaching Old Basic New Tricks

One of the most fascinating aspects of computer languages like Lisp, Logo, and Forth is that they aren't completely defined; you can customize them to suit your needs. The longer you use them, the more functions your implementation of the language has. The computer constantly "learns" new commands as you program.

Model I/III Basic, on the other hand, seems set in cement because most of its code, as well as Disk Basic's, is stored in unchangeable ROM. Whenever you type in NEW, RUN, or CLEAR, the computer forgets everything you've done and starts with a clean slate. But Basic really isn't set in cement; you can make changes to it. Several commercial products are available that let you extend TRS-80 Basic with new commands. You can also add your own commands to Basic to make it do whatever you want.

During the next few months, I'll describe ways you can add commands to Basic. This month I'll explain some of the fundamental concepts; in future columns I'll include some specific commands as well as methods for hooking them into the Basic interpreter.

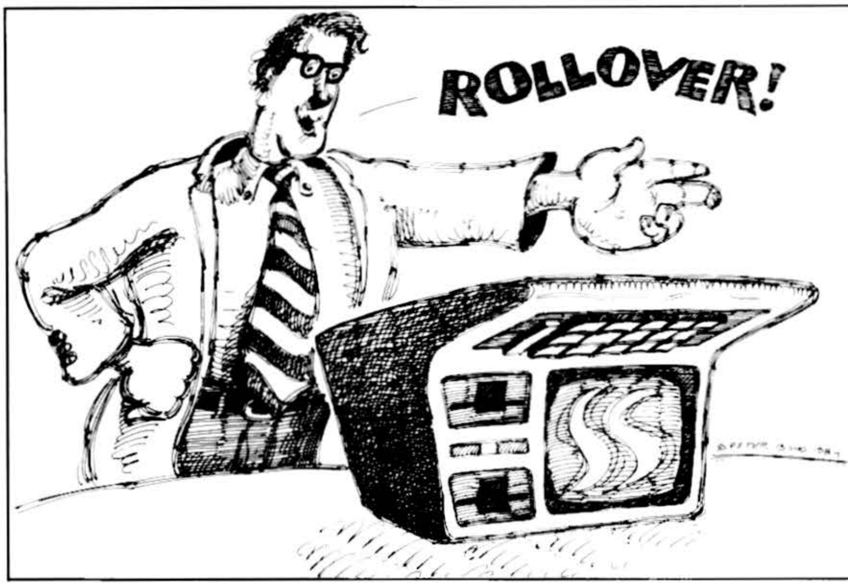


Illustration by Peter Bono

Basic's Structure

As you'd expect from any large program, the TRS-80's ROM Basic is well-organized. When you run a program, the Basic interpreter begins executing each command with the HL register pointing to the last byte that the computer parsed and executed. The interpreter then skips over spaces, tabs, and line feeds to find the next byte in the program, which it places in the A register and uses to determine its next action.

Every Basic command is free to use the Z80's registers as necessary as long as it leaves the correct values in HL and AF and restores the stack when finished. Also, the computer executes every Basic command as a subroutine, and each command ends with a RET instruction unless it discovers a program error. You can write your own Model I/III Basic commands as long as they have the structure of a subroutine. In fact, you can do this in several ways.

The real challenge isn't writing new commands, it's finding an appropriate way to hook them into the Basic interpreter. There are four general solutions. You can substitute a new command for a Disk Basic verb, tack one on to the end of a Disk Basic verb as a new syntax form, link one through a DOS exit vector, or (my favorite) hook one into Basic's command interpreter.

Key Word	Hex Address	Key Word	Hex Address
CVI	4152	FIELD	417C
FN	4155	GET	417F
CVS	4158	PUT	4182
DEF	415B	CLOSE	4185
CVD	415E	LOAD	4188
EOF	4161	MERGE	418B
LOC	4164	NAME	418E
LOF	4167	KILL	4191
MKI$	416A	&	4194
MKS$	416D	LSET	4197
MKD$	4170	RSET	419A
CMD	4173	INSTR	419D
TIME$	4176	SAVE	41A0
OPEN	4179	LINE	41A3

Table. Addresses of the 3-byte vectors for each Disk Basic keyword.

Making the Changes

The Table lists the 3-byte vector addresses for each Disk Basic command

and function. When you use Cassette Basic, it loads each vector with a JP instruction to the L3 error routine (the only exception is that the Model III uses TIME$).

When you run Disk Basic, it loads each vector with a JP instruction to a routine that supports its command or function. You can replace any Disk Basic verb with a new routine by changing the vector's jump address; if you do so, of course, the original function no longer works.

Program Listing 1 demonstrates this technique; the program turns the Disk Basic Name command into a beep command, and produces a short tone through the cassette port or through the internal speaker of a Model 4 in Model III mode (see line 160 for the change for the Model III mode).

Along with the other listings this month, Listing 1 is an example rather than a working program. The new routine sits in the middle of unprotected memory and bombs if Basic overwrites it. (I'll discuss loading and protecting techniques in a future column.)

You can load Listing 1 from DOS Ready and then enter Disk Basic, or load it once you enter Disk Basic. Then use the DEF USR command shown in the comments to hook the routine to Basic. Whenever the interpreter comes across the command Name, it will produce a beep and then continue normally. Since the new routine doesn't require any parameters, it doesn't have to worry about updating the HL register; it merely maintains the value that Basic put there.

Three problems accompany replacing a Disk Basic command with a new routine. First, unless you're using Cassette Basic, you have to give up a function to get a new one. Second, the name you choose for the routine generally has no relationship to its new function. And third, only a limited number of Disk Basic commands are available; if you want to establish a full set of new commands, you'll quickly run out of possible hooks.

The Key Box

LOAD 80

Model III
Assembly Language
Editor/Assembler

Program Listing 1. Demonstration showing how to replace Disk Basic verbs by changing the vector's JP address.

```
               00010 ;Demonstration Routine #1
               00020 ;
               00030 ;Use of the NAME command in Basic will create
               00040 ;a "beep" from the cassette port (or Model 4
               00050 ;internal speaker). Disconnects the normal
               00060 ;NAME command if your Disk Basic uses it.
               00070 ;
               00080 ;To initialize:
               00090 ;  Load from DOS Ready, then enter Basic.
               00100 ;  Type "DEFUSR = &HA000: A=USR(0)"
               00110 ;  Then test by typing NAME.
               00120 ;
418E           00130 NAME    EQU     418EH          ;Addr. of NAME vector
0080           00140 TONE    EQU     80H            ;Change if needed
0080           00150 LENGTH  EQU     80H            ;Change if needed
00FF           00160 PORT    EQU     0FFH           ;Use Port 90H for
               00170 ;                              Mod 4 in Model III mode
               00180 ;  Initialization Section:
A000           00190         ORG     0A000H         ;Pick convenient addr
A000 3EC3      00200 BEGIN   LD      A,0C3H         ;Jump instruction in A
A002 210CA0    00210         LD      HL,START       ;Get routine address
A005 328E41    00220         LD      (NAME),A       ;Put JP in place
A008 228F41    00230         LD      (NAME+1),HL    ;Put addr. in place
A00B C9        00240         RET                    ;Return from USR call
               00250 ;
               00260 ;  Beep Routine called by NAME:
A00C 1680      00270 START   LD      D,TONE         ;Tone and length
A00E 1E80      00280         LD      E,LENGTH       ; into DE
A010 0EFF      00290         LD      C,PORT         ;Output port in C
A012 3E02      00300         LD      A,2            ;Toggle sound off
A014 F3        00310         DI                     ;Interrupts off for tone
               00320 ;
A015 3D        00330 SNDON   DEC     A              ;A = 1
A016 ED79      00340         OUT     (C),A          ;Send it out
A018 42        00350         LD      B,D            ;Get count
A019 10FE      00360         DJNZ    $              ;And wait 1/2 cycle
               00370 ;
A01B 3C        00380 SNDOFF  INC     A              ;A = 2
A01C ED79      00390         OUT     (C),A          ;Send it out
A01E 42        00400         LD      B,D            ;Get count
A01F 10FE      00410         DJNZ    $              ;And wait 1/2 cycle
               00420 ;
A021 1D        00430         DEC     E              ;Drop length count
A022 20F1      00440         JR      NZ,SNDON       ;Repeat until done
A024 AF        00450         XOR     A              ;A = 0
A025 ED79      00460         OUT     (C),A          ;Turn off cassette port
A027 FB        00470         EI                     ;Interrupts back on
A028 C9        00480         RET                    ;Return to interpreter
               00490 ;
A000           00500         END     BEGIN
00000 TOTAL ERRORS
16670  TEXT AREA BYTES LEFT
```

End

Program Listing 2. Demonstration showing how to create the simple beep generator as a Disk Basic verb followed by an exclamation point.

```
               ;Demonstration Routine #2
               ;
               ;Use of CMD! will create a "beep"
               ;from the cassette port (or Model 4
               ;internal speaker). Does not disturb normal
               ;CMD uses.
               ;
               ;To initialize:
               ;  Load from DOS Ready, then Enter Basic
               ;  Type "DEFUSR = &HA000: A=USR(0)"
               ;  Test by typing CMD!
               ;
4173           CMD     EQU     4173H          ;Vector addresses
0080           TONE    EQU     80H            ;Change if needed
0080           LENGTH  EQU     80H            ;Change if needed
00FF           PORT    EQU     0FFH           ;Use Port 90H for
               ;                              Mod 4 internal spkr
               ;
               ;  Initialization Section:
A000                   ORG     0A000H         ;Pick convenient addr.
A000 2A7441    BEGIN   LD      HL,(CMD+1)     ;Get orig. addr.
A003 2215A0            LD      (ORIG+1),HL    ;Stuff address
A006 210DA0            LD      HL,START       ;Get new addr.
A009 227441            LD      (CMD+1),HL     ;Sub ours for orig
A00C C9                RET                    ;Return from USR
               ;
               ;  Beep Routine called by CMD!
A00D F5        START   PUSH    AF             ;Save status flags
A00E 7E                LD      A,(HL)         ;Get value
A00F FE21              CP      '!'            ;Time to beep?
A011 2804              JR      Z,YES          ;Go if yes
               ;
A013 F1                POP     AF             ;Else recover status
A014 C30000    ORIG    JP      $-$            ;Go to orig. routine
               ;
A017 118080    YES     LD      DE,TONE<8+LENGTH ;Tone and len. to DE
A01A 0EFF              LD      C,PORT         ;Port to C
A01C 3E02              LD      A,2            ;Toggle sound off
```

Listing 2 continued

Listing 2 continued

```
A01E F3         00480           DI                          ;Don't interrupt beep
                00490 ;
A01F 3D         00500 SNDON     DEC     A                   ;A = 1
A020 ED79       00510           OUT     (C),A               ;Send it out
A022 42         00520           LD      B,D                 ;Get count
A023 10FE       00530           DJNZ    $                   ;Wait 1/2 cycle
                00540 ;
A025 3C         00550 SNDOFF    INC     A                   ;A = 2
A026 ED79       00560           OUT     (C),A               ;Send it out
A028 42         00570           LD      B,D                 ;Get count
A029 10FE       00580           DJNZ    $                   ;Wait 1/2 cycle
                00590 ;
A02B 1D         00600           DEC     E                   ;Drop length count
A02C 20F1       00610           JR      NZ,SNDON            ;Loop until done
A02E AF         00620           XOR     A                   ;A = 0
A02F ED79       00630           OUT     (C),A               ;Turn off port
A031 FB         00640           EI                          ;Interrupts back on
                00650 ;
A032 D1         00660           POP     DE                  ;Restore stack
A033 23         00670           INC     HL                  ;HL==>Char after "!"
A034 C9         00680           RET                         ;Return to interpreter
A000            00690           END     BEGIN
00000 TOTAL ERRORS
16535   TEXT AREA BYTES LEFT
```

End

Program Listing 3. Demonstration showing how to add a beep function to Disk Basic by interrupting the Basic interpreter as it starts to parse a command.

```
                00010 ;Demonstration Routine #3
                00020 ;
                00030 ;Use of a '!' will create a "beep"
                00040 ;from the cassette port (or Model 4
                00050 ;internal speaker). Operates by intercepting
                00060 ;control from Basic's command interpreter.
                00070 ;
                00080 ;To initialize:
                00090 ;  Load from DOS Ready, then enter Basic.
                00100 ;  Type "DEFUSR = &HA000: A=USR(0)"
                00110 ;  Test by typing !
                00120 ;
4003            00130 INTERP    EQU     4003H               ;Interpreter link
0080            00140 TONE      EQU     80H                 ;Change if needed
0080            00150 LENGTH    EQU     80H                 ;Change if needed
00FF            00160 PORT      EQU     0FFH                ;Use Port 90H for
                00170 ;                                     Mod 4 internal spkr
                00180 ;
                00190 ;  Initialization Section:
A000            00200           ORG     0A000H              ;Pick convenient addr
A000 2A0440     00210 BEGIN     LD      HL,(INTERP+1)       ;Get orig. addr.
A003 2218A0     00220           LD      (ORIG+1),HL         ;Stuff into program
A006 210DA0     00230           LD      HL,START            ;Get our address
A009 220440     00240           LD      (INTERP+1),HL       ;Sub ours for orig.
A00C C9         00250           RET                         ;Return from USR
                00260 ;
                00270 ;  Beep Routine:
A00D E3         00280 START     EX      (SP),HL             ;Get top of stack
A00E 7D         00290           LD      A,L                 ;Get lsb of return addr.
A00F FE5B       00300           CP      5BH                 ;  and test it
A011 2003       00310           JR      NZ,NO               ;Go if not xx5BH
A013 7C         00320           LD      A,H                 ;Get msb of return addr.
A014 FE1D       00330           CP      1DH                 ;  and test it
A016 E3         00340 NO        EX      (SP),HL             ;Restore stack & HL
A017 C20000     00350 ORIG      JP      NZ,$-$              ;Go if wrong return addr.
                00360 ;
A01A CD781D     00370           CALL    1D78H               ;Get next character
A01D FE21       00380           CP      '!'                 ;Beep requested?
A01F 2803       00390           JR      Z,YES               ;Go if it is
A021 2B         00400           DEC     HL                  ;Else fix HL
A022 18F3       00410           JR      ORIG                ;And leave
                00420 ;
                00430 ;"!" Was found
A024 118080     00440 YES       LD      DE,TONE<8+LENGTH  ;Tone and len. in DE
A027 0EFF       00450           LD      C,PORT              ;Port address to C
A029 3E02       00460           LD      A,2                 ;Toggle for off
A02B F3         00470           DI                          ;Don't interrupt beep
                00480 ;
A02C 3D         00490 SNDON     DEC     A                   ;A = 1
A02D ED79       00500           OUT     (C),A               ;Send it out
A02F 42         00510           LD      B,D                 ;Get tone count
A030 10FE       00520           DJNZ    $                   ;Wait 1/2 cycle
                00530 ;
A032 3C         00540 SNDOFF    INC     A                   ;A = 2
A033 ED79       00550           OUT     (C),A               ;Send it out
A035 42         00560           LD      B,D                 ;Get tone count
A036 10FE       00570           DJNZ    $                   ;Wait 1/2 cycle
                00580 ;
A038 1D         00590           DEC     E                   ;Drop length count
A039 20F1       00600           JR      NZ,SNDON            ;Loop until done
A03B AF         00610           XOR     A                   ;A = 0
A03C ED79       00620           OUT     (C),A               ;Turn off port
A03E FB         00630           EI                          ;Interrupts back on
                00640 ;
A03F 23         00650           INC     HL                  ;Point to char after !
A040 C9         00660           RET                         ;Return to interpreter
A000            00670           END     BEGIN
00000 TOTAL ERRORS
16253   TEXT AREA BYTES LEFT
```

End

It doesn't matter which Disk Basic command you choose; the idea is to create a new syntax that would normally be impossible.

Another technique is to add to the syntax of an existing Disk Basic command. It doesn't matter which Disk Basic command you choose, although CMD is the most likely candidate. The idea is to create a new syntax that would normally be impossible.

Program Listing 2 shows how you can create the simple beep generator as "CMD!" (or almost any other Disk Basic verb followed by an exclamation mark). However, this technique only works if you pick a Basic command normally used as the first or the only command in a Basic function. Don't try to use commands like INSTR that can only occur after an equals sign.

Written this way, the new command is essentially a filter. When the program invokes the Disk Basic verb associated with the command, it checks to see if the next character(s) is part of the new syntax. If not, control passes to the regular routine for that Basic verb. If you're invoking the new routine, it performs its operations, sets HL properly, and uses a Return instruction to get back to the normal flow of Basic.

Listings 1 and 2 differ in two important ways. First, the initialization section of Listing 2 doesn't destroy the original address in the CMD vector; instead, the program saves that address (at ORIG + 1 in line 430) so that it can jump to DOS's CMD routine when a command other than "CMD!" is interrupted.

Second, the body of the beep routine starts by saving the AF registers on the stack (line 370). Before the interpreter calls a Basic command, the computer processes the character following the command token (the 1-byte representation of the command verb). Basic points the HL register to that next byte, then sets the status flags to indicate whether the byte is a colon (Z

flag set), numeric (C flag set) or neither (NZ and NC set).

Many routines start by examining the flags as a first check for correct syntax. If the flags were not correctly restored in line 420, the normal CMD processor would likely report a syntax or type mismatch error.

Program Listing 3 shows a third way to add a beep function to Disk Basic. The program works by interrupting the Basic interpreter as it starts to parse a command. If it finds your special command, the program executes it; otherwise control passes to the normal Basic interpreter.

Basic begins parsing and executing each command in ROM at 1D5A hexadecimal (hex) with an RST 10H instruction. From 0010 hex, the program sends control to RAM location 4003 hex. If you're using Cassette Basic, TRSDOS, or NEWDOS80, 4003 hex contains the instruction JP 1D78H, which sends control to a routine that finds the next character in the program, and adjusts HL to point to it (LDOS, DOSPLUS, MULTIDOS, and the Stringy Floppy system use a different jump at 4003 hex to perform some special instructions).

Listing 3 interrupts the normal flow of control at 4003 hex. First, starting in line 370, it tests the return address on the stack to be sure that this RST 10 hex call came from 1D5A hex. If the return address isn't 1D5B hex, the program gets out of the way by jumping to the original address stored at 4003 hex.

Basic changes keyboard symbols to a form completely different from their ASCII codes.

If the address on the top of the stack was correct, Listing 3 calls 1D78 hex to find the next valid character and checks to determine if it's an exclamation point. If not, the program decrements the HL register once to point to the previous byte and again control passes to the original address (lines 460–500).

Finally, if the program finds the correct return address, and if the next input character is an exclamation point, the program produces a beep. It ends by pointing HL to the next position after the exclamation point and then returning to the Basic interpreter. You must end the routine with HL and the stack set correctly, and with the Z flag set (line 700).

If you want to experiment with this final technique, select new command names carefully. Unless your program carefully checks each name, you must make sure that each new command starts with a character that could not normally appear at the beginning of a Basic command. If, for example, you chose a command A, you could never use a variable that started with the letter A without generating a syntax error.

Also, Basic changes keyboard symbols (such as the arithmetic operators) to a tokenized form completely different from their ASCII codes. If you use one of those symbols as your command name, your program will have to check for the tokenized form.

Coming Up

Next month, I'll discuss another technique for adding new commands to Basic, and start to develop a program that lets you add many new Basic commands in a single program.

Remember that if you write to me and want a reply, you must enclose a stamped, self-addressed envelope. I am happy to try to answer your questions about Assembly language or use of your Model I, III, or 4. However, because of the amount of mail I receive, I cannot respond if you don't include a stamped envelope. In any case, I always welcome your comments, questions, and suggestions. ■

Readers who subscribe to CompuServe can ask questions or take part in discussions on topics covered by "The Next Step." Go PCS-117 to the Writers' and Editors' Special Interest Group (WESIG) and leave your questions or comments addressed to Hardin Brothers. Feel free to join in conversations started by others.

You can also write to Hardin at 280 N. Campus Ave., Upland, CA 91786. Enclose a stamped, self-addressed envelope if you want a reply.

80 LIST OF ADVERTISERS

Reader Service Number	Advertiser	Page
82	Aerocomp Inc.	118, 119
215	Alcor Systems	123
136	Allen Gelder Software	114
140	Allied Systems Co.	161
476	Alpha Bit Comm., Inc.	92
17	Alpha Products Co.	14, 15, 17
74	ALPS	33
303	Alternate Choice	37
342	Amherst Equipment Corp	81
141	Anitek Software Products	23
400	Anthropomorphic Systems	157
390	Applied Microsystems Inc.	13
3	Argonaut Systems	143
102	Armstrong Genealogical Systems	173
16	Autospec	165
383	Astro-Star Enterprises	88
152	BCCOMPCO	157
*	Beck Mfg	103
7	Bible Research	165
168	BlueRidge Software	59
381	Bodex Corp.	53
300	BT Enterprises	91
299	CB Microdex	173
86	Compulogic	169
357	Computer Friends	89
18	Computer Plus	99
45	Cornucopia	33
223	Crest Software	90
539	Desert Sound	143
27	Design Software	90
282	DFW Computer Center	37, 81
367	Digital Images	33
204	Diskcount Data	104, 105
*	Diskette Connection	169
48	Disks-N-Data	77
62	Displayed Video	159
216	EAP Co.	125
178	EZWare Corp.	123
125	EC Data/Tepra International	172
85	Educational Micro Systems	77
45	Electric Webster	33
*	*80 Micro*	
	Load 80	96
	Mailing list	92
	Moving	97
	Subscription Ad	64
	Subscription problems	173
134	Elek-Tek	164
144	Elek-Tek	127
172	Engineering Management Consultants	173
350	EnFleur Corporation	175, 102
214	Fort Worth Computers	59, 155
496	Good Software Corp.	169
103	Good Software Corp.	171
185	Gooth Software	155
455	Hard Drive Specialists	133
61	Harmon Hardware & Supply	88
355	HDP	72
9	H&E Computeronics	CIV, 67, 69
175	Howe Software	47
425	Indiana Software Development	125
148	Intelligent Technologies Co.	143
534	Jameco Electronics	53
244	Jaste Enterprises Inc.	175
101	J&M Systems	63
126	JMG Software International	31
229	JRM Publications	110
52	JStar Enterprises	173
485	Kalglo Electronics	135
462	Langley-St. Clair, Inc.	49
161	Logical Place/K & S Comp. Ser.	49
131	Logical Systems	165
251	Logical Systems	7
301	Logical Systems	111
115	Lynn Computer Service	71
112	Joe Lynn Computing	85
218	Magnetics	97
250	Marymac	81
111	McSystems	72
*	Micro Data Supplies	25, 26, 27, 28
157	Micro-Images	137
464	Micro-Labs Inc.	143
181	Micro Systems Software	167
137	Miller Microcomputer Services	73
317	Mimar Inc.	108
411	Montezuma Micro	40, 41
416	Montezuma Micro	150, 151
*	Mumford Micro Systems	135
379	Nautilus Computers Inc.	92
*	New Classics Software	109
281	Nibblenotch	92
232	Nocona Electronics	87
54	Nodvill Software	163, 175
95	Northwest Computer Algorithms	131
127	Noteworthy Software Inc.	175
36	Omnisoft Research	77
201	Optimal Tech. Inc.	155
151	Orion Instruments	49
207	Pacific Exchanges	102, 173
324	Pel/Tek	30
124	Perry Computers	147
143	Perry Computers	115
176	Personal Computer Products	162
366	Pica	149
306	Powersoft	54
260	Pro/Am Software	129
76	Producer, The	2, 3
449	Prof Jones/Frogg House	93
42	Programming & Systems Management	170, 135
174	Professional Tax Software	74
8	Prosoft	55
1&2	Prosoft	4
91	Prosoft	9
30	Prosoft	10, 11
51	Prosoft	175
194	Quant Systems	74
117	Quick Labels Systems	175
75	Radio Shack	CII, 1
129	Remsoft Inc.	75
265	Ridge-Tech Inc.	59
512	Rockware Data Corp.	32
343	Rogo Computer Products	102, 163
53	Rose Electronics	173
368	Safeware	129
203	Scientific Engineering Labs	163, 175
258	Smart Data Inc.	171
252	Smart Data Inc.	173
192	Software Concepts	53
104	Software Factory, The	175
80	Software One	110
*	Software Support	94, 95
64	Southworth	173
444	Southwestern Digital	153
213	Standard Software	164
43	Star Software Co.	108
442	Stevens Computer Discount	135
71	Sublogic Communications Corp.	129
285	Sun Research Inc.	131
456	Sunlock Systems	125
189	Tab Sales	168
347	Talley Communications	144
266	T/Maker	CIII
81	Total Access	20
227	Trisoft	97
32	TVG Systems	123
169	Unicorn Software	131
*	Vespa Computer Outlet	113
211	United Software Associates	164
41	Warriors & Warlocks	175
37	Wiley Inc.	174
123	Wilson Technical	33
79	Zygotron	173

For further information from our advertisers, please use the Reader Service card.
**This advertiser prefers to be contacted directly.*

Advertising Sales (603) 924-7138
or (800) 441-4403

West Coast Sales (415) 328-3470

A Rematch Between Two Data Base Systems

I'll devote part of this column to clearing up a problem of transferring graphics-based software between the Model I/III and the Tandy 2000. I'll also revisit an earlier topic: my comparison of Profile III+ to dBase II.

Illustration by Phil Geraci

Customized Graphics

Now that you have a tool for converting Model I/III Basic software to run on the Tandy 2000 (see October 1984, p. 178), you may have encountered another problem: Some programs won't convert because they use Model I/III-specific graphics. Paul Opitz's CHAREDIT program in the final issue of the *TRS-80 Microcomputer News* (volume 6, issue 6) solves the problem by letting you customize the Model 2000 character set.

While using Opitz's program to design a complete Russian alphabet, I realized that I could also replicate the Model I/III graphics character set on the Model 2000. The set I created replaces Model 2000 characters 128–191 with the appropriate Model I/III graphics characters. That allowed me to convert many of my Basic programs without any reprogramming. Unfortunately, this method doesn't work with programs that depend on the Set, Reset, and Point commands to display graphics.

The Program Listing (p. 149) supplies Basic subroutines that duplicate the Model I/III's Set, Reset, and Point functions on the 2000. With these subroutines saved to disk in ASCII format, it's a relatively simple (albeit tedious) task to change Model I/III graphics commands to linkages for these subroutines. I recommend editing the ASCII text of the program with a capable word processor.

The GRINIT routine is the heart of the program. It loads the Model I/III graphics character set modified by CHAREDIT and sets up the byte masks for later use. The next three routines are all very similar; they share much of the same code. I chose not to put this code in one more subroutine because another function call would have slowed the program down too much. To increase speed, remove all comments and any extraneous blanks, combining as many statements on one line as possible. Table 1 lists the program's variables. The Figure uses a typical character cell divided into graphics pixels to illustrate the variables and their use.

If you'd like the character set editor (CHAREDIT and FONTINIT), the character sets (Russian and Model I/III graphics), and/or the graphics routines in the Listing, send a disk and a self-addressed mailer to me at 1519-A Carswell Circle, Bolling Air Force Base, Washington, DC 20336. Make sure the mailer has correct return postage. For the character set editor, specify the type of display you're using (monochrome/color, graphics/no graphics).

Battle of the Data Bases—Round Two

In my first column (August 1984, p. 177), I illustrated dBase II's versatility by comparing its major features to those of Profile III+. I must admit that this was unfair; comparing programs that run under different operating environments is never really fair. However, the Small Computer Co. called my attention to a very powerful extension to the Profile III+ system called filePro-16 (see Table 2). FilePro-16 is available for the Model 2000 and is a single-user, feature-for-

```
 ----------------
|        |        |
|        |        |   Ø   R
|        |        |       o
|        |        |       w
 ----------------
|        |        |       N
|        |        |   1   u
|        |        |       m
|        |        |       b
 ----------------         e
|        |        |       r
|        |        |   2
|        |        |
|        |        |
 ----------------

     Ø        1

  Column   Number
```

Figure. Screen coordinate algorithm.

feature adaptation of the multiuser Profile-16 system currently available for the Model 16.

FilePro-16 contains an extensive reference manual, program disks, an installation manual, and a start-up manual with a beginner's and an advanced tutorial. The system is huge. Hard disk Model 2000 users will generally have no problem with the program's size, but floppy users will be dismayed at having to use separate creation and runtime disks, as with Profile III+. Remember, Model 2000 disks hold four times as much as Model III disks, but filePro-16's code still doesn't fit on the two program disks—the special character print codes are on another disk.

FilePro-16 is a superior product. The limitations of Profile III+ have been corrected. For example, you can add data and lengthen or shorten any data base file easily. It automatically updates indexes once you define them. You can also build very powerful menus that let you structure applications for easy use by someone unfamiliar with the filePro-16 commands.

FilePro-16's added features are very capable: the ability to spread files over several disks, flexible and numerous input/output format features, and excellent precision in math calculations. FilePro-16 allows over 16 million records per file, and 4,608 characters per record.

With all of these exceptional features in mind, I'll reconsider my original comparison of data base management systems, this time pitting dBASE II against filePro-16. In each of the areas mentioned above, filePro-16 exceeds dBase II's specifications or provides an easier way to perform an operation. Why, then, is there so much demand for dBase II? First, it's essentially universal, providing identical capabilities to any computer that runs CP/M or MS-DOS. Don't discount this feature; it's a most important consideration when you have access to multiple computers.

Probably more important, dBase II, while providing all the data base management functions most people need, is really more of a programming environment than a DBMS. This product's great success derives from its superb flexibility in developing applications. An entire software sub-industry is based on dBase programming.

Variable	Description
ZX	Specifies the horizontal location of the graphics pixel (0-159)
ZY	Specifies the vertical location of the graphics pixel (0-71)
CX	Character location within the selected row containing the pixel (1-80). This is the integer quotient of ZX/2.
CY	Character location within the selected column containing the pixel (0-24). This is the integer quotient of ZY/3.
CL	Selects the graphics column by examining the remainder from the calculation of CX.
RW	Selects the graphics row number by examining the remainder from the calculation of CY.
BT	Specifies the bit number within the byte for the action to be performed. This value ranges from 0 to 5 and is used to select the appropriate mask from either the SM() or RM() arrays.
CH	Contains the value of the character located in the screen memory at the location specified by CX and CY.

Table 1. Variables used in the Program Listing.

Since these two systems cost the same, I must recommend dBase II as the DBMS system of choice because of its greater wealth of support and portability of data and applications. However, if you also use Profile-16 on the Model 16 computer, you may find filePro more appropriate. These two products are virtually identical and provide unparalleled compatibility. Were filePro-16 less expensive, I'd heartily recommend it for use on the Model 2000.

Making dBASE II Better

While I'm on the subject of dBase, I must mention two superior products that I've encountered: DB/RA and dBRx, powerful extensions that overcome some of dBase II's data handling limitations. Although designed specifically for the IBM PC, both run fine on the Tandy 2000.

DB/RA adds the power of one-, two-, or three-dimensional arrays to dBase-II. You can use these arrays with character, logical, or numeric data just as you'd use any other memory variable. In fact, this software extends the number of memory variables available from 64 to over 65,000. A simple statement beginning with the key word Call performs these functions. DB/RA incorporates its own syntax analyzer so these special call statements resemble familiar dBase commands.

DBRx is a math and string extension package making several transcendental functions (sine, cosine, logarithmic, etc.) available for use in data computations. This means you can do most complex calculations within the DBMS. The package also supplies three string functions (Pack, BTRIM, and NOSPACE) that enhance the appearance of text data when you display it or print it out. Pack compresses multiple blanks into one, BTRIM trims blanks from the left side of a character string (as opposed to the normal Trim function for removing blanks from the right side), and NOSPACE removes all spaces from the text string. ■

dBase II Data Base Manager
Tandy/Radio Shack
One Tandy Center
Fort Worth, TX 76102
26-5352
$495

filePro-16 Data Base Management System
The Small Computer Company, Inc.
230 W. 41st Ave., 12th Floor
New York, NY 10036
Available through Radio Shack Express Order system
$495

dBRx—The dBase II Math/Stat Program
dB/RA—The dBase II Array Program
Gryphon Microproducts
P. O. Box 6543
Silver Spring, MD 20906
dBRx; $150
dB/RA; $200

Table 2. Model 2000 product index.

You can reach John B. Harrell III c/o this column, 80 Micro, *80 Pine St., Peterborough, NH 03458, or via CompuServe at 73016,1326.*

Program Listing. Graphics conversion routines.

```
8800 ' ************************************************************************
8810 '          GRINIT - set up values for the Model-III graphics routines
8820 '
8830 '          No parameters are expected and the values returned are in the
8840 '          arrays SM() and RM() used by the graphics routines.
8850 '
8860 '          GRINIT should only be called once per program execution or an
8870 '          error will occur when attempting to redimension the arrays.
8880 '
8890 '          It may also be necessary to reposition your DATA statements
8900 '          to prevent conflict with GRINIT.
8910 '
8920 ' ************************************************************************

9000 DIM SM(5),RM(5)
9010 RESTORE 9060                        'set up proper location for DATA  read
9020 FOR I=0 TO 5
9030    READ SM(I),RM(I)                 'read the data masks
9040 NEXT I
9045 BLOAD"mdiiiact"                     'load the graphics character set
9050 RETURN
9060 DATA 1,254
9070 DATA 2,253
9080 DATA 4,251
9090 DATA 8,247
9100 DATA 16,239
9110 DATA 32,223

9900 ' ***********************************************************************
9910 '          SET a graphics pixel using Model-III graphics
9920 '                 call by setting ZX and ZY to the location of the pixel
9930 '                 and execute using a GOSUB
9940 '                 ZX must be in the range of 0 to 159
9950 '                 ZY must be in the range of 0 to 71
9960 '
9970 '          This subroutine requires initialization by calling GRINIT
9980 '
9990 '************************************************************************

10000 CX = INT(ZX/2) + 1: CL = ZX AND 1         ' calculate col (cx) and gr col
10010 CY = INT(ZY/3) + 1: RW = ZY - 3 * (CY-1)  ' calculate row and gr row
10020 BT = RW + RW + CL                         ' bit position within the gr
byte
10030 CH = SCREEN(CY,CX)                        ' get char at current screen pos

10040 IF CH < 128 OR CH > 191 THEN CH = 128     ' if not graphics -- make blank
10050 CH = CH OR SM(BT)                         ' set proper bit to turn on
pixel
10060 LOCATE CY,CX: PRINT CHR$(CH);             ' locate and replace screen char

10070 RETURN

19900 ' ***********************************************************************
19910 '          RESET a graphics pixel using Model-III graphics
19920 '                 call by setting ZX and ZY to the location of the pixel
19930 '                 and execute using a GOSUB
19940 '                 ZX must be in the range of 0 to 159
19950 '                 ZY must be in the range of 0 to 71
19960 '
19970 '          This subroutine requires initialization by calling GRINIT
19980 '
19990 '************************************************************************

20000 CX = INT(ZX/2) + 1: CL = ZX AND 1         ' calculate col (cx) and gr col
20010 CY = INT(ZY/3) + 1: RW = ZY - 3 * (CY-1)  ' calculate row and gr row
20020 BT = RW + RW + CL                         ' bit position within the gr
byte
20030 CH = SCREEN(CY,CX)                        ' get char at current screen pos

20040 IF CH < 128 OR CH > 191 THEN CH = 128     ' if not graphics -- make blank
20050 CH = CH AND RM(BT)                        ' set proper bit to turn on
pixel
20060 LOCATE CY,CX: PRINT CHR$(CH);             ' locate and replace screen char
20070 RETURN
29900 ' ***********************************************************************
29910 '          TEST a graphics pixel using Model-III graphics
29920 '                 call by setting ZX and ZY to the location of the pixel

29930 '                 and execute using a GOSUB
29940 '                 ZX must be in the range of 0 to 159
29950 '                 ZY must be in the range of 0 to 71
29960 '
29970 '          This subroutine requires initialization by calling GRINIT
29980 '          The function return value is returned in variable ZZ:
29981 '                 ZZ = -1 (TRUE) -- the pixel is currently set
29982 '                 ZZ = 0 (FALSE) -- the pixel is currently reset
29983 '
29990 '************************************************************************

30000 CX = INT(ZX/2) + 1: CL = ZX AND 1         ' calculate col (cx) and gr col
30010 CY = INT(ZY/3) + 1: RW = ZY - 3 * (CY-1)  ' calculate row and gr row
30020 BT = RW + RW + CL                         ' bit position within the gr
byte
30030 CH = SCREEN(CY,CX)                        ' get char at current screen pos

30040 IF CH < 128 OR CH > 191 THEN CH = 128     ' if not graphics -- make blank
30050 CH = CH AND SM(BT)                        ' set proper bit to turn on
pixel
30060 ZZ = (CH <> 0)                            ' check if bit is set and return

30070 RETURN
```

End

✓ See List of Advertisers on Page 145

SAVE A BUNDLE WITH ONE OF OUR BUNDLES

We have put together special bundles of software just in time to save you a bundle of money. Didn't you ever wonder why CP/M was so popular? The reason is the software availability. All programs (except EP) are for the Model 4/4P and have been optimized to install and run without hassle. For example, we have memory-mapped WordStar® and it runs circles around the standard version available elsewhere. We also added printer drivers for the Daisywheel II and the DMP-2100. Our CP/M® 2.2 is the best around. Read the reviews. You *know* the programs. You *know* the prices. This is the years' best value. You owe it to yourself to let these CP/M® programs and thousands of others (many in the public domain) start serving your needs.

Bundle #1
WordStar®
MailMerge®
SpellStar™
StarIndex™
Montezuma Micro CP/M®
$299 Delivered
RETAIL VALUE $1009

Bundle #5
Electric Pencil™
Red Pencil™
Blue Pencil™
$149 Delivered
Works with TRSDOS, NEWDOS-80, DOSPLUS and MULTIDOS (Model 1,3,4)
RETAIL VALUE $240

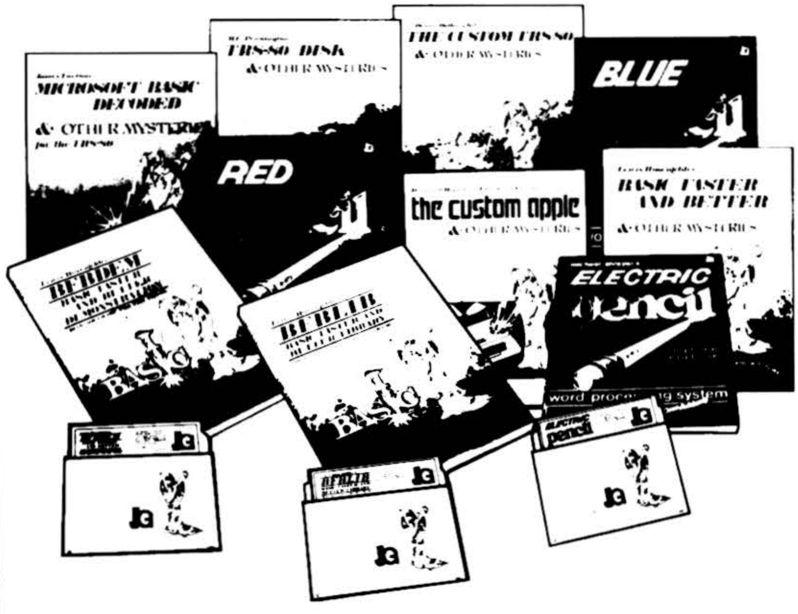
Bundle #2
WordStar®
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
$399 Delivered
RETAIL VALUE $1354

Bundle #3
WordStar®
MailMerge®
SpellStar™
StarIndex™
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
$499 Delivered
RETAIL VALUE $1699

Bundle #4
WordStar®
MailMerge™
SpellStar™
StarIndex™
dBASE II®
TurboPascal™
Montezuma Micro CP/M®
$599 Delivered
RETAIL VALUE $1553

Save A Bundle on these books and disks

Take advantage of our volume discounts. Buy any three items from this list and deduct $5 from your total order. Buy four...deduct $10. Buy five...deduct $15. Buy six...deduct $20 and so on. Buy a bunch, save a bundle. Please add $1 each for shipping. Add $5 to orders under $50 for handling.



TRS-80 Disk & Other Mysteries. The "How to" book of data recovery for the TRS-80 Model I disk operating system. 128 pages. Retail $22.50 **Now $18**

Microsoft BASIC Decoded & Other Mysteries. The complete guide to Level II and BASIC. 312pages. Retail $29.95 **Now $24**

The Custom TRS-80 & Other Mysteries. The complete guide to customizing TRS-80 Hardware and Software. 336 pages. Retail $29.95 **Now $24**

BASIC Faster & Better & Other Mysteries. The complete guide to BASIC programming tricks and techniques. 290 pages. Retail $29.95 **Now $24**

BASIC Faster & Better Library Disk. Contains 121 functions, subroutines and user routines. Search merge, compare and listing routines plus array handlers, BASIC overlays and video drivers. Retail $19.95 **Now $16**

BASIC Faster & Better Demonstration Disk. Contains 32 demos of the Library Disk contents above. Retail $19.95 **Now $24**

BASIC Disk I/O Faster & Better & Other Mysteries. Programming tips and techniques to store/retrieve data from disk. 432 pages. Retail $29.95 **Now $24**

BASIC Disk I/O Faster & Better Demonstration Disk. All of the major demo programs and library of disk I/O subroutines in 25 BASIC programs. Random, Indexed Sequential and TREESAM file handlers included. Retail $29.95 **Now $24**

Machine Language Disk I/O & Other Mysteries. A guide to machine language disk I/O for the TRS-80. 288 pages. Retail $29.95 **Now $24**

TRSDOS 2.3 Decoded & Other Mysteries. Detailed explanation of the Model I disk operating system. 298 pages. Retail $29.95 **Now $24**

How To Do It On The TRS-80. A complete applications guide to the TRS-80 Model I, II, III, 100, and Color Computer. 352 pages. Retail $29.95 **Now $24**

The Custom Apple & Other Mysteries. Who cares? Retail $29.95 **Now $19**

Due to the nature of this business, there are **NO REFUNDS** on software. We accept American Express, MasterCard, Visa and COD. Your card is not charged until we ship your order. Personal and company checks are accepted without delay (call for details). Delivered prices are for the lower rightmost 48 States only. Prices and specifications subject to change without notice. If you don't understand any of this, just give us a call and we will take the time to explain it until you do. First-timers are always welcome here.

MONTEZUMA MICRO

✓416

214-339-5104
Redbird Airport, Hangar #8
P.O. Box 32027
Dallas, Texas 75232 *"WE KEEP YOU RUNNING"*

ORDER NOW . . . TOLL FREE
800-527-0347
800-442-1310
The Toll Free lines are for orders only
Specifications subject to change without notice

Continued from p. 39

menu is Make Basic Lines, which takes everything already entered (saved in a collection of disk files) and pulls it all together into a single, functioning Basic program. You must answer a few prompts, but the process is nearly automatic. The resulting program contains enough comments so that most people familiar with Basic can modify any of it they choose.

But you aren't finished yet. The Producer has created a program whereby you can input data, store it on disk, and edit it at will. But the report structures need to be defined to make that data useful. Once again there is a menu option, Build Reports, that does most of the work.

While the report generator is flexible, it isn't as impressive as the rest of the package. You select which fields of each record to print and in what order. Also, for each report, you can predefine a range of records used to generate the report, or define prompts so the user can make the selections. You can also define a report to create mailing labels.

The Producer lets you define up to nine different report forms for each data-handling program. The different reports will appear on a menu available to the user of the final program.

The Freeform Report Generator

The format of the reports generated is limited unless you use the Freeform Report Generator (available with the enhanced Producer package). This program is the best part of the entire package. It lets you use a variety of report formats, complete with extra calculations, subtotal fields, and grand totals. You can even combine information developed with two or more Producer-generated programs if you have some Basic programming skill.

While the Freeform Report Generator is the most complex part of The Producer, it's difficult to imagine any data base report it couldn't generate. But, the Basic programs it does create are the only part of the package that operate more slowly than I would like.

When you're finished developing both the data-entry programs and the final report forms (which you can add to or modify at any time), you're ready to put all of it together in a Basic program. The last selection from the main menu is Build Program. After answering a few simple questions, The Producer does all the work for you. You can run the final product either under the tiny DOS supplied or transfer it to a DOS of your choice. You could also compile the program with Microsoft's Bascom, though I haven't tried it.

No matter how complex your program development, the finished product is easy to use. It opens with a combination logo/menu page (you can create your own logo if you want) from which you can enter new data for the data base, edit data previously entered, reorganize the structure of a data base for faster access, perform global search and replace operations on your data, go to a report menu, or exit from the program. All of the standard operations are easy to perform.

Weaknesses

The Producer does have a few weaknesses. First, when defining the record form for the data base, you're limited to information you can fit on a single screen. You can define a total of 32 fields, and the total possible length of each record is limited to 251 bytes. The Producer uses variable-length random-access records to store information, and needs 5 bytes of each record for its binary-tree organization. If you want to use The Producer on NEWDOS80, which allows records longer than 256 bytes, you can alter a Producer program to increase record size, but I haven't yet needed to try.

The system uses a special form of Trashman to speed up string handling and avoid long garbage collection pauses. In general, everything (except reports created with the Freeform Generator) runs fast enough so that all the disadvantages of Basic seem to have been avoided.

The Producer expects that you will want to enter all the string data in uppercase. You can use mixed case easily while creating a program, but only with some unnecessary difficulty in the final program. Such a restriction would make sense on a Model I, but not on a Model III.

The Producer's documentation says that a technical manual was to be available by early 1984, but when I called about it, I was told that no one had even started to write it yet. A quarterly newsletter, as well as technical help by phone, is available to registered owners.

Overall, The Producer is an excellent package. I've used it to create three separate and very different types of data bases and found it capable of doing everything I wanted it to. If you are looking for a flexible data base manager, seriously consider The Producer. ■

CompuServe on $5 a Day

by Gary A. Shade

How to Get the Most Out of CompuServe is one of the best books I've read. It not only covers the major areas of this mammoth consumer information service in an easy-to-read manner, but it can save you money. It was a pleasure to read, which is something I can't say about most computer-related books.

Information services like CompuServe are the fastest-growing segments of the telecommunications industry. All services of this type charge for the amount of time you go on-line. Many people (myself included) have found this to be costly, especially when you're first learning to use the system. Charles Bowen and David Peyton recognized the shortcomings of CompuServe's documentation. This book will eliminate your fear and apprehension, and ultimately save you money in connect charges.

The 277-page book is arranged in lessons, each dealing with a different

How to Get the Most Out of CompuServe

★★★★★

Charles Bowen and David Peyton
Bantam Books Inc.
666 Fifth Ave.
New York, NY 10103
277 pp. spiral bound
$12.95

Well written?	★★★★★
Organized?	★★★★★
Thorough?	★★★★★
Readable?	★★★★★

aspect of the system. You start with the basics: logging on, changing your password, and configuring your system. Each excursion you make on CompuServe builds on what you learned in the previous lesson, giving you the knowledge and confidence to explore further.

The authors state that using the lessons will amount to approximately $36 worth of on-line time, according to current CompuServe rates. But they demonstrate so many shortcuts and advanced features that the charges seem trivial. Even if you plan on only being an occasional user, the $36 is still a wise investment.

Bowen and Peyton discuss and access every phase of CompuServe within the lessons, from retrieving the weather forecast for your area to using the bland but popular CB simulator. The lessons include the commands necessary to efficiently use each service. Many of the tips have already cut my on-line time and costs.

Full Coverage

Both authors have been sysops (system operators) for various CompuServe special-interest groups (SIGs), so they know their way around the system. They even take you on a tour of The Good Earth SIG (HOM-145), of which Dave Peyton is the sysop.

A trip to the new Compu-U-Store is one of the topics discussed in Chapter 14. A consumer can search the store for an item (over 50,000 are stocked) and order it through this service, which offers goods at a discount. While there is a membership fee for using the Compu-U-Store, the authors walk you through by using the free demonstration.

Other topics and lessons include banking by computer, the electronic mall, financial services, and how to buy software through CompuServe. The reader is shown how to prepare an electronic message through e-mail (electronic mail) and how to post it. Similarly, the authors show how to leave a message on the national bulletin board, which all CompuServe subscribers can access and read.

Advanced lessons demonstrate how to turn off the CompuServe menus (saving you time and money) and how to create your own custom menu. An "On-Line Survival Kit" is included in the Appendix that summarizes the commands used in the different areas. It is arranged in a logical, easy-to-reference manner.

Conclusion

I've been a CompuServe subscriber for nearly three years and I feel comfortable with the services I normally use. Yet I still felt intimidated by the costs of learning new aspects of the network.

During one of my first experiences with CompuServe, I found myself watching screen after screen of text roll by without knowing how to stop it. I frantically searched the manual trying to find the code to terminate it. In frustration, I turned the power off to the modem, disconnecting myself. After calming down, I realized I had just thrown away a half hour of the free time given to new subscribers. I'm sure many other people have had the same type of experience.

The alternative to purchasing *How to Get the Most Out of CompuServe* would be to order the many different user manuals from CompuServe, which explain the areas within the network. But these do little to eliminate the intimidation, confusion, and frustration felt by new users.

This book will save you time and money, from log-on to sign-off. It is for anyone interested in using CompuServe more efficiently, whether a first-time user or experienced veteran. ■

Great Zeus: A Solid Model I/III/4 Editor/Assembler

by Ronald A. Cangro

When I first got Cosmopolitan Electronics' Zeus Editor/Assembler, I was a little disappointed to find that it had a line-oriented rather than a screen-oriented editor. But my apprehension dissipated when I used the system and found it to be fast, versatile, and full of features not found in other editor/assemblers.

Zeus combines the best characteristics of both a line and a screen editor. You can edit anywhere on the screen by moving the nondestructive cursor, but program changes affect the current line only, speeding correction time. And the assembler supports some impressive features: multiple expressions on a single line, a three-pass assembly operation that significantly speeds up assembly time, comment blocks at the beginning of the object code, and labels. While Zeus lacked a few things, like a macro capability, a built-in debugger, a direct assembly to memory, tape input/output, and a cross-reference facility, it's a good package for developing small- to medium-sized programs.

The Editor

The disk has its own operating system, MULTIDOS (also manufactured by Cosmopolitan Electronics), that runs on a Model I, III, or 4. When you boot up Zeus, the familiar DOS logo appears. Loading Zeus from disk automatically puts you in the edit mode.

The Zeus editor has two modes of operation: line-oriented and character-oriented. On boot-up, you're in the line-oriented mode and you invoke all commands with a single keystroke. Most of the letters correspond to the functions they perform: L loads a source file, D deletes a line, and I lets you insert code (see Table 1 for a complete list of commands).

Once you load text in the buffer, you can scroll through it; you display a full page of text (16 lines) by pressing the enter key.

Edit commands let you find specific

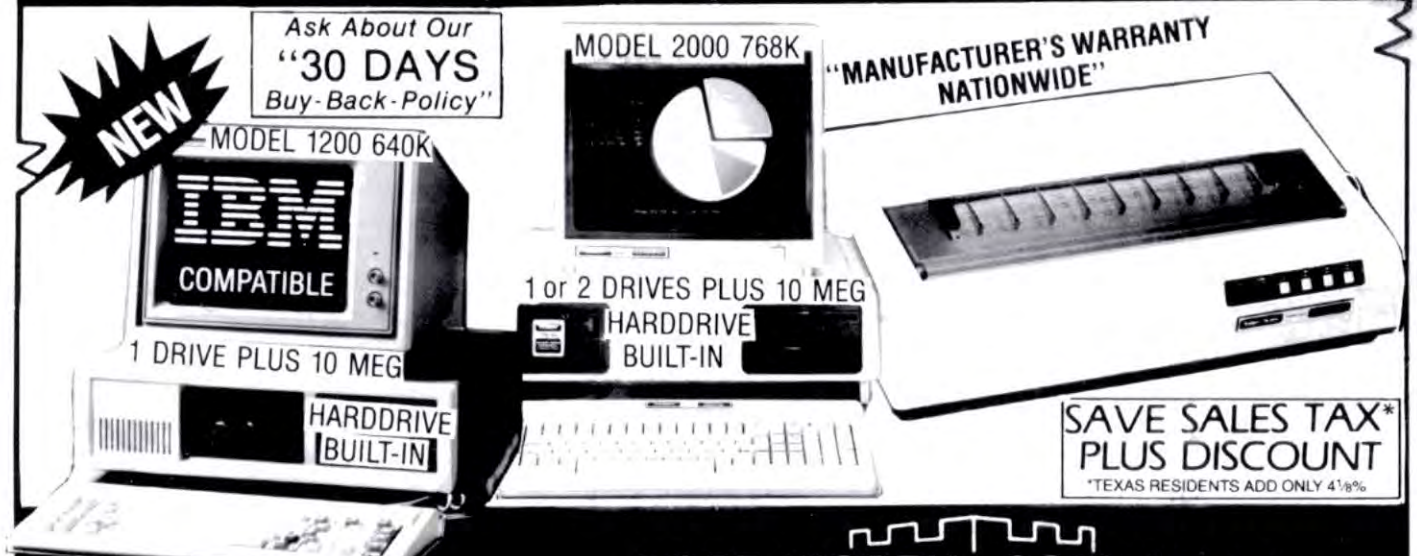
Zeus Editor/Assembler

★★★

Cosmopolitan Electronics Co.
5700 Plymouth Road
Ann Arbor, MI 48105
313-688-6660
Models I, III, and 4
32K RAM
One disk drive
$79.95

Easy to use?	★★★★☆
Good docs?	★★☆☆☆
Bug free?	★★★☆☆
Does the job?	★★★☆☆

OPTIMAL TECHNOLOGY, INC.
EP-2A-79
EPROM PROGRAMMER
POWER
PERSONALITY MODULE
PROGRAMMING SOCKET
PROGRAMMING VOLTAGE



Ask About Our
"30 DAYS
Buy-Back-Policy"
NEW
MODEL 1200 640K
IBM
COMPATIBLE
1 DRIVE PLUS 10 MEG
HARDDRIVE
BUILT-IN
MODEL 2000 768K
1 or 2 DRIVES PLUS 10 MEG
HARDDRIVE
BUILT-IN
"MANUFACTURER'S WARRANTY
NATIONWIDE"
SAVE SALES TAX*
PLUS DISCOUNT
*TEXAS RESIDENTS ADD ONLY 4⅛%

text character sequences, exact line numbers, and even lines referenced by a specific label. And you can insert text at any time with dynamic line renumbering, allowing unlimited text insertion between any two lines.

You can edit any character within a specific line by using the Edit command, which puts you in the character-oriented edit mode. This is similar to the Basic interpreter editor, with one exception: you see exactly what you're editing at all times. The nondestructive cursor moves left and right over the line, and you use simple control sequences to delete, insert, or overtype a character, or to move to the end of a line. Use either the down-arrow or clear keys as the control key.

The editor allows a global search and replace of a particular string. For example, you could change all occurrences of STRING1 to STRG1 in one operation. Single-quoted text and comments remain unaffected unless you specify that option. This is a flexible and useful command although it doesn't work on opcodes and operands. You can also move or duplicate a line (or group of lines) anywhere in the text buffer.

It's possible to display the amount of memory the source text, symbol table, and Get file buffers use, as well as how much memory you have left. This is helpful in gauging where to split your source text.

Although you probably won't realize it until it's too late, Zeus's most useful editing command is X. You use it to recover source text when restarting Zeus. You only have to crash your computer once to appreciate it.

The editor also checks every source line for correct syntax as you enter it. It even checks the source code that you load from an ASCII file or an EDTASM format file before putting it in memory. This saves a lot of time tracking down elusive syntax errors during assembly.

Since Zeus allows editing and assembling without destroying the source code in memory, you may think this means there isn't much room left for serious programming. But the program allows 34K for source and symbol code, and supports the Get pseudo-op that makes it possible to break your source code into modules and store them on disk. The assembler then reads the disk files in sequence. You could actually use the 34K for symbol table storage only.

One handy feature that Zeus lacks is a keyboard macro capability, which would have made it easier to enter commonly used phrases and words with a single keystroke.

The Assembler

The assembler has a rich assortment of features. Table 2 lists the available pseudo-ops. It supports the common Intel and Zilog pseudo-ops, including multiple expressions on the same line with the DB, DW, and equivalent pseudo-ops. This feature alone makes it worth the price.

It handles conditional assemblies using the If...End...If clause and an ERR error switch. Zeus also provides an assortment of subcommands to control assembly listing and printout, error-handling, object-code generation, and symbol tables. Table 3 shows the assembler subcommands available.

While the assembly is in progress, you can pause and restart the listing. It uses 16-bit operations while evaluating expressions. It does multiplication, division, addition, subtraction, and modulo arithmetic as well as the logical Or, And, left and right shift, and exclusive Or in label evaluation.

One of Zeus's unique features is its three-pass operation. During the first pass, the assembler creates the symbol table and defines the values of the DS, End, EQU, and ORG pseudo-ops. After this phase, you can reference all the lines by label names. During the second pass, the assembler does the actual assembly and directs output to the screen and object file. It then sets a flag. All the additional assemblies with this flag set will proceed with pass three; an instant assembly on all processed source code in the buffer. This definitely speeds up development time.

Command	Description
A	Assemble source code
B	Command mode calculator
C	Global change
D	Delete line(s) from the text buffer
E	Edit specific text line
F	Print line with the specified label
G	Printer format command
H	LPRINT line(s)
I	Insert line(s)
J	LPRINT raw data
K	Kill file
L	Load file
M	Move/duplicate text line(s)
N	Reset (new) text buffer
O	Opcode/operand Reference
P	Print line(s)
Q	Quit/exit
R	Reference
S	Save file
T	Print sorted label table
U	Memory used
X	Recover source text on reentry

Table 1. Zeus editor/assembler commands.

Pseudo-op	Description
COMM	Comment
DEFB or DB	Define byte
DEFL or DL	Define label
DEFM or DM	Define message
DEFS or DS	Define space
DEFW or DW	Define word
End	Terminate assembly
ENDIF	Delimit conditional assembly
ERR	Abort assembly
EQU	Equate
Get	Include source from disk file
If	Begin conditional assembly
List	Control printing
ORG	Origin
Page	Eject to top-of-form
SBTL	Subtitle
TITL	Title

Table 2. Assembler pseudo-ops.

Tidbit[#] 17

I used to wonder why MULTIDOS disks provide only 75K RAM of free memory in double-density. That was before I realized that I could get 349.5K of memory as opposed to a skimpy 75K with the quick and easy method described here.

First, at the track count prompt, invoke the back-up command using drive zero as the source drive. If you've previously formatted the disk, type in Y to format the 96 tracks; otherwise, the computer does this automatically.

The duplicator program verifies sectors and tracks up to track 35, where it stops and prompts you to press the enter key, responding with the "Data record not found during read" message. Ignore this and type in backup again. Now repeat the procedure described above. This time, however, don't reformat the disk. By not reformatting, you're able to copy the rest of the disk by skipping the verification process.

Now follow the prompts, and when finished, press the enter key, making sure that you have a MULTIDOS system disk in drive zero. To check your free memory, call up a directory of the original disk. You should have a disk with 96 tracks and 349.5K of free memory.

Barry LaLone
Lake Orion, MI

See List of Advertisers on Page 145

Another unique feature is the COMM pseudo-op. This generates a comment block at the beginning of the object file. It lets you identify your object code by listing the first portion or by using a disk-zap utility.

Additional Assembler Features

Other features abound. The program has a built-in calculator that lets you convert from decimal to hexadecimal and back again. You can access this from the edit mode.

Many assemblers' weak point is in printing out the source listing. Zeus includes a configuration program; in addition to modifying object code to work with different disk operating systems, the ZCON program sets up printer defaults. You can specify the number of printed lines per page, number of characters per line, total number of lines per page, and whether to use a form-feed or line-feed top-of-form. You can output raw data to the printer at any time with the J command. This lets you send special printer codes even when you're in the middle of an edit session.

Labels have also been a sore spot in other assembler packages. Zeus allows every character and character case to be significant. This amount of flexibility makes the code almost self-documenting.

The disk operating system deserves some mention. You receive a mini-copy of MULTIDOS called *ZEUS* supplied on a single-density disk. It has the capability of operating in double-density mode with automatic hardware recognition and automatic density recognition. Table 3 shows the library commands that are available in *ZEUS*.

Command	Description
Append	Add one file to the end of another
Auto	Automatic, invincible execution of a command or executable file on power-up
Boot	Reboot the system
Clear	Clear memory from 5200 to TOPMEM
Clock	Enable/disable screen clock
CONFIG	Set default drive attributes on power-up (stepping speed, density, and number of sides)
Date	Set date
DIR	Print disk directory (5 options)
Dump	Transfer memory to disk file
Forms	Set printout parameters
Free	List available disk file space
Kill	Remove a file from the disk
LIB	Print a list of the library commands
List	List a file on the video screen
Load	Load an object file from the disk to RAM
Patch	Modify a disk sector
Print	List a file to the printer
Rename	Change a file name
RESTOR	Recover a killed file
SETCOM	Set RS-232 parameters (Model III only)
SETDIR	Alter directory address marks (Model I only)
Skip	Read a 40-track disk in an 80-track drive
Time	Set time
TOPMEM	Set upper memory limit
Verify	Reread disk sectors for errors
$	Duplicate a single file (similar to Copy)

*Table 3. *ZEUS* library commands.*

Documentation

If there's one area where Zeus could use some improvement, it's in the documentation. The problem is more of quantity than quality. What is documented is done fairly well. There's a section detailing each command operation, a fairly large part on the pseudo-ops, and a brief section explaining the creation of system disks and backup copies.

The problem is the lack of a tutorial to show you how to load the editor, enter a simple program, then assemble and list it. As it is, a user is on his own and has to sort through the command examples to understand all of Zeus's features. I'm still not sure how to exit the program back to DOS without resetting the computer.

An index would also have been a useful addition, especially if the documentation were expanded. Also missing is a description of all the Z80 opcodes. The assumption is that the user is familiar with Assembly language and requires a fast, inexpensive, easy-to use-assembler.

Conclusion

Zeus is a system that should be used by people who require a powerful package to develop smaller programs. But for people who do a lot of sophisticated programming, it's limited without a macro capability, a built-in debugger, a direct assembly to memory, tape input/output, or a cross-reference facility.

On the positive side, it doesn't require disk overlays. The editor is very capable and as easy to use as a full-screen editor. Best of all, it isn't necessary to transfer it to another operating system because *ZEUS* is powerful enough all by itself.

Zeus is a logical progression from the popular EDTASM program. If EDTASM satisfies your programming needs, you may want to check out Zeus. ■

This is TRSDOS 6.2

by Hardin Brothers

Logical Systems Inc. has made microcomputer history by publishing the (almost) complete source code for TRSDOS 6.2 in three volumes. To my knowledge, this is the first time that the source code for a new and living operating system has been published for distribution. Of course, these books aren't inexpensive; at $99 apiece, they may have established a record as the most expensive paperback books ever published.

Inside the Books

For your $99, you get an 8½- by 11-inch book filled with commented TRSDOS source code. Volume 1, *The System*, contains the source code for Boot/SYS (or Lowcore), SYS0/SYS (also called Sysres, since it always re-

mains resident in memory), SYS1 through SYS5 and SYS9 through SYS13.

Volume 2, *The Libraries*, contains the source code for the library modules SYS6 through SYS8. The final volume of the series, *The Utilities*, includes the source code for almost all the TRSDOS 6.2 utility programs.

The three books contain almost 1,100 pages of code, enough to keep even the most dedicated hacker reading and studying for weeks (if not months). LSI prefaces each program with a short description of the functions it performs. The comments throughout the books are sufficient to give an experienced Assembly-language programmer all the information necessary to understand the programs' inner workings (see the sidebar for examples from *The Source*).

Who Needs These Books ?

Why would anyone want such a set of books? Radio Shack has exorcised most of the Model 4 TRSDOS bugs. And it's unlikely that anyone will use them to type in all 1,100 pages of code, since TRSDOS is distributed free with the Model 4/4P.

I think there are four important and likely uses for these books. First, some people (including me) just like to know how things work. How does TRSDOS handle the video screen? How does it parse commands? What happens when you install Memdisk? If you want to know as much as possible about your Model 4, these books (along with Radio Shack's *Model 4 Technical Reference Manual*) will provide all the answers a software nut could want.

Also, you may want to make modifications to your personal copies of TRSDOS 6.2. Perhaps you'd like to reinstate the Kill command, in addition to Remove, from DOS Ready.

The Source:TRSDOS/ LS-DOS 6.2

★★★★½

Logical Systems Inc.
8907 North 55th St.
P.O. Box 23956
Milwaukee, WI 53223
800-248-3535
Volume 1-The System $99
Volume 2-The Libraries $99
Volume 3-The Utilities $99
$249 for all three

Well written?	★★★★☆
Organized?	★★★★★
Thorough?	★★★★☆
Readable?	★★★★★

Notes from *The Source*

by Hardin Brothers

Below are some examples of technical information culled from *The Source*. Be sure to make any DOS changes to a back-up copy of TRSDOS 6.2—not an earlier version—and to test the new code thoroughly before moving it to your working copies of the system disk.

- Generally, when you reboot your computer, programs in memory are left untouched. However, TRSDOS 6.2 uses memory locations 4300–43FF hexadecimal for the boot routine, so any program in that area will be destroyed.
- If you want to change the definition of the function keys temporarily while in Basic, the lookup table with their values is stored from 0918–091D hexadecimal. The values are stored in this order: F1, shift F1, F2, shift F2, F3, shift F3. You can POKE new values into those bytes, but the computer will react improperly if you try to define them as break, clear, or pause. Be sure to reset the values (or reboot the computer) before using any program that expects the F keys to have their original values.
- You need change only a single byte in SYS1/SYS to activate the Kill command in addition to Remove. LIB will also add Kill to the list of commands it displays if you add this patch:

```
PATCH SYS1/SYS.LSIDOS (X'2054'= "K")
```

- The @DATE SVC puts a formatted version of the system date into a buffer as MM/DD/YY. But it also returns with DE pointing to DOS's date storage area (DATE$+0) at 0033H. The storage scheme is:

```
DATE$+0 Year (80-87)
DATE$+1 Day of month (1-31)
DATE$+2 Month (1-12)
DATE$+3 Bits 0-7 of day of year
DATE$+4 Bit 0=bit 8 of day of year
        Bits 1-3 contain day of week
        Bit 7 set if leap year.
```

You could find the day of the week in Basic with this code:

```
D% = ((PEEK(&H0037) OR 14)/2) - 1
D$ = "SunMonTueWedThuFriSat"
DAY$ = MID$(D$,D%*3+1,3)
```

- A 2-byte patch space in the Click/FLT makes the audible click operate on a specific key or set of keys. The current code is:

```
2474 0000 1060  DW  00
2476 201C 1070  JR  NZ,LEAVE
```

Also, at 247A hexadecimal, the computer loads the D register with the click tone value and the E register with the click duration value. One interesting change to the Click/FLT is the following:

```
.Patch to CLICK/FLT.FILTER to make
. a beep only when <ENTER> is pressed.
.
.First, test for <ENTER>
X'2474' = FE 0D
.
.Set the duration
X'247B' = 80
.
.Set the tone
X'247C' = FF
.
.End of patch.
```

- I dislike the Device command default parameters. I'd much rather have the D parameter default to off and the B parameter default to on. This patch will accomplish that:

```
.Patch to change DEVICE defaults
.Apply to SYS6/SYS.LSIDOS
.
.Position to DEVICE Library command
L61
.
.Set D parameter to default D = N
X'241E' = 00 00
.
.Set B parameter to default B = YES
X'2580' = FF FF
.
.End of patch
```

The necessary bytes are sitting in SYS1/SYS's command interpreter just waiting for you to change a NOP byte (00H) to a 'K'. Perhaps you want to add your own supervisory calls (SVCs) or to determine the side effects of some of the present calls.

You could also use the books to create your own utility programs that take full advantage of what's already available with TRSDOS 6.2. I've experimented recently with a split-screen terminal program for CompuServe's CB simulator and special-interest group conference areas. I have the program working as a dumb terminal, but I'd like to add a capture buffer and other enhancements found in the TRSDOS's COMM program. After studying COMM's source code, I've found ways to add the appropriate patches. I won't have to spend hours duplicating work that has already been done.

Probably the most important purpose of the source code is to give software developers all the help they need in producing packages for the Model 4. The inner workings of the Model I/III ROMs have been published for several years, and have helped software writers considerably. A similar knowledge of the Model 4's operating system may (hopefully) lead to more interesting and complex programs for the Model 4.

What's Missing

While these books are nearly complete, they do lack a few things. Whenever the code accesses the Model 4's hardware directly through a port, the source lines have been omitted but the object code is still present. I assume these omissions are made to respect Tandy's proprietary secrets and also to mark those sections of code that are specifically Model 4-oriented (the same code is essentially used for Model II, 12, and 16 computers, and LSI hopes that other manufacturers of Z80 computers will begin using TRSDOS 6.2 as well).

Also, the source code for the Help utility is missing from Volume 3. This last omission is unfortunate, because the Help command can be expanded to cover topics in addition to DOS commands. As of this writing, neither LSI nor Tandy has explained how to do so (I will, however, discuss how to add new Help files in a future installment of The Next Step).

Missing from these books, of course, is the Basic source code (owned by Microsoft) as well as MODELA/III, probably for similar reasons. But I didn't expect to see either of those.

These books are an important addition to my library. It's one thing to be able to use TRSDOS 6.2 well: knowing the source code gives you more insight into how programs interact with the system. Whether you should buy one or all of these books depends on your level of sophistication and the complexity of the programs you need to write. The wealth of information in these books isn't difficult to find or understand, but their cost may deter many hackers and hobbyists from buying them. ■

Listing continued from p. 130

```
7067 E5                 PUSH   HL            ; THE STACK
7068 D5                 PUSH   DE
7069 3E10               LD     A,10H         ;WRITE BYTE TO ENABLE
706B D3EC               OUT    (TRSEXE),A    ; EXTERNAL I/O ACCESS
706D DBE0               IN     A,(TRSRDI)    ;CHECK IF EXT. I/O INT.
706F CB5F               BIT    3,A           ;I/O BUS INT.?
7071 201A               JR     NZ,NOTHER     ;IF NOT LOW, NOT HERE
7073 3E0C               LD     A,0CH         ;GET OCW2 WORD FOR 8259A
7075 D328               OUT    (PICP0),A     ;CHECK FOR 8259A INT.
7077 DB28               IN     A,(PICP0)     ;GET 8259A RESPONSE BYTE
7079 CB7F               BIT    7,A           ;INTERRUPT GENERATED?
707B 2810               JR     Z,NOTHER      ;IF ZERO, NOT BY 8259A
707D CB27               SLA    A             ;ELSE, DOUBLE VALUE TO
                                             ; GET TABLE OFFSET
707F E60F               AND    0FH           ;CLEAR UPPER BITS
7081 210070             LD     HL,INTTBL     ;POINT TO TABLE BEG.
7084 5F                 LD     E,A           ;PUT OFFSET INTO E REG.
7085 1600               LD     D,0           ;CLEAR UPPER OFFSET BYTE
7087 19                 ADD    HL,DE         ;POINT TO INT. VECTOR
7088 5E                 LD     E,(HL)        ;GET LOW ADDR. BYTE
7089 23                 INC    HL            ;INCREMENT TABLE PTR.
708A 56                 LD     D,(HL)        ;GET HI ADDR. BYTE
708B EB                 EX     DE,HL         ;PUT ADDR. INTO HL
708C E9                 JP     (HL)          ;JUMP TO THE SERVICE ROUT
                  ;
708D 21BC70       NOTHER LD    HL,RESTOR     ;GET RETURN ADDRESS OF ISR
7090 E5                 PUSH   HL            ; AND PUT IT ON THE STACK
7091 2A1270             LD     HL,(VECTOR)   ;GET ISR ADDRESS
7094 E9                 JP     (HL)          ;CALL THE SUBROUTINE
7095 1825               JR     RESTOR        ;RESTORE STUFF AND RETURN
                  ;
7097 3A1170       TIMER0 LD    A,(TIMCNT)    ;GET INT. COUNTER
709A 3C                 INC    A             ;INC. IT FOR THIS ISR
709B 321170             LD     (TIMCNT),A    ;SAVE IT AGN IN MEM.
709E FE04               CP     4             ;HAVE FOUR INTS. OCCRD?
70A0 200E               JR     NZ,DONE       ;IF NOT, FINISH UP
70A2 AF                 XOR    A             ;ELSE, CLEAR COUNT
70A3 321170             LD     (TIMCNT),A    ; IN MEMORY.
70A6 3A1070             LD     A,(PPIBYT)    ;GET 8255A PORT B BYTE
70A9 C610               ADD    A,10H         ;INCREMENT LED DISPLAY
70AB 321070             LD     (PPIBYT),A    ;SAVE BACK IN MEM.
70AE D321               OUT    (PPIB),A      ;WRITE TO 8255A PORT B
70B0 3E24         DONE   LD    A,24H         ;GET LOW TIMER COUNT BYTE
70B2 D324               OUT    (PITT0),A     ;WRITE TO TIMER 0
70B4 3EF4               LD     A,0F4H        ;GET HIGH COUNT BYTE
70B6 D324               OUT    (PITT0),A     ;WRITE TO TIMER 0
70B8 3E20         BADINT LD    A,20H         ;GET EOI (OCW2) CMD
70BA D328               OUT    (PICP0),A     ;SEND TO 8259A
70BC 3E08         RESTOR LD    A.08H         ;WRITE BYTE TO ENABLE
70BE D3E0               OUT    (TRSINE),A    ; EXTERNAL I/O INTS.
70C0 D1                 POP    DE            ;RESTORE REGISTERS
70C1 E1                 POP    HL            ; FROM STACK
70C2 F1                 POP    AF
70C3 FB                 EI                   ;RE-ENABLE INTERRUPTS
70C4 C9           NOPROC RET                 ;DONE - RETURN FROM INT.
7014                    END    INTRPT
00000 TOTAL ERRORS
25366  TEXT AREA BYTES LEFT
```

End

Continued from p. 130

one second. Using the 250 KHz clock, the count value of 0F424H results in a 0.25 second interrupt. Thus, four interrupts will make up one second.

The last part of the initializing section requires that we provide for other system interrupts. If an interrupt occurs and we determine that the 8259A is not interrupting the CPU, we jump to the location otherwise used by the system.

The interrupt service routine first sends a 0CH (OCW3) to the 8259A, telling the 8259A that we are using polled mode. The 8259A then sends back a value which tells the interrupt status of the controller (see Fig. 12). If the high bit (bit 7) is set, an interrupt is being requested. Also, if an interrupt is being requested, the highest priority interrupt is given by the lower three bits of the returned byte. This can be used as an offset into a table of ISR addresses (by first shifting it left one bit).

If the ISR determines that the 8259A did not generate the interrupt, service is passed to the normal TRS-80 ISR. Otherwise, it adds one to its count variable. If four interrupts have not yet occurred, it finishes by restarting the 8253 timer for another 0.25 second interrupt, and issuing a nonspecific end-of-interrupt (EOI) to the 8259A via OCW2. This tells the 8259A that servicing is complete for the interrupt just required, and to arbitrate the next lower priority interrupt. Interrupts are then re-enabled, and a return from interrupt is executed.

If the interrupt was found to occur from the 8259A, the count variable is cleared (to count four more interrupts) and the LED display is updated by 1. The service routine then finishes by restarting the 8253 timer and sending the EOI to the 8259A.

Conclusion

The 8259A interrupt controller provides more capability and flexibility than described here. If you are interested in some of the more advanced applications of the device, you should get a copy of the data sheet from Intel Corporation. Since the G.P. I/O board project also comes in CMOS (which may be important to some of you), the 8259A is available in CMOS from the same sources. ■

1983 Intel Microprocessor and Peripheral Handbook
Intel Corporation
Literature Department
3065 Bowers Avenue
Santa Clara, CA 95051

Zilog 1981 Data Book
Zilog, Inc.
10340 Bubb Road
Cupertino, CA 95014

1982/83 Mostek Z80 Designer's Guide
Mostek Corporation
1215 West Crosby Rd.
Carrollton, Texas 75006

Write to Roger C. Alford at Washtenaw Digital Systems, P.O. Box 2014, Ann Arbor, MI 48106.

Quantity	Description	Distributor	Part Number	Price
1	8259A Programmable interrupt controller IC	JDR	8259	6.90
1	7416 Hex inverter/drivers (O.C.)(TTL) IC	JDR	7416	.25
1	74LS04 Hex inverter (LS TTL) IC*	JDR	74LS04	.24
1	74LS138 3-to-8 decoder IC*	JDR	74LS138	.55
1	74LS32 Quad 2-input OR gate (LS TTL) IC†	JDR	74LS32	.29
1	100 μF/35 V electrolytic capacitor (PC mount)*	RS	272-1028	.79
1	.1 μF/50 V disc capacitor*	RS	272-135	.25
6	1k ohm resistor (¼ watt)‡	RS	271-1321	.08
1	74LS74 Dual D-type flip-flop(LS TTL) IC§	JDR	74LS74	.35
1	8 pos. terminal block*	RS	274-687	1.89
1	2 pos. terminal block*	RS	274-656	.60
1	.1" matrix grid prototype board	RS	276-158	1.95
1	40 pos. cable header (W/W)Ω	DK	R241-ND	5.58
1	40 pos. ribbon cable edge connector Ω	DK	R503-ND	3.80
1	40 pos. ribbon cable socket connector	DK	R306-ND	3.73
1 ft.	40 cond. ribbon cable Ω	DK	R007-ND	0.00
1	50 pos. cable header (W/W)π	DK	R247-ND	6.93
1	50 pos. ribbon cable edge connector π	RS	276-1566	4.95
1	50 pos. ribbon cable socket connection π	DK	R307-ND	4.65
1 ft.	50 cond. ribbon cable π	DK	R008-ND	0.00

* Stand-alone board only.
† Reset circuitry use and Model III/4 required.
‡ One less required for general-purpose I/O board addition. Two less required if reset circuitry not implemented.
§ Use with reset circuitry only.
Ω Model I only.
π Models III/4 only.

Addresses:

JDR Microdevices, 1224 S. Bascom Ave., San Jose, CA 95128, 800-538-500 or 408-995-5430 outside California; 800-662-6279 within California.

Radio Shack (RS) National Parts Division, 900 East Northside Drive, Fort Worth, TX 76102, 817-870-5662.

Digi-Key Corp. (DK), Highway 32 S., P.O. Box 677, Thief River Falls, MN 56701, 800-346-5144 or 218-681-6674.

Table 1. Parts list and ordering information.

✓ See List of Advertisers on Page 145

Fast Impressions

The Fastext-80 bidirectional dot-matrix printer ($259) gives you 80 characters per second (cps) print speed at 10 characters per inch (cpi), and features a line buffer and six-pitch capability.

Fastext-80 also features a self-test switch, an 80-character print line, emphasized or elongated print, and bit-image graphics.

Pitches include pica, elite, condensed, and enlarged printing. The printer supports the 96-character ASCII set and can print in seven languages.

Its drop-in cassette ribbon has a life of one million characters. Fastext-80 comes with friction feed and a Centronics parallel interface. A tractor feed is also available.

For more information contact Smith-Corona at 65 Locust Ave., New Canaan, CT 06840, 203-972-1471.

Reader Service ✓ 563

The Fastext 80 dot-matrix printer prints seven languages in six type fonts.

Running with IBM

Xcalibur, from Micro Projects Engineering Inc. (3951 Higuera St., Suite B, Culver City, CA 90230, 213-202-1865), lets you run IBM PC software on the Models I, III, 4, and 4P.

This enhancement board gives you PC compatibility to run MS-DOS 2.11, Multimate, dBase II, WordStar, MultiPlan, and other popular IBM PC software packages. Add a monochrome monitor card ($199.95) and 128K RAM ($198) and you can run Lotus 1-2-3.

Xcalibur comes with an IBM-compatible bus connector for use with IBM expansion cards and MS-DOS. It installs easily through cable connections inside your computer.

The Xcalibur board is $495, and includes an MS-DOS disk. Additional RAM chips and other upgrade hardware are also available.

Reader Service ✓ 552

Adventures In Pixieland

Pixie Quest, from Seal Software (4 White Birch Court, Turnersville, NJ 08012, 609-228-6169), is a Model I adventure game that leads you and your companions on a trip through the forbidden wood of Balmar and the caves of the evil sorcerer, Tralon.

Your goal is to return pixies Frolie and Pixel to the village of Brahlee. Along the way you'll find treasures, puzzles, and meet jabowockies, ogres, and evil gremlins.

The game's parallel paths through the woods mean that failure to solve a puzzle won't leave you at a dead end. Ultimately, however, you must cope with every predicament using ingenuity, imagination, and whatever you find on your journey.

Pixie Quest ($34.95) requires 48K RAM and one drive.

Reader Service ✓ 553

Programming for Fun

When you're tired of working with your TRS-80, *101 Programming Surprises & Tricks for Your TRS-80 Computer* ($10.95) gives you games, novelties, and techniques to make computing fun again.

This strictly-for-fun programming guide includes music and sound programs, animation programs, one- and two-player games, and programming tricks useful for more serious programming applications.

If typing in listings cramps your style, a disk that includes all the programs is available for $19.95.

For more information contact Tab Books Inc., Blue Ridge Summit, PA 17214, 717-794-2191.

Reader Service ✓ 554

Model 100 Means Business

Business Finance ($39.95) is Tandy Corp.'s 12-program financial analysis package for the Model 100. Calculation programs calculate net present value, internal rate of return, financial management rate of return, and depreciation.

Also included are analysis programs for rent or purchase decisions, lease/purchase analysis, machinery replacement, and bond analysis.

An annuity/annuity-due program does calculations for savings accounts, compound interest, annuities, leases, loans, financial modeling, and other investments.

The portfolio analysis program generates a series of portfolio distributions to minimize risk for a given rate of return or maximize return for a given risk.

Business Finance requires 16K RAM and is available from Radio Shack stores.

Why the user-friendly DOSPLUS IV is looking friendlier than ever.

DOSPLUS IV		TRS * DOS	
FEATURE	COST	FEATURE	COST
File Editing and disk editing	included	Utilities Package	40.00
M-ZAL (Machine Language Development System): Text editor, linker, assembler	included	ALDS	199.00
Built-in speed	included	TRS * DOS 6.2	40.00
Basic Programming enhancement	included	not available	*
Directory Verification repair	included	not available	*
Disc mapping	included	not available	*
Total Cost	**$169.95**	Total Cost	**$279.00**

Good friends give you good deals on great products. Just like Micro-Systems Software has with its unique DOSPLUS IV program. It's the friendliest, most powerful *and* most complete DOS ever to operate a TRS-80.

In fact, if you wanted all the benefits DOSPLUS IV offers for its low, low price of $169.95 (including the incredible M-ZAL) you'd have to pay over $100 more. And still wouldn't find *all* the features that have been programmed into DOSPLUS IV.

So, check the comparison chart above. Then check out DOSPLUS IV for yourself. You'll discover you get much more program for much less money.

And, after all, isn't that what friends are for?

DOSPLUS® IV ✓181

Another fine software package from Micro-Systems Software, Inc.

CALL 1-800-327-8724 FOR THE DEALER IN YOUR AREA.

For more information contact Tandy Corp./Radio Shack, 1800 One Tandy Center, Fort Worth, TX 76102.

Reader Service ✔ 556

Computer Card Game

Casino, the card game, is now a computer game. You play against your Model I, III, or 4, as it keeps score, shuffles the cards, and plays dealer.

The game includes an instruction manual with sections on rules of the game and playing strategy. Casino is available on 16K cassette ($12.95) or 32K disk ($16.95). Contact Dynacomp (Catalog Dept., P.O. Box 18129, Rochester, NY 14618, 716-442-8960) for more details.

Reader Service ✔ 557

Workstation 100 makes using your Model 100 easier.

Portable Accessories

The Workstation 100 kit ($45.95) contains six attachments (also available separately) to make your Model 100 more efficient.

The Power-100 ($29.95) is a rechargeable battery pack that powers the Model 100 for 9–12 hours. Support legs ($3.95) angle the Model 100 for easier typing.

The kit also includes a clip-folder typing stand ($7.55), a pen ($2.75), a mechanical pencil ($1.75), and a battery case ($3.95). Attachments are compact enough to fit into the Model 100 carrying case.

Contact A.R.M.S. (12131 Old Buckingham Road, Midlothian, VA 23113, 804-794-6675) for more information.

Reader Service ✔ 555

The Tax Preparer Cometh

The Tax Commissioner income tax preparation software ($895) gives attorneys, accountants, and tax preparers a fast and efficient alternative for generating client tax returns.

It prepares returns during a one-stop interview with

See List of Advertisers on Page 145

clients, and eliminates transposition of information and errors that frequently accompany this step in tax preparation.

The program displays each tax form. You enter the data, and Tax Commissioner computes and prints out the tax return. You can print data only, or choose from more than 35 preprinted forms.

Tax Commissioner requires a Model 2000, or a Model III/4 with CP/M 2.2. Contact TAXbyte Inc., 3803 28th St., Moline, IL 61265, 309-764-7245 for more information.

Reader Service ✓ 567

Soho Corp.'s stackable disk boxes are $4 each.

Super for the Model 4

Super, the data base management system from the Institute for Scientific Analysis Inc., is available for the Model 4. Super 2.3 is menu-driven, and features on-screen editing, a calculator program, and a filing system.

The program lets you set up and create data bases, save report formats, print reports, and sort information. Super 2.3 is available for the Models I, II, and III. It costs $199 and includes a manual. The program is also available as an upgrade.

For more information contact Institute for Scientific Analysis Inc. at 36 E. Baltimore Pike, Suite 106A, Media, PA 19063.

Reader Service ✓ 564

Disk Boxing

The D-60 compact disk storage box ($4) from Soho Corp. stores up to 60 5 ¼-inch disks.

It features a smoke-tint see-through cover and has four plastic dividers for organizing disks. Disk boxes stack for easy storage. For more information contact Soho Corp. at P.O. Box 20081, Ferndale, MI 48220, 313-543-5209.

Reader Service ✓ 559

Graphics Answers

The Inovion Personal Graphics System (PGS) is a 78K RAM stand-alone graphics generator that creates up to 2.1 million different colors.

PGS runs with any computer with an RS-232C port and features a 512- by 480-pixel display, 19-inch monitor, and an icon menu driven by a mouse or trackball.

A frame grabber/digitizer lets you freeze a signal from your TV, VCR, or video camera and manipulate the picture.

Synthetic color routines change the color saturation or hue of a group of pixels. PGS retains textures and three-dimensional quality while enhancing colors.

PGS is $3,495 from Inovion Corp., 195 E. Gentile St., Layton, UT 84101, 801-546-2850.

Reader Service ✓ 566

Cassette and disk labels from Quick Label Systems are re-markable.

Computer Calling

You're sitting at home, enjoying a cool glass of Jose Cuervo tequila, when the phone rings. You pick it up, hear a high-pitched tone, and turn to your TRS-80. "It's for you," you say, handing the telephone to your computer.

The scenario sounds futuristic, but the technology is just a phone call away. William J. Cook's *The Joy of Computer Communication* ($5.95) shows you how to access the world of computer telecommunications to get stock market quotes, make business deals, send electronic mail, and do your shopping.

The book also includes instructions for accessing the Dow Jones News/Retrieval Service, banking at home, subscribing to an electronic version of *The Washington Post*, and accessing other information services.

For more information contact Dell Publishing Co. Inc., 1 Dag Hammarskjold Plaza, 245 E. 47th St., New York, NY 10017.

Reader Service ✓ 560

Label Mark-ups

The re-markable labeling system from Quick Label Systems (437-B Main St., Laurel, MD 20707, 301-725-0050) lets you erase and reuse disk and cassette labels.

Labels include clear plastic overlays that you can wipe clean and re-mark an unlimited number of times. The kit includes 24 labels, the marking pen, and instructions for $4.95.

Reader Service ✓ 551

✓ See List of Advertisers on Page 145

NEW PRODUCTS

Fujitsu America's compact DPMG9 dot-matrix printer runs at 180 cps.

Fujitsu Ink

The Model DPMG9 compact dot-matrix printer ($695) from Fujitsu America Inc. boasts 80-column, bidirectional printing with 180 cps draft-quality printing (25 cps correspondence-quality).

The printer features friction and tractor feeds, automatic perforation skip, 255 character fonts, and a language-select function. It prints international characters, superscripts, subscripts, and images.

The DPMG9 weighs 19 lbs. and measures 12 by 16 by 4 inches. A Centronics interface is standard, with RS-232 interfacing available.

For more information contact Fujitsu America Inc. at 3055 Orchard Drive, San Jose, CA 95134, 408-946-8777.

Reader Service ✓ 561

Write the Volks

Volkswriter Deluxe ($295), the popular word processor for the IBM PC, is available for the Models 1200 and 2000.

Features include foreign-language and scientific character sets, on-screen tutorials, multiple fonts and colors, and compatibility with Lotus 1-2-3, VisiCalc, and other commercial spreadsheet programs.

Volkswriter Deluxe for the Model 2000 takes advantage of the Model 2000's extra memory, faster processing speed, and color capabilities.

Minimum system requirements are 128K RAM and two drives. Volkswriter is available through Radio Shack stores. Contact Lifetree Software Inc. (411 Pacific St., Monterey, CA 93940, 408-373-4718) for more information.

Reader Service ✓ 568

Tenant Fever

Good Software Corp. (12900 Preston Road, Dallas, TX 75230, 214-239-6085) has announced the Real Estate Investor III package for the Models 1000, 1200, and 2000.

Investor III ($249) uses your criteria to analyze real estate investment potential, and generates profit projections for potential investments. Projections range up to 20 years for more than a dozen financial criteria. Tax formulas are based on 1985 tax laws.

The quick-test feature generates a rough estimate of an investment's potential. Investor III also performs sensitivity analysis to evaluate the impact of

The Volksmodem 12 features control from any Hayes-compatible program, with tone or pulse dialing.

changes in expenses, taxes, appreciation, or other parameters on property investments.

Reader Service ✓ 558

It's a Volksmodem

The Volksmodem 12 300-/1,200-baud intelligent modem ($299) features auto dial/auto answer, tone and pulse dialing, busy signal detection, dial tone detection, and telephone set jack.

The unit works with any Hayes-compatible programs. It includes a voice/data switch, two modem configuration switches, status indicator lights, and five programmable set registers. A 12-volt ac adapter is also included.

Contact Anchor Automation (6913 Valjean Ave., Van Nuys, CA 91406, 818-997-7758) for more information.

Reader Service ✓ 562

Stellar Words

The WordStar word processor program, WordStar Professional Options package, and WordStar Professional package are available for the Models 1200 and 2000.

WordStar ($495) features self-help menus, block moves, global word search, justification, and formatting. Print options include boldface, underline, double strike, overprint, superscript and subscript.

WordStar Professional Options package ($345) contains three utility programs for WordStar. The MailMerge text merging utility creates form letters. CorrectStar checks spelling against three dictionary files (1,500–65,000 words).

StarIndex creates indexes, tables of contents, and lists of figures and tables automatically. Programs are also available separately.

The WordStar Professional ($695) includes WordStar, MailMerge, CorrectStar, and StarIndex. For more details, contact Tandy Corp./Radio Shack at 1800 One Tandy Center, Fort Worth, TX 76102.

Reader Service ✓ 565

LOOK!!

Low prices on hard disk drives by Percom* for TRS-80**
Model I, III & 4

10 meg-$1295.00
20 meg-$1795.00
30 meg-$2395.00

all are new and carry full factory warranty

Driver & Cable-$85.00

Call-513-268-2108

Mail certified check or money order to:

JASTE ENTERPRISES, INC.
7679 Sweet Potoato Road
Brookville, OH 45309

*TM Percom Data
**TM Tandy Corp.

244

PLAY CHESS WITH THE MASTERS!
INTRODUCING
CHESS CLASSICS

Entertaining, educational chess software to improve your chess game.

Each disk contains 64 games played by masters from Morphy to Fischer and Karpov.

Use CHESS CLASSICS to look at a great game and see if you can guess the winner's move.

Built-in scoring system gives you an approx. US Chess Federation rating for your success in picking the right move.

Available for TRS-80 Model I, III and IV
KING PAWN DISK $59.95
QUEEN PAWN DISK $59.95
BOTH DISKS $100.00

(CT Residents add 7½% sales tax)
Send check or money order to:

Noteworthy Software Inc. 127
12 Noteworthy Dr., Danbury, CT 06810

EN FLEUR CORP.

WINDOWPAD

YES WE DO WINDOWS SO CAN YOU

Windowpad an intergrated windowing and note processing utility is ready for your TRS-80 Model I III III/4 Windowpad runs along with your application program or stand alone. Just $21.25 Postpaid. Send Check or Money Order to EN Fleur Corp. 2494 Sun Valley Cir. Silver Spring MD. 20906 350

[MD. Residents Add 5% Tax]

Quit Wasting Your Time!

Use our friendly programs. They save keystrokes, speed up Basic operation, and add extra features to most Model 1/3/4™ disk systems. Try them and see for yourself (or write for information).

FAST4/CMD **$14.95**
M4™ only. Full speed, no side effects

AutoPAWS **$14.95**
M1/3/4™ Stops scrolling when CRT fills

UTILIPAK **$24.95**
Assorted utility programs for M1/3/4™

MENUGEN **$54.95**
M3/4™ Includes BASIC speed-up files

Add $3 per order packing and postage. Immediate shipment by 1st Class Mail.

the software factory

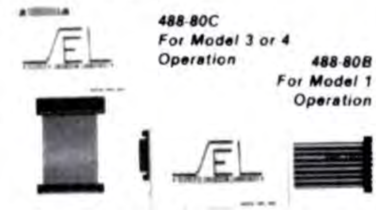
12101 N. Western View
Oklahoma City, OK 73132-9270
(405) 728-3312 104
VISA MC

IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1, 3 or 4, Level 2 or DOS with a minimum of 16K.

488-80C For Model 3 or 4 Operation

488-80B For Model 1 Operation



Model 448-80B or 488-80C Price: $425
+ 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

---

NODVILL DIET PROGRAM
$69.95

"TAKE A BYTE"©

MAINTAIN, LOSE OR GAIN WEIGHT
With the NODVILL DIET PROGRAM "Take a Byte" and a TRS-80 D.D. Disk M I/III/4 or P-C Compatible

you can quickly and accurately

- **Calculate** caloric and nutritive food intake
- **Evaluate** nutritive value of your diet
- **Compare** daily diet to individual RDA Chart
- **Create** personalized daily meals and menus
- **Plan** varied daily menus based on sound nutrition
- **Save** records of daily meals and menus for future planning
- **Print** nutrition charts, food, meal, menu and grocery lists

"**Take a Byte**" is a Modular **BASIC** Program

MAIN Program Menu
(1) Recommended Daily Dietary Allowance (RDA) CHART
(2) 733 Expandable Random Access FOOD LIST Data File
(3) 28 Nutritional MEAL LIST Data File Examples
(4) 7 Balanced DAILY MENU LIST Data File Examples
(5) GROCERY LIST Program Module
(6) 25 Page USER'S MANUAL including Charts and Tables

ALL Data Files can be EXPANDED and MODIFIED systematically and flexibly to reflect your personal diet, your choice of diet book, or your doctor's suggested diet

For more information Call (203) 431-6449
To Order: Mail a check or money order for $69.95 plus $3.00 (S & H) to:

NODVILL Software
24 Nod Road
Ridgefield, Conn. 06877
54

NEW!

"Where *is* that Movie?'

Find it quickly, easily with

VIDEO TAPE TRACKER

Takes the work out of managing your video library: easy to use • stores over 1,000 titles • prints numbered labels • lists by tape and/or title • searches by tape, title, or performer • very fast • ready-to-use

TAPE TRACKER is a real time saver and the price is right. Make your life easier, order today toll free

(800) 824-7888, oper. 422

PLEASE SPECIFY
IBM PC 64K 1 disk
TRS-80 I or III 48K, 2 disks
Tandy 2000

PROSOFT® 51
Dept. Box 560
No. Hollywood, CA 91603
(818) 764-3131

ALL THIS FOR ONLY: *$39.95*

TERMS: VISA, MC, checks, COD. Please add $2.00 shipping in U.S. In Canada $5.00 overseas. sales tax in Ca. Most orders filled within 48 hrs.

ANNOUNCING

WARRIORS AND WARLOCKS
SCENARIO: (CASTLE MYRHAVELL)
THE GREATEST ROLE-PLAYING FANTASY ADVENTURE EVER WRITTEN
FOR THE TRS-80

This game is for the strategy buff as well as the hack and slash player ages 12 and up

FEATURES
- high speed mini 3-d graphics
- parties of up to 9 characters
- characters recruit from 12 professions
- hand to hand combat
- archery weapons
- 3 classes of magic
- automatic game saving
- backup and recovery utilities
- menu driven operation
- excellent documentation
- newsletter for registered owners

WILL RUN ON
MODEL I (48K) TRSDOS NEWDOS80 DOSPLUS
MODEL III (48K): NEWDOS80 DOSPLUS
MODEL IV (in III mode)
2 drives for generation
1 drive for play
(C:64 version available - 1 *1541 drive)

PRICE $39.95 U.S. ($49.95 CAN.)
CHEQUE OR MONEY ORDER
payable to 408614 Ontario Ltd.
allow 2 weeks delivery for money order or certified cheque
6 weeks for personal cheques
DEALER INQUIRIES WELCOME

MAIL ORDERS TO:
WARRIORS AND WARLOCKS
56 ECCLESTON DR., SUITE 311
TORONTO, ONT., CAN.
M4A 1K9
(416) 759-8649 41

QUICK LABEL SYSTEMS

WRITE ON— —WIPE OFF LABEL SYSTEM

1000 Entry Changes Each Label
Label Will Outlast the Device
Permanent & Changeable Entries
Ideal for Color Coding & Indexing
Complete Kit of 24 Labels & Overlays

1 Quik-Pen & Stor-Pak . . . $4.95

Specify for:
- Floppy Disk
- VHS Video
- Cassette (Data-Audio)
- Hard Disk
- Beta Video

Write or Call:

Quick Label System
301-725-0050 117

437-B Main Street, Laurel, Md. 20707

VISA — MASTERCARD

DEALER INQUIRIES INVITIED

Able Enable

Enable ($695), an integrated package for the Model 1200, includes word processing, a spreadsheet, graphics, database management, and telecommunications.

Modules offer fast, easy transfer of data. Enable integrates data from all modules in one window for printing or data transmission. You can create graphs from the spreadsheet or data base program, then insert the graphs, spreadsheet, and data base data into a word processing document on screen.

The system is compatible with files from dBaseII, Lotus 1-2-3, VisiCalc, WordStar, EasyWriter I, and Volkswriter. Enable also lets you use modules simultaneously. For instance, you can use the spreadsheet program, print a report, and receive stock quotes over the wire all at once.

Current owners of Lotus 1-2-3 can buy Enable for $200, and dBase II users can buy it for $400, with proof of purchase. For more information contact The Software Group at Northway Ten Executive Park, Ballston Lake, NY 12019, 800-338-4646.

Reader Service ✓ 572

PFS:Planning

Software Publishing Corp. (1901 Landings Drive, Mountain View, CA 94043, 415-962-8910) offers a full line of PFS: products, including its new PFS:Plan spreadsheet program for the Models 1200 and 2000.

PFS:Plan ($140) lets you develop budgets, forecasts, and other financial models without learning complex program formulas and command sequences.

Quick entry keys let you generate row and column headings with one keystroke. PFS:Plan displays formulas next to rows and above column headings in English terms.

PFS:Plan accepts files written with Lotus 1-2-3, Multiplan, or VisiCalc, and integrates with all other PFS: series programs.

Other PFS: series programs include the PFS: Write word processor ($140), PFS:File information management program ($140), PFS:Report report generator ($125), PFS:Graph chart and graph program ($140), PFS:Proof spelling checker ($195), and PFS: Access communications program ($195). PFS:File and PFS:Report are also available for the Models III, 4, and 2000.

Reader Service ✓ 571

On-Line Health

Home Health Guide for Children ($69.95) solves the problem of deciding when to call the doctor for your child by providing information on 155 health problems.

Dr. Barton D. Schmidt, a pediatrician, describes and defines symptoms and causes of illnesses, recommends action, home care, and preventive measures. Health problems covered range from diaper rash and mumps to common colds, selecting shoes, and emergency situations.

The three-disk Model 4 package features built-in instructions and a menu for easy use. It's available at Radio Shack stores. Contact Clinical Reference Systems (P.O. Box 20308, Denver, CO 80220, 800-821-2794) for more information.

Reader Service ✓ 570

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.*

DIFFERENT TRACK

Sweetware Inc.'s newest microcomputer is getting its licks in the personal computer marketplace.

Eat-Only Memory

Sweetware Inc.'s newest microcomputer gives users a taste of high technology. It's a state-of-the art keyboard and monitor, constructed entirely of gourmet milk chocolate.

The complete system weighs in at 1.2 lbs.—truly portable. The chocolate keyboard and monitor are $16.95. A ½ lb. package of two Eat-Only disks is $11.95 (add $2 postage per item).

For more information contact Sweetware Inc., 516 Shelburne Road, S. Burlington, VT 05401, 802-862-6939.

Reader Service ✓ 569

New Products Index

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • KAYPRO™

FREE--TRS-80 Mod I,III & 4 programs supplied on DosPlus (minimum system). Complete DosPlus also available.

The VERSABUSINESS™ Series

Each VERSABUSINESS module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software**.

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.
All CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

To Order:
Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535) ~9

- add $3 for shipping in UPS areas
- add $4 for C.O.D. or non-UPS areas
- add $5 to CANADA or MEXICO
- add proper postage elsewhere

DEALER INQUIRIES WELCOME
All prices and specifications subject to change / Delivery subject to availability.

COMPUTRONICS

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977