80 micro
the #1 magazine for Tandy users

JUNE 1986
USA $4.00
CANADA $4.50
A CWC/I PUBLICATION

80 micro
6/86
#77

Advanced Basic

```
10 CLS
20 PRINT "           In This Issue          "
30 PRINT "_____"
40 PRINT "1. How to Use Boolean Operators
         by Hardin Brothers"
50 PRINT "2. File Chaining Made Easy
         by Bruce Tonkin"
60 PRINT "3. Inside the INSTR Function
         by Lewis Rosenfelder"
70 PRINT "_____"
80 PRINT "<Press Any Key to Continue>";
90 A$=INKEY$
100 IF A$="" THEN 90
110 CLS
120 PRINT "           Also Inside           "
130 PRINT "_____"
140 PRINT "1. Review: The Model 3000 Delivers"
150 PRINT "2. Add a Hard Drive to Your 1000"
160 PRINT "3. Assembly-Language on the 8088"
170 PRINT "4. Build Your Own Null Modem"
180 PRINT "5. And a Whole Lot More!"
190 PRINT "_____"
200 GOTO 200
```

06
0 74470 65947
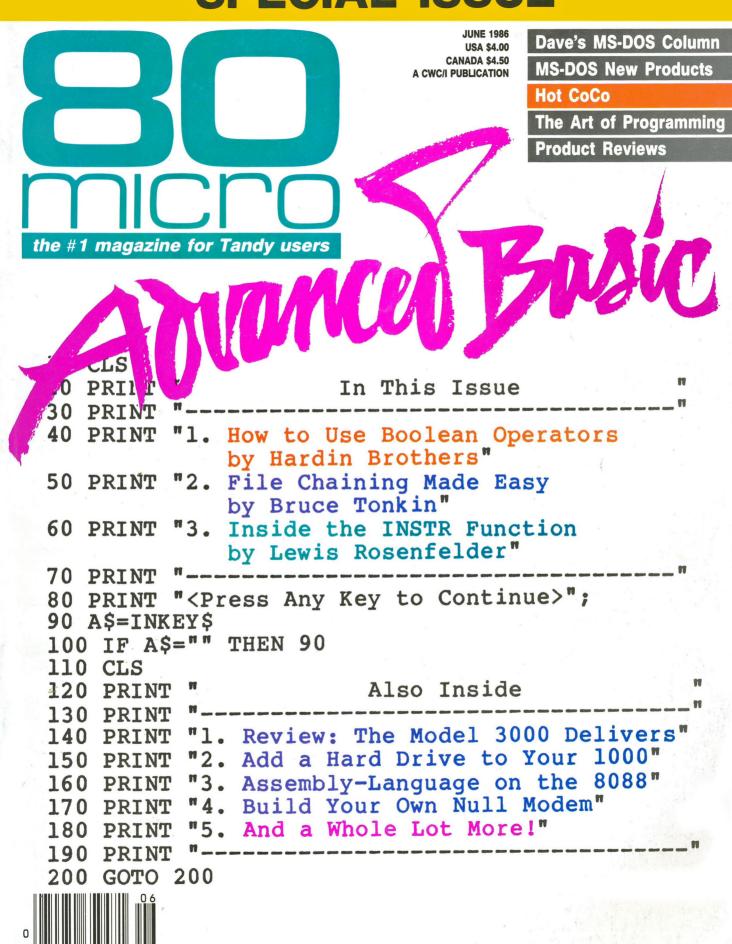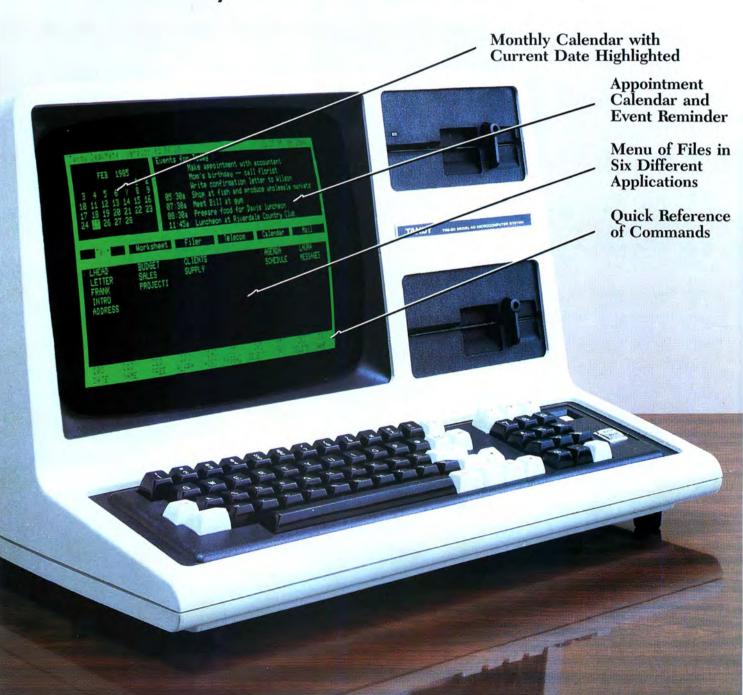
A

The Tandy Model 4D with DeskMate. Only $1199.

It runs hundreds of programs... and your first six are on us.

Monthly Calendar with
Current Date Highlighted

Appointment
Calendar and
Event Reminder

Menu of Files in
Six Different
Applications

Quick Reference
of Commands

Ready to use the minute you put it on your desk

The Tandy Model 4D (#26-1070) is the perfect desktop computer for busy managers, professionals, educators, and home users. The Model 4D even comes with valuable DeskMate software on diskette, featuring the applications today's computer user wants most.

Plus, because a huge software library is available, the Model 4D can be used for forecasting, word processing, database management, education, and for thousands of other applications.

WORKSHEET Lets You Set Up a Budget and Ask "What If . . . ?" Questions

Get the DeskMate® advantage!

DeskMate puts your Model 4D to work right from day one with word processing, spreadsheet analysis, electronic filing, telecommunications and electronic mail. DeskMate is extremely user-friendly, using the same commands throughout the programs. Each day you'll be greeted by a menu that displays a calendar, appointment schedule and a list of your files.

TEXT. Word processing made simple. Control-key editing features make it easy to compose and edit text right on the screen. Then print correction-free copies as often as you wish with an optional printer.

WORKSHEET. This electronic spreadsheet features "plain-English" entries to make complex calculations. View countless "What if . . . ?" situations in seconds. You can print all or a portion of your spreadsheet. It's perfect for everything from personal finance to marketing.

FILER. Create a name-and-address file, small inventory list, recipe file, and more. Simple commands let you create and find files easily. You can sort files by one or more fields and print your files in any order you wish. Keep all your important information at your finger tips.

TELECOM. By adding a telephone modem, DeskMate lets you communicate with a host computer, information service, or another terminal. Information you receive can be printed or stored on diskette. You can also send files to other computers. TELECOM will even dial the phone number of anyone listed in FILER—at the touch of a key.

TEXT Processing Lets You Compose and Edit Letters and Reports

CALENDAR. Set up an agenda for a month, as well as your daily appointment calendar. Finding, adding, and deleting events is easy. And you can also place events in an alarm file that will sound an alarm through the Model 4D's built-in speaker to remind you of appointments.

MAIL. This program lets you send and receive messages to and from other DeskMate users over the phone (telephone modem required).

Self-contained and expandable

The Model 4D is a stylish, self-contained desktop unit with a 12", built-in 80 × 24 display. The 4D comes with 64K of internal memory, plus two built-in double-sided disk drives for 736K of storage.

The Model 4D features a built-in RS-232C serial interface for communications with other computers or information services (requires modem). A parallel printer interface is also included. And you can expand with more memory, one or more floppy drives, and a hard disk.

FILER Program Lets You Set Up a Handy Name-and-Address File

Come in today!

The Model 4D—a versatile, expandable, all-in-one desktop computer at one low price. Drop by your local Radio Shack store or Computer Center and see it today!

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

PowerSoft NewsFlash #8

Thank you for reading our latest installment of PowerSoft's Newsflash. This space will contain information that doesn't really fit into a regular ad-type format. We'd appreciate knowing if these columns are of any value to you. Remember that these columns are written 2-3 months before they are printed, so if something gets out-of-date, that's why, but we try to keep it current! Thanks to all those who have already written or called with your support! **We love to hear from you!** For fastest service on a catalog request, DON'T use the Reader's Service Card, but call or write instead! We appreciate receiving your comments as well as your orders!

A Visit from Peterborough...

We were delighted to meet with Peter Hutchinson and William Smith from *80-MICRO* last week (actually March 12th). They were in the area visiting advertisers and let us in on a few things that will be happening with *80-MICRO* in the near future, and it sounds great! Look for exciting new developments soon! And that means TRS-80 too.

Is the Model 4D dead or alive?

A collection of rumors, news, and views...

A rumor currently circulating that TRSDOS 6™ support will be dropped by Tandy at the end of this year (and therefore the 4 and support thereof) does not seem to be true.

Even though the 4D is not generally carried in the stores, our sources tell us that the 4D will *continue to be available for the next two years.* Probably to fulfill school contracts is a guess, but they can be ordered. ->>*We just received the May *80-MICRO* and were delighted to see that Tandy had a two-page spread on the 4D! That's great, but the ad says to *"see it in your local store."* Trouble is, they aren't there and the manager doesn't want to order one unless he has a deposit! *(Why aren't they in the stores??)*

Hint to Tandy - lower the price on the 4D a bit, put it in the stores where people can see it, and it will *sell and sell* to Model I/III/4 users who don't need a PC but would like to update their hardware and want to stick with the TRS-80! **How about a 4HD? Put a 20 meg drive in it for $1499-$1599!** *How 'bout it?* (These Model I's and III's and even older 4's are starting to wear out! Does that mean that we have to start over??) The 4D will also sell to new customers based on the amount of software available and versatility of the machine. It is a GREAT computer - no question, and can sell right alongside the MS-DOS machines for the right price.

Look how long Apple has been selling the old APPLE II technology! About eight years now and still a very major part of their product line. They would no longer think of dropping APPLE II support, than they would getting out of computers. It is the bread and butter of their product line.

The reality of the PC world is there, however. Apple was not smart enough to recognize this reality until just recently with their recent promise to "bridge the gap" at their January stockholders' meeting. On the other hand, TANDY has done a fantastic job of getting into the PC arena with their very complete and highly competitive line of MS-DOS machines. That should not mean there isn't enough room or time at the Towers for the 4D anymore, however. (Unofficial sales figures we have heard from our sources, *not Tandy*, say that the 4/4P/4D was the second best seller for Tandy last year, topped only by the 1000.) It, like the CoCo and portables, can augment the line. But if they are not carried in the stores, and stay at their current price level, they will die.

OLDTIMER's Dept. (revisited)

Anyone going back a few years with the TRS-80™ will probably remember the name LEO CHRISTOPHERSON (from 80-US). Leo was the absolute master of graphics, and came up with many technics which paved the way for other authors. Leo's main contribution to the history of the TRS-80 was his fabulous games - which are classics. Names like ANDROID NIM, DANCING DEMON, DUELING DROIDS, BEEWARY, SNAKE-EGGS, and others will probably jog a brain cell or two for many. These were the first games to bring SOUND and fully animated graphics to the TRS-80. His characters have a life of their own in the machine. They even blink their eyes! Leo's games were more than fun and innovative. These were not the average shoot-em-up type games. They each taught something - whether it was counting, strategy, skill, humor, or even coordinating music with dance steps. We have recently completed negotiations with Leo to bring back his wonderful games (on disk for the first time) as a low-priced collection titled *LEO'S GREATEST HITS. More info coming soon!*

GOODIES OF THE MONTH

*TRS-80 Goodie of the Month - PowerMAIL Plus for the I/III or 4/4P/4D. Received a FOUR-STAR review in *80-MICRO* and EXCELLENT in all catagories in *INFOWORLD!* The very best mailing list program there is - and it runs on the TRS-80! Normally $99.95, this month we are reducing PowerMAIL Plus to only $79.95! (+$3 s/h)

*MS-DOS Goodie of the Month - Your choice of either DISK OPTIMIZER or DoubleDOS (normally $49.95 each) for only $42.95 each or *both for $79.95!* Save $20! See an earlier Newsflash or SoftLogic ads for descriptions. *Not for Tandy 2000 or 1000* (*if running MS-DOS 2.11.) Tandy 1000 running PC-DOS 2.1 or 3.1 or TANDY's MS-DOS 3.1 is OK.

PowerSoft Products
17060 Dallas Parkway, Suite 114
Dallas, TX 75248 • (214) 733-4475

Add $3 s/h (US) for "GOODIES". Prepaid orders only. • TRS-80 is a trademark of the TANDY CORP.

page 42

page 48

Features

To simplify proofreading and debugging, many 80 Micro Basic listings include a checksum value for each program line. For information on typing in listings containing checksums, see How to Use 80 Micro Program Listings on p. 112.

Departments

page 54

LOAD 80

Load 80 disk and transfer the files to a TRSDOS system disk according to simple on-screen directions. If you own a Model 4, copy the Model 4 programs from the Load 80 disk to your TRSDOS 6.x disk using the COPY 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 system requirements 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 $21.47 or on cassette for $11.47, including postage. To place a subscription order, or to ask questions about your subscription, please call us toll free at 1-800-343-0728 between 9 a.m. and 5 p.m. Or, you can write to Load 80, 80 Pine St., Peterborough, NH 03458.

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.

Load 80 programs are ready to run, and can save you hours of time typing in and debugging listings. Load 80 also gives you access to Assembly-language programs if you don't have an editor/assembler. And, it helps you build a substantial software library.

Using Load 80 is simple. 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, you boot the

---

Directory

Index
Article: Pulling Strings (p. 38).
System: Models I, III, and 4, 32K RAM.

Using INSTR to access and manage random disk files.
Language: Disk Basic.
Cassette filespec: B.
Disk filespec: INDEX/BAS.

Phone List
Article: Truth AND Consequences (p. 42).
System: Model 4 (Models I and III with changes), 32K RAM.

A demonstration program for testing valid phone list input with logical operators.
Language: Disk Basic.
Cassette filespec: C.
Disk filespec: PHONE/BAS.

Data Statements
Article: Have a Nice Data (p. 64).
System: Models I, III, and 4, 32K RAM.

Simplify the process of coding machine language with Data statements.
Language: Disk Basic.
Cassette filespec: D.
Disk filespec: MAKEDATA/BAS.

Keyboard Utility
Article: Different Strokes (p. 72).
System: Model III, 32K RAM; TRSDOS 1.3, Series 1 editor/

assembler.

A copy and macro utility that cuts down on typing.
Language: Assembly.
Cassette filespecs: READYK (SRC), READYK (CMD).
Disk filespecs: READYKEY/SRC, READYKEY/CMD.

TRSDOS 6
Article: The Next Step (p. 106).
System: Model 4, 64K RAM; EDAS editor/assembler.

Inside the TRSDOS 6 drive control table.
Languages: Basic, Assembly.
Cassette filespecs: F, SWAP (CMD), PARK (CMD).
Disk filespecs: DCT/BAS, SWAP/SRC, SWAP/CMD, PARK/SRC, PARK/CMD.

Checksum
Article: How to Use *80 Micro* Program Listings (p. 112).
System: Models I, III, and 4, 32K RAM.

Each line of most *80 Micro* Basic programs contains a checksum, which is equal to the sum of the ASCII values in that line. Use our checksum program to check the listings you type in against the original program.
Language: Disk Basic.
Cassette filespec: E.
Disk filespec: CHECKSUM/BAS.

BAS = Basic   SRC = source code   CMD = object code

TANDY IS DANDY...

...until you compare

	Tandy Memory Plus	Zuckerboard Expansion Memory
DMA	Inc.	Inc.
with 256K	$319.95	$109
with 512K	$519.90	$149
Options:		
Clock/Calendar	$ 99.95*	$49
256K Upgrade	$199.95	$49
RAM Disk/ Print Spooler	N/A	$49
Warranty	90 Days *with mouse	2 Years

MADE IN U.S.A.

INTRODUCTORY OFFER   SMA

Send the Zuckerboards I need for my TANDY 1000 right away. Here's my check ☐ or money order ☐ **MADE PAYABLE TO A.T.D.**

_____ EM with 256K	@ $109 = _____	Subtotal _____
_____ EM with 512K	@ $149 = _____	CA residents
_____ Clock/calendar	@ $ 49 = _____	add 7% tax _____
_____ 256K upgrade	@ $ 49 = _____	Shipping _____ $6.00
_____ RAM Disk/		**Amount**
_____ Print spooler	@ $ 49 = _____	**Enclosed**
_____ Multifunction Board	@ $249 = _____	
_____ 10 MB Hard Disk	@ $529 = _____	
_____ 20 MB Hard Disk	@ $599 = _____	

NAME_____

ADDRESS_____

CITY _____ STATE _____ ZIP _____

TELEPHONE ( ) _____

A.T.D. 1287 Lawrence Station Rd., Sunnyvale, CA 94089

NEW!!

ZUCKERBOARD PRODUCTS FOR THE TANDY 1000

Multifunction Board $249
Includes DMA controller chip, 512K memory, Clock/calendar with 20 year battery, Serial port, and Ramdisk/Print spooler software.

Internal Hard Disk   10 Mbyte $529
                          20 Mbyte $599
Half-height. Includes controller and cables. Mounts internally, above floppy drive A. Requires system with DMA already installed.

CALL TOLL-FREE TO ORDER
        CA (800) 233-6874
           (800) 222-4920

Dr. Dr. Zucker

It's another
ZUCKERBOARD

ⓐⓣⓓ   1287 Lawrence Station Road • Sunnyvale, CA 94089 • (408) 734-4631

ZUCKERBOARD is a Trademark of Advanced Transducer Devices, Inc.
TANDY 1000 is a Trademark of Radio Shack, a Division of Tandy Corporation.

All prices subject to change without notice due to fluctuations in the chip market.

MS-DOS or Bust

In the fall of 1984, I had dinner with the president of a company that sold software for Tandy's I/III/4 line. Remarkably, he wasn't interested in marketing products for MS-DOS computers, even though Tandy was about to introduce the Model 1000. He was a dyed-in-the-wool Z80 man and didn't want to learn a different microprocessor. He also was intimidated by the massiveness of the MS-DOS market and certain that he would quickly be overwhelmed by the many larger companies.

"When the Model 4 goes, then I'm gone," he declared.

A year and a half has changed his perspective. His company now is not only selling MS-DOS products, but is advertising them in the PC magazines.

He's not alone. Driven by an instinct for self-preservation, an increasing number of third-party manufacturers in the TRS-80 and Color Computer markets are stepping into the MS-DOS fray.

The shift to MS-DOS is no surprise. The companies in the Tandy market have hung on over the years because they know how to stay with the consumer. They moved from the I to the III and then to the 4, and they realize that MS-DOS is the next logical step.
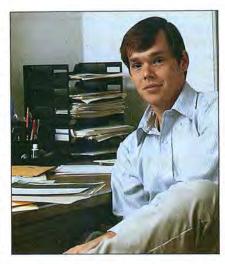
In fact, companies that are not developing MS-DOS products are committing suicide. Vendors that try to survive by selling exclusively to the Z80 and 6809 markets are destined to follow in the failed footsteps of the companies that stuck exclusively with the Model I.

To Market, to Market

The market opportunities unquestionably exist and are growing. Tandy is selling Model 1000s at a healthy clip. The folks in Fort Worth claim that they've overtaken Compaq as the number 1 compatibles manufacturer. Some people say that Tandy is matching IBM sales machine for machine.

The proud new owners of Model 1000s are looking for all sorts of products—memory boards, disk drives, monitors, and software. The companies that supply these people are guaranteeing their own health and welfare. As Bob Dylan once wrote, you don't need a weatherman to know which way the wind blows.

Old-guard Tandy vendors are not the only ones who stand to benefit. The PC market is loaded with companies selling every manner of product to IBM owners. Smaller vendors would be smart to think about shifting to the Tandy 1000 market, where the competition isn't as stiff and customers are searching for products. Some might be better off as larger players in a smaller market than as a smaller players in a larger market.

Of course, Tandy is doing whatever it can to hold on to the new users. Its Express Order Software (EOS) service is a vigorous attempt to get people to buy software at the local Shack or Computer Center. But Tandy's strategy has weaknesses. Most obviously, not every third-party vendor is selling through EOS. Also, not every store manager is doing his best to encourage EOS sales. Finally, the consumer often pays more through EOS than he would by mail order or at a generic computer store.

Also helping the market is Tandy's unwillingness to support third-party hardware products. Fort Worth would much prefer that you give your local RSCC $320 for a memory expansion board than see you pay someone else $110 for an equivalent product.

The number of Tandy 1000s in use, combined with Tandy's inability to serve all of the people all of the time, leaves the door wide open for companies interested in expanding their product lines.

Consumer Report

Of course, the consumer is who will most benefit from a resurgent market.

Model 1000 owners will have a choice of products from vendors who treat the 1000 not simply as another PC clone but as a distinct entity. These companies will understand that Model 1000 users have unique needs to be met and special problems to be solved, and will tailor their services accordingly.

Users of the I, III, 4, and Color Computer will be assured that the companies they've depended on over the years will still be around.

But the Tandy market will not grow on its own. Its vitality depends on the support of the consumer. You can take an active role in determining whether manufacturers commit themselves to developing and selling MS-DOS products.

First, if you're looking for a specific product, don't simply walk into your Radio Shack store and buy whatever they've got in stock. Shop around. Compare product features and prices. Call advertisers in 80 Micro and the PC magazines and find out if they've got what you want.

Second, ask third-party vendors whether they support the Tandy 1000. Don't settle for some answer like, "Yeah, we support most PC compatibles"; it's a sign that the company probably doesn't understand the difference between a 1000 and an IBM PC.

Finally, let manufacturers know what you think they should be making or selling. A good company listens to its customers. If it starts receiving regular requests for a particular product, it might decide to market that product. We know of at least one company that is now selling Tandy 1000 memory boards in response to consumer demand.

As Tandy goes, so goes the market. Tandy has done its part; the rest is up to us, the vendors and users.

Shack Flak

Over the years, many TRS-80 and Tandy owners have complained about lousy service from their local Radio Shack stores. On the other hand, I've heard many customers who've had nothing but praise for Radio Shack.

I figure the time has come for an informal survey of 80 Micro readers. What kinds of experiences, good or bad, have you had with Radio Shack?

If you feel good about Radio Shack, send your letter to Don't Whack the Shack, 80 Micro, 80 Pine St., Peterborough, NH 03458. If you feel otherwise, send your letter to Whack the Shack at the same address. We'll publish the best letters in an upcoming issue. ■

See What You Can Do With
DOTWRITER!

DOTWRITER lets you create spectacular, eye-catching signs, invitations, letterheads, large sideways banners, catalogs, or even books. It is just what you need to turn your dot-matrix printer into a versatile typesetting machine. And it is available for your TRS-80 **Model 4/4P** (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 (black on white), and even print right-justified proportional text.

DOTWRITER includes the printing program, complete documentation, and fourteen useful typefaces (60 to 90 characters per typeface). We will include the 170-page Letterset Reference summary at half-price ($10.00) with your order.

To use DOTWRITER, just write your text with any popular TRS-80 Word Processor (such as ALLWRITE or SuperScripsit), add the necessary formatting commands, and DOTWRITER will do the rest.

36 more letterset disks are available separately. Each has 3–12 complete typefaces. The disks costs less than $25 each and you may purchase them at any time.

SIDEWAYS
SPREADSHEETS

If your VisiCalc spreadsheets are too wide for your printer, our "LONG-VIEW" option may be just what you need. It is an add-on that turns spreadsheets sideways so that DOTWRITER can print them down the page instead of across. LONGVIEW comes with three additional fonts.

EQUIPMENT
REQUIREMENTS

DOTWRITER needs a TRS-80 I, III, 4 or 4P with 2 disk drives and 48K of memory. Separate versions of DOTWRITER support EPSON MX-80 with Graftrax, MX-100 with Graftrax-Plus, and FX, JX, RX; C.ITOH 8510/1550; MICROLINE 84-2/92/93; RADIO SHACK DMP 110-2100/CGP-220; GEMINI 10X/15X and other STAR printers.

We printed our samples on an Epson; sizes may vary on other printers. Many of the fonts shown above are available at extra cost.

Send for free print samples! We've only shown you a few of the 240 DOTWRITER fonts. For the best in TRS-80 graphics printing, we suggest you order DOTWRITER today, toll-free.

80 readers' choice micro 19 83

Please specify Printer and Computer when ordering.

DOTWRITER	$99.95
LONGVIEW	29.95
Additional Letterset disks (3–12 fonts/disk)	24.95
	3 for 49.95
Letterset Reference Book	20.00

FREE bonus disk with two Banner fonts when you order DOTWRITER!
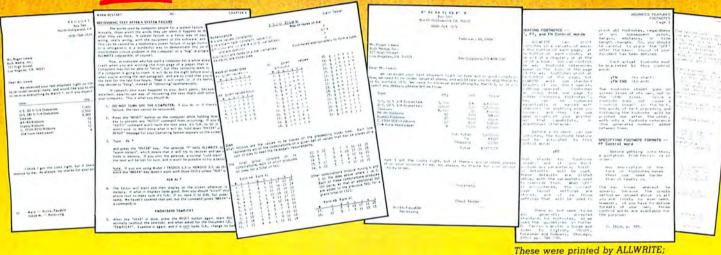
ORDER NOW, TOLL-FREE
(800) 824-7888, oper. 422

PROSOFT®

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. Sales tax in CA. Most orders filled within one day.

will skip bad sectors, read the rest of the file, 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 pre-programmed 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.

ALLWRITE's superb documentation will get you started quickly. 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 23 years, and understand the importance of good documentation.

ALLWRITE works with all major DOS's on Models 1, 3, and 4/4P.

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.

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

Note to teachers: ALLWRITE makes it very easy to generate multiple-choice exams *and* answer keys. Ask for free instructions when ordering.

Mail-Order Maxims

by Daniel Harmon

Guidelines to making successful mail-order purchases.

Mail-order business is as American as apple pie, but you might encounter problems not found in over-the-counter merchandising. Apart from the obvious fact that you can't examine a product until it arrives, you might receive goods flawed because of negligent manufacture, inspection, or handling. And if you find yourself slighted, accidentally or maliciously, you have to contend with a stranger hundreds of miles away in a different legal jurisdiction.

That doesn't mean you should shun the mails. A shrewd buyer can take steps before ordering and after delivery to ensure satisfaction. And the mail-order branch of the microcomputer industry does lie within reach of our system of justice.

Under the Uniform Commercial Code, a buyer can reject a product "within a reasonable time" if, after usage, it evidences a breach of warranty. What's a reasonable time? That would be up to a court to decide. Just make sure you bring any problem to the seller's attention as soon as you discover it.

An Ounce of Prevention

First, you should follow these steps when you order a product by mail:

● Read the advertisement's fine print and know exactly what you're buying. If you can't find details you want in the ad; if some of the published information seems unclear, contradictory, or wrong; or if you don't understand the meaning of a disclaimer, phone or write the seller for clarification before placing your order.

● Ask in advance about the seller's return policies. Some houses charge "restocking" fees; some require that the buyer pay return postage.

● Never send a cash payment by mail.

● When your order arrives, save all boxes, wrappings, instructions, and other papers until you're sure the product is OK. If it was damaged or lost in transit, carriers usually will make amends. United Parcel Service, for example, either sends someone to check the package or returns it to the shipper at no cost to you. If the U.S. Postal Service handled the shipment, you can lodge a complaint with your local post office or contact Consumer Advocate, U.S. Postal Service, Washington, DC 20260.

● Never return a damaged product without communicating with the seller first. True, you might save time by returning it immediately, and you might receive a replacement or refund by return mail, but you could also create complications for yourself.

How to Complain Effectively

What happens when you've paid your money, taken delivery, and you wind up with shoddy merchandise or an item you don't want? What happens if you send in your check and the days turn to weeks, the weeks to months, and still you receive nothing?

You're not at the seller's mercy; you do have recourses. But before you take action, remember that it's most unlikely that you're dealing with a crook. The U.S. Office of Consumer Affairs estimates that consumers resolve 88 percent of their problems simply by contacting the seller or the seller's supervisor. Mail-order merchants generally aren't out to steal people's money, they're out to do business.

If you do run into a problem, you should first write a letter to the company's president or consumer contact. Your letter should include the dates you ordered and received the item and identify the product by model name/number and serial number. Detail the problem and ask for the form of satisfaction you deem appropriate: replacement, refund, or substitution. Attach copies, not originals, of the advertisement that attracted you and of receipts, warranties, canceled checks, or other pertinent documents.

Bear in mind that it won't hurt, and might help, to squelch your irritation and keep the letter's tone friendly. The person you're writing probably isn't the person who caused your woes, but one who can expedite a remedy.

Unless the seller specified that delivery might take longer, the law requires that vendors deliver merchandise within 30 days of the order. If a company fails to deliver within the allotted time, you have the right to cancel your order at no cost. If you exercise that right, the seller has seven days to refund your money or credit it to your charge account.

Note that when you negotiate mail-order problems, the last thing you should do is threaten to "turn the matter over to my lawyer." Hiring a lawyer should be your final resort. Lawyer's fees aren't cheap, and winning your case won't necessarily exempt you from having to pay costs.

Keep copies of all correspondence to and from the seller.

Wait three weeks. If the matter isn't settled, try a second letter, repeating your complaint and declaring that if you don't get immediate satisfaction, you'll notify the deputy chief postal inspector in the vendor's home region.

For the vendor, this now becomes serious business. The U.S. Postal Service has a nationwide law enforcement arm authorized to investigate mail fraud. The Postal Service can cut off mail deliveries to a vendor's address. If the Postal Service convicts a seller of mail fraud, he could face a stiff fine and possibly jail.

The government is most likely to launch a full investigation when it receives numerous complaints against a seller. Moral: Don't hesitate to report a truly serious problem. If you're really up against an unscrupulous dealer, yours could be the complaint that brings Uncle Sam into the action.

Other consumer defenders include the Mail Order Action Line of the Direct Marketing Association (6 E. 43rd St., New York, NY 10017). This commercial trade association follows up complaints by contacting vendors. It has no power of enforcement, though.

Better Business Bureaus and state or local consumer protection offices might be able to help you, but it can be confusing to find the right place to make your particular complaint. The U.S. Office of Consumer Affairs' *Consumer Resource Handbook* lists the appropriate agencies. You can get a free copy by writing to Handbook, Consumer Information Center, Pueblo, CO 81009.

Finally, if you're having a problem with an *80 Micro* advertiser, you might want to send a copy of your complaint to *80 Micro*, 80 Pine St., Peterborough, NH 03458. ■

Daniel E. Harmon is the managing editor of The Lawyer's PC *newsletter. P.O. Box 1108, Lexington, SC 29072.*

One Lost Sheep

A letter from Paul R. Withun of El Paso, TX (Input, March 1986, p. 12) produced several dozen impassioned responses from readers. After hearing a friend boast of the "whole line of equipment" and plentiful free software for his Commodore 64, Paul regretted buying his Model 4.

—Eds.

I'm so sick of hearing "Trash-80" that I'd like to chainsaw every Commodore computer ever made. Stand tall, shoulders back, chest out, face to the wind, and take confidence in yourself and your computer.

Who was calling the Tandy line "Trash-80" in the early Model I days? Some of the greats of modern computers, such as LNW, SOL, Pets, Atari, Timex/Sinclair, and Apple with its disk Basic and overheated boards. Where are many of these name-callers now? In the "trash" can.

They were buried by fantastic numbers of intelligent, clear-thinking, enthusiastic Tandy owners. Paul's friend, the Commodore 64 owner, bought himself a toy computer—pretty pictures and sounds, and that's about it.

Paul should join a TRS-80 users group and take advantage of classes at his local Radio Shack Computer Center.

Long after the Apples have gone to seed, the PCjr is a memory, and the Commodore has gone down with the ship, the TRS-80 will be humming along—balancing the checkbook, helping the kids with homework, writing letters, playing games, and more.

Ted Paul
Fredonia, NY

The Model 4 is one of the most powerful 8-bit systems available. Some of its capabilities, such as pipes and filters, are often found only in far more expensive and complex systems. The true powers of the Model 4 will become evident to Paul as he delves deeper into the joys of computing. If he chooses to expand his system, Radio Shack offers a wide range of choices—additional drives, high-resolution graphics, and communications capabilities.

For software, he should start with *The 1986 Radio Shack Software Reference and Tandy Computer Guide*, which con-

tains 96 programs for the Model 4. The book covers only programs available directly through Radio Shack. We also offer an *Applications Software Sourcebook* (catalog number 26-1378, $7.95) listing more than 2,500 ready-to-run programs sold by a variety of vendors.

CompuServe Information Service offers free software on its LDOS, Tandy Professional, and Education special-interest groups (SIGs) for the Model 4. Paul can also communicate with other users through CompuServe.

With all this support, I hope Paul can find the help he needs to take full advantage of the Model 4. Computers are wonderful tools for work, learning, and fun. All it takes is an inquiring mind and a sense of adventure.

Amy Arutt
Asst. manager of
marketing information
Radio Shack

The Model 4 is not a games machine, like the Commodore 64, but it's a damn good business machine. I bought mine in August 1984, a year before Paul got his, and I've never regretted my decision. I've found that the Model 4 per-

80 Micro's BBS is open 24 hours a day. It offers programs you can update and download, special-interest groups, and a classified section. You can reach the board at 603-924-6985; UART settings are 300/1,200 baud, 8-bit words, 1 stop bit, no parity.

forms at least as well as MS-DOS computers in the applications that are most important to me.

The mailing list and bookkeeping programs for my business, a New York City repertory company, are modified versions of programs I copied from books and *80 Micro*.

I, too, have had to endure childish references to "Trash-80," but so what? I wouldn't have subjected my business to the clumsiness of a Commodore toy computer for the sake of some ignoramus' opinion.

Jerry Engelbach
New York, NY

While the early Model I days might have rightfully led, in some cases, to TRS-80 being taken as an abbreviation for "trash," subsequent Tandy computers are more accurately described as "the right stuff."

Anthony G. Perri Jr.
Kenosha, WI

Paul's friend with the Commodore 64 may call the Model 4 a "Trash-80" if he wants, but that's only because he stumbled onto a computer that matches his intelligence, rather than one that *requires* intelligence.

David J. Oleksak
Vancouver, WA

Paul hit the nail on the head.

A fairly sophisticated software package for the Commodore 64 costs $39.95, while a comparable Model 4 package from an *80 Micro* advertiser costs $100.

Commodore will eventually bury Tandy in the recreational PC market because the Model 4 line isn't designed for the recreational user any more than the IBM is. Unfortunately, many of us who wanted to use the Model 4 as a hobby learned this too late.

Jon P. Fishback
Sunnyside, WA

I get sick when I see all the Commodore 64 programs in stores. There seem to be so few programs for the Model 4, and they cost much more than Commodore programs.

Radio Shack doesn't approve of computer clubs that would be helpful to their customers, perhaps because they fear programs would be traded and they

wouldn't be able to sell the few they have.

Lois E. Patterson
Normal, IL

Is It Safe?

As an electrical engineer, I recommend that people *not* try Joe Lindsey's modification of the Epson MX-series printer to increase printing speed ("Speed Writing," January 1986, p. 52).

With this change, the Epson print head heatsink, stepper motors, and power supply will run hotter and ultimately fail because they weren't designed to dissipate the extra heat. I've also seen power transistors and other devices explode from overheating.

Epson designed the printer to run slower than it is "capable" of for just this reason.

Lee Pelletier
Wenham, MA

I disagree with Mr. Pelletier on the heatsink's limitations. Epson engineers built in a timer to regulate power in the print driver circuit. A balancing effect between pulse width and frequency keeps the power dissipation within safe limits.

Although I, too, have seen components swell, explode and even vaporize trying to dissipate too much power, I haven't noticed even the lights dim when I turn on my faster printer. Several people I know have been using the modification for more than a year without incident.

—Joe Lindsey

Bring Back Piracy

I beg to differ with Eric Maloney's assertion that copy-protection is no longer an issue (Side Tracks, September 1985, p. 8). *80 Micro* has simply chosen to ignore it.

Apart from TRSDOS, I know of no Tandy disk program that isn't copy-protected. I've passed up VisiCalc at a terrific sale price because I won't knowingly buy protected media, period. You might make friends among TRS-80 users by running articles revealing Tandy's copy-protection schemes.

Willi Wald
Hamilton, Ontario

Starbridge Reinforced

An improved Starbridge DOS has been released since David Engelhardt reviewed the first version (May 1986, p. 31).

This faster version lets you copy twice as many files, put system files on disk with the format option, and run 25 (instead of only nine) programs from the custom menu. It has back-up and restore options, improved on-screen keyboard help, and is no longer copy-protected.

Kim Mackintosh
Starbridge Technologies
Los Angeles, CA

Translation, Please

I will be obtaining an editor/assembler program so I can type in Assembly-language listings from *80 Micro*. I have read that all assemblers handle the programs a little differently.

I suggest that you require authors to mention the name of the assembler program used, so those of us who don't really know Assembly language can copy, assemble, and run these listings without changes.

Frederick L. Mass
Garnerville, NY

Most Model III source code listed in 80 Micro was created with Radio Shack's EDTASM, most Model 4 source code with EDAS. In future issues, we'll specify the editor/assembler used. For a discussion of differences in editor/assembler programs, see Hardin Brothers' "Mutual Understanding," (June 1985, p. 82).

—Eds.

Misosys' *Guide* Fades Away

Your readers may be interested to know why we at Misosys chose not to reprint *The Programmer's Guide to TRSDOS*.

The *Guide* was first published in August 1983. Since the book was directed at the Assembly-language programmer, it was devoted more to the hacker than the general Model 4 user. The bottom line total is 1,248 books sold in over two years—a dismal showing.

We just don't expect the market to present significant quantities of new Model 4 owners to justify a reprint of 500 books—the minimum quantity necessary for reprinting. If the TRS-80 community of users wishes us to continue publishing this book, they need to get more serious in their purchase of it.

Roy Soltoff
President, Misosys Inc.
Sterling, VA

Fleet-Footed Model 4

It would be rather presumptuous to say that the Model 4 is the best microcomputer on the market. But for those who write programs in Basic, that's not far from the truth. The diversity and completeness of commands available in Model 4 Basic are matched only by the IBM implementa-

tion. I've recently found also that Model 4 Basic is the fastest interpreter among the major Basic implementations.

As an example, I ran this one-line program:

10 FOR I = 1 TO 5000:A = A + 1:NEXT

on different computers with the following results:

1. Model 4, 15 seconds
2. IBM PC, 16 seconds
3. Apple II +, 18 seconds
4. Apple IIe, 18 seconds
5. Commodore 64, 20 seconds
6. Model III, 34 seconds

Obviously, these figures will vary depending on the operations involved, but the ranking remains the same in most cases (the IBM, for instance, deals with complex operations better than the Model 4 does).

Roberto Refinetti
Santa Barbara, CA

Of TRSDOS, MS-DOS, And Tandy

I've learned to appreciate the TRSDOS 6.2 operating system and other programs by Logical Systems Inc. In fact, I've introduced 11 people to the Model 4. They seem to like their machines as much as I like mine.

Now Tandy wants to put all their efforts into MS-DOS machines.

If MS-DOS (and the software that runs under it) is the best the computer industry can come up with, and if Tandy lets the Model 4 die, then it's back to pencil and paper when my Model 4 dies.

William R. Greep
San Jose, CA

Refundamentals

Your readers may not know that if a Radio Shack product goes on sale within 30 days of purchase, they can get a refund for the difference if they request one within the 30 days. Tandy says the customer is, in effect, returning the merchandise for a full refund (as company policy explicitly allows) and buying it back at the reduced price.

James P. Wright
Corpus Christi, TX

A Tandy spokesperson says such refunds aren't part of official policy, but Tandy is committed to resolving any reasonable complaints that customers bring to its attention.

—Eds.

Send your correspondence to Input, 80 Micro, 80 Pine St., Peterborough, NH 03458. We reserve the right to edit letters.

Send your questions or problems dealing with any area of Tandy/Radio Shack microcomputing to Feedback Loop, 80 Micro, 80 Pine St., Peterborough, NH 03458.

Q: Lawrence Pavco's disk drive cleaning program in the November Feedback Loop (p. 14) works well, but since I don't have four drives, I made the following changes:

```
10 CLS:Z$ = " ":POKE16412,1
```

which stops the cursor from blinking (a personal preference);

```
51 IFAA$ = "A"THEN64
```

which changes the pointer to line 64;

```
64 INPUT "ENTER NUMBER OF DRIVES";
   T:N = T – 1
```

which takes your keyboard input on the number of drives and assigns it to a variable; and

```
70 FORC = 0TON:'the rest of this line remains
   the same
```

which inserts the drive variable into the program. Also, the colon that should be after the first command statement in line 70 was missing from the Listing. *(Darcy Swensen)*

A: Good work on solving the problem. Thanks.

Q: I've been trying to interface a Model 4 to communicate via the RS-232 with a 1050 General Electric Numerical Control used in machine tools.
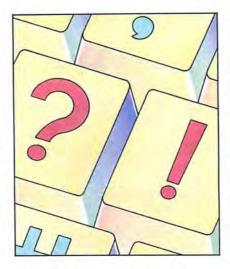
The RS-232 is set and connected as follows:

```
        DB-25         DB-25
          2———————————3
          3———————————2
          4—┐       ┌—4
          5—┘       └—5
         ┌—6           6—┐
         │ 7           7 │
         └—20—————————20—┘
```

Set * CL to COM/DVR
Setcom (B = 1200, S = 2)
Link * DO * CL
Link *K1 *CL
BASIC

I can communicate perfectly from the Model 4 to the GE Control. But when I try to communicate back to the Model 4, I get only part of the information and the error message "Undefined line number, syntax error."

Maybe the GE is too fast for the Model 4. Do you know what I'm doing wrong? *(Georges Rouleau, Redford, MI)*

A: Pin 4 of the GE should be connected to pin 5 of the Model 4, and pin 5 of the GE to pin 4 of the Model 4. Pin 4 is normally the hardware request-to-send line and pin 5 is the hardware clear-to-send line. For more information on building an adapter device, see "Transfer Points," p. 78 of this issue.

Q: I own a Tandy 1000 with the one-chip BIOS ROM, version 1.00.00. Is Tandy ever going to notify me of the updated ROM version 1.01.00 and whether I'll be allowed to upgrade? Will they charge me because I own a 1000, and will I be allowed to put the new BIOS in my own computer? I feel left in the dark. *(Jeffrey Johnson, Mesa, AZ)*

A: Tandy usually tries to notify computer owners of updates, but can't always keep up with the hundreds of thousands of machines sold every year. The solution is to periodically stop by your local Radio Shack Computer Center and ask about updates.

You don't need to change BIOS ROMs because you already have the most recent version. The original Model 1000 used the two-chip BIOS, but now Tandy is shipping it with the one-chip BIOS ($13.01). The two-chip BIOS is not even available as a replacement part. If your BIOS dies, Tandy automatically replaces it with the one-chip version.

When you change the BIOS chips, you also have to change the memory array chip, U53 ($7.15), so the BIOS can correctly address the memory. You also need to change the zero-ohm jumper resistors R1 and R2. If you have the one-chip BIOS, R1 must be jumpered. The two-chip BIOS uses R2.

Q: I'm having a problem with the GW-Basic (2.02) Screen statement on my Tandy 1000. In the older version of GW-Basic, you could display screen zero while the program was writing in screens 1, 2, and 3. It seems this feature doesn't work in Tandy's new version, 1.01. The active screen will display itself regardless of which screen I select for display.

For example, if I code SCREEN 0,0,0,0, the Print statements that follow will be written to screen zero. With SCREEN 0,0,1,0, the Print statements should be written to screen 1 and screen zero should remain on the monitor, unchanged—that is, until the statement SCREEN 0,0,1,1 is issued, when the monitor will display the contents of screen 1.

In the new GW-Basic, every time you change the active screen the display screen disappears and the active screen writes itself on the monitor. This is annoying because the software I've written will no longer work properly. Any suggestions? *(Jim Woodruff, Pineville, LA)*

A: That one has me stumped. Microsoft calls the change an "improvement," and won't do anything to help. Any suggestions from readers?

Q: I own a Tandy 1000 computer and a VM-2 composite monochrome monitor. The display leaves much to be desired. I do a lot of work on spreadsheets and find the poor resolution of a composite monitor annoying. Will a color (RGB) monitor give me better text? If not, can I use a TTL monitor and display card, and do I have to disable the built-in display electronics? *(David G. Trulli, Playa del Rey, CA)*

A: The composite signal is the problem: as long as you use it you'll have poor resolution. Switching to an RGB monitor and output will noticeably improve text quality.

You don't have to go to TTL electronics, but if you do, don't worry about the built-in electronics. The new TTL card will just intercept the bus line, take out the information it needs, and send it through its own output port.

Q: I've never received update notices for the MS-Pascal and MS-Assembler software I bought a little over a year ago for my Tandy 2000. Advertisements for these packages state that new versions are available. My Radio Shack dealer

thinks new versions don't exist. Any suggestions? *(Peter M. Jagielski, Ada, OH)*

A: There may be new versions of MS-Pascal and MS-Assembler that don't run on the 2000, which isn't completely IBM-compatible. Most MS-DOS packages require modification before they'll work on the 2000. Unless Microsoft says there is an updated version of those programs for the 2000, no updates or upgrades are available.

Q: I've written a program (see the Program Listing) to help Gil Seiler suppress the Daisy Wheel II printer's built-in line feed (October 1985, p. 16). I'm not sure the program will work with Gil's AT&T 6300, since that means working with the BIOS ROM, but it works on my IBM PC.

I used PC-DOS INT 27H (terminate-but-stay-resident) to interject a file terminate in INT 17H. Because the Daisy Wheel II cable has no pin provision for the hardware interrupt, I decided to send in the two codes each time INT 17H experienced a carriage return (0D hexadecimal).

The procedure is as follows: assemble with MASM (ASM), link, and generate the COM file with EXE2BIN DAISY_II.EXE DAISY_II.COM. *(Frank P. Barnes, New York, NY)*

A: Thank you for the solution.

Q: I've installed two double-sided drives in my Model 4 (early version, black-and-white monitor), and built two power supplies so the standard Radio Shack drives can be used externally. I realize the drives must be addressed as zero and 1; this was no problem, since I had an old Model I cable to guide me. Unfortunately, the drive motors start when the power is turned on and won't turn off, even after a drive has been addressed.

I was able to access the drives as 2 and 3 after realizing the drive-select lines are different. They work but still won't turn off. I know my cable is working because another external drive works fine as either drive 2 or 3, and it turns off when it's supposed to.

Perhaps the boards for these drives are different. Should I cut a trace or install a jumper? There are no jumpers on the board to configure the drives. *(Pete A. Lopresti Jr., Birmingham, AL)*

A: You forgot to put a termination resistor pack in either of the outboard drives. The floppy drive controller automatically includes a termination resistor on the disk drive control lines, but has no such provision for outboard drives. You must supply a termination resistor for the last drive on your out-

Program Listing. Removing the Daisy Wheel II printer's built-in line feed.

```
; DAISY_II.ASM generates a COM file that stays resident in DOS. Whenever
; there is a CR code (0DH), the resident program sends codes 1BH (27) and
; 15H (21) to the printer. This causes the DAISY-WHEEL II printer to
; cancel the extra line-feed it normally would generate.

INTERRUPTS          SEGMENT AT 0H        ;This is where the printer
            ORG     17H*4                ; interrupt holds address of
PRINTER_INT LABEL   DWORD                ; service routine.
INTERRUPTS          ENDS
CODE_SEG            SEGMENT
            ASSUME  CS:CODE_SEG
            ORG     100H                 ;Make into COM file.
FIRST:      JMP     LOAD_CR              ;First time jump to initialize.
            OLD_PRINTER_INT  DD          ? ;Location of original printer
                                         ; interrupt.
LF_OUT      PROC    NEAR                 ;INT 17H will now come here.
            CMP     AX,0DH               ;Is character a CR (0DH)?
            JNZ     OUT                  ;No. Jump to ROM-BIOS routine.
            PUSH    AX                   ;Yes. Hold 0DH in stack.
            MOV     AX,1BH               ;Prepare for first code 1BH.
            INT     17H                  ;Send to DAISY-WHEEL II.
            MOV     AX,15H               ;Prepare for second code 15H.
            INT     17H                  ;Send to DAISY-WHEEL II.
            POP     AX                   ;Bring back 0DH from stack.
OUT:        JMP     OLD_PRINTER_INT      ;Jump to ROM-BIOS routine.
LF_OUT      ENDP
LOAD_CR     PROC    NEAR                 ;Procedure initializes resident prog.
            ASSUME  DS:INTERRUPTS        ;Data segment will be interrupt area.
            MOV     AX,INTERRUPTS
            MOV     DS,AX                ;Puts 0000H segment into DS.
            MOV     AX,PRINTER_INT       ;Get offset address of INT 17H and
            MOV     OLD_PRINTER_INT,AX   ; put it into holding buffer.
            MOV     AX,PRINTER_INT[2]    ; Segment address. We can now jump
            MOV     OLD_PRINTER_INT[2],AX ;to address of ROM-BIOS INT 17H.
            CLI                          ;Disable interrupts during change.
            MOV     PRINTER_INT,OFFSET LF_OUT  ;Now load the address of our
            MOV     PRINTER_INT[2],CS          ; printer routine into ptr int.
            STI                          ;Re-enable interrupts.
            MOV     DX,OFFSET LOAD_CR    ;Set up everything but LOAD_CR to
            INT     27H                  ; stay and attach itself to DOS.
LOAD_CR     ENDP
CODE_SEG    ENDS
            END     FIRST

LISTING ONE.
```

End

board cable. You can order this part, a 6-pin 150-ohm pack, from Radio Shack National Parts (900 E. Northside Drive, Fort Worth, TX, 76102, 817-870-5662). The pack costs $1.08 plus shipping; the part number is ARX0241. You can also have your local Radio Shack store order it, using the reference number 26-1164 in addition to the part number.

Q: I paid $265.35 to replace the power supply in my Model III to repair a shrinking screen display. How can I prevent this problem, and if it happens again, what should I check first?

The computer wouldn't boot when I got it back from repairs. The repairmen said the head amp was loose and repaired it for nothing, but I don't think that was the whole problem. The head amp was replaced one previous time for $100. Can I replace the part myself if it goes again? The *Technical Reference Manual* never refers to the head amp. Does anyone sell a repair and troubleshooting manual for this unit? *(Howard Kepple, Pittsburgh, PA)*

A: Sometimes power supplies go bad and nothing can be done about them. Should yours fail again, consider building one based on the diagram in the technical manual. That should be

cheaper than having the unit repaired; it may, however, cause problems if repairs are needed for some other reason.

There is no part called a head amp: the term refers to the amplitude of the signal sent to the disk drive head. Some drives have a plug-in 3470 chip that controls the head amplitude.

With an oscilloscope, you can troubleshoot the drive head electronics to see if the amplitude is within the manual specifications. If your drive's 3470 chip isn't in a DIP socket, replacing it will be a lot of work.

The latest Model III *Technical Reference Manual* lists both types of drives. If your manual doesn't contain this information, get a newer one from Radio Shack National Parts.

Q: I own a 16K Model III with an Aerocomp single-drive kit, and bought two Jameco Electronics kits to upgrade to 48K. Jameco's instructions only cover 4K to 16K conversions and the company hasn't yet answered my query. I've sent you a copy of the CPU board.

Perhaps I have a bad C ROM. On MEM-TEST (TRSDOS 1.3), the checksum should be 2B91, 2EF8, or 2F65, but my C ROM's checksum is 2F84. *(Bill Rollins, Honolulu, HI)*

A: Your CPU (central processing unit) board has three rows of eight sockets in one corner. Only one row has chips—the other two are empty. To upgrade the memory, plug 16K RAM chips into the empty sockets and you'll have a 48K machine. Make sure all the chips are oriented properly and their notches are pointing in the same direction.

The 2F84 checksum is correct for your computer's ROM C.

Q: I'm having problems with the speed adjustment on a new Radio Shack disk drive for my Model I. When it was new, the drive ran at 301.5 rpm. Most of the time I get a 302-plus reading. At this high speed, the drive will only boot systems and load some programs in drive zero. In other drives, the result is murder to the operator: lost data, parity errors, and so on.

I want to control the speed, as I am able to do with the original Shugart drives. Can you help? *(Randy Schuetz, Mendota, IL)*

A: Speed is not the problem. The drives, controller, and software are designed to allow the drives a 1-percent drift from specifications. As far as the controller and software are concerned, a drive speed between 297 and 303 rpm is OK. The 300 rpm specification is only meant to be an average.

Drive head alignment may be the real culprit. To align the drive yourself, get J&M Systems' Memory Minder diagnostic program for disk drives (15100-A Central SE, Albuquerque, NM, 87123, 505-292-4182). The Model I version costs $89.

Also, Gerard Kiernan's *80 Micro* article "Making Adjustments" (March 1986, p. 44) explains how to modify the drive speed on the Models III and 4 without opening the case.

Q: I'm a nurse and a freelance writer of medical articles and books. For letters and short documents, SuperScripsit performs adequately, but for long documents, the processor is becoming a pain.

On several occasions I've typed in a lengthy manuscript (greater than 20 pages), only to have file access denied, either at the point where I've gone to print a segment or after printing a draft. Sometimes I get the message, "There is no more space left on this diskette," even though there are many free bytes left. When I try to reaccess the file (after going to TRS-DOS to check disk space), I receive the message, "File access denied."

Other times, I'll be typing very fast and "Please wait a moment" comes up, but instead of the usual pause followed by a return to the last character position, the next page comes up blank with an arrow in the upper-left corner. After much confusing shifting between pages when I try to recall the original page, print commands produce a "Please wait a moment" blurb that goes on for many moments. By then, I'm completely hung up and have to reset. Attempts to reaccess the file bring up the message, "Record number out of range." File access is denied.

Should I buy a disk "zap" program, or give up and get another word processor? *(Pat Koch, Temple Hills, MD)*

A: Keep an eye on the file name in the lower left corner of the screen to avoid running out of space. When you start to run out of room, the file name will disappear and the number of spaces left for characters will appear. Quit the file immediately.

The amount of room left on the disk isn't directly related to what the directory says when you call it at the DOS prompt. Don't continue to use a file if SuperScripsit says there's no more room. Start a new floppy disk to continue the document.

If you feel you must stay on the same disk, call the Compress a Document routine from the menu. This may free enough room for your needs.

In the cases where you were able to print a draft, but subsequent attempts to access the file failed, you may have exceeded the available disk room (at least according to SuperScripsit) when you returned to the draft and made editorial changes and additions.

At frequent intervals, stop writing and send everything you have to disk via the control-W command.

I suggest you get the book *SuperScripsit Word Processing For The TRS-80 Models III, 4, and 4P* by Jeanette J. Bieber-Moses (Scott, Foresman and Co., $17.95, ISBN 0-673-18086-7).

Q: I have a Model 4 with two disk drives and 64K, and I'd like to run a four-page Basic program that was written all in capital letters on a Model I. Is there a program to convert capital letters to lowercase? I don't mind having to change the first letter of each sentence, but rewriting the entire program is out of the question. *(Mario G. Mucciolo, Floral Park, NY)*

A: Load the program into memory and save it as an ASCII file with the command SAVE "filename",A. Then write a simple Basic program that reads the target file a line at a time with LINE INPUT, uses the MID$ function in a For. . .Next loop to scan the line and add ASCII 32 to each character (except for those with an ASCII value below 65), and then writes the new, all lowercase line back to your disk. The program works slowly, but it's easy to write and quicker than retyping the whole program.

When you reload the program, Basic will automatically convert the commands and variable names to uppercase, leaving the text (anything between quotes) in lowercase. All that remains is to put capital letters in the proper places.

Q: Is there a program or method to make the Epson printer 100-percent compatible with the Model 4P? *(Leon Field, Tinton Falls, NJ)*

A: I'm not sure what you mean by compatible. Three programs for making the MX-80 and RX-80 *print* dumps of the Model III screen (which the Model 4P emulates) were published in this column in November 1985 (p. 16).

We don't yet have a program listing for dumping Model 4 screen graphics. Does anyone know of one?

Q: Some of the new daisy-wheel typewriters come with computer interfaces, but I've found none for my Model III. Do you know of any? *(D.B. Cameron, Middleburg Heights, OH)*

A: Most typewriters that are supposed to double as computer printers are designed as typewriters first and computer printers second, and are usually meant to work with an RS-232 or a similar interface. Most manufacturers make translation boxes that accept Centronics parallel input and convert it to something usable by the typewriter.

Start by determining which interface is built into the typewriter. If it's a Centronics standard parallel port, you can probably just connect it to the computer printer port, although you may have to get a special cable if the connector on the typewriter isn't a Centronics. If the typewriter has a Centronics connector, you can use the normal printer cable sold by Radio Shack.

If the typewriter uses an RS-232-type connector, you have to configure the computer to send the material to be printed out the RS-232 port (if you have one), or buy the typewriter manufacturer's conversion box. Again, you must make sure the proper cables are available.

It's cheaper to buy an inexpensive letter-quality printer rather than fool around with printer typewriters. ■

Terry Kepner is a freelance writer, programmer, and editor. He writes monthly columns for Portable 100 *Magazine,* Pico, *and* 80 Micro, *and is publisher of* The Kepner Letter.

Zenith Walks Away With Portables Contract

Hot Item

The Internal Revenue Service took industry pundits down a peg in February when it awarded its prized laptop computer contract to *Zenith Data Systems* of Glenview, IL. Estimated at $27 million, the contract is the largest laptop order ever placed by a federal agency. The agreement calls for Zenith to deliver 15,000–18,000 Z-171 laptops to the IRS over the next 18 months. IRS field auditors will use the computers to access the agency's mainframe files, review books, and file reports when conducting on-site tax audits.

Each unit will have 256K of RAM; two 5¼-inch disk drives; a backlit, 10-inch diagonal, 25-line by 80-column liquid-crystal display (LCD) screen; an internal 1,200-baud modem; video adapters for external monitors; and rechargeable battery packs. Zenith will also bundle in applications software, including Enable, an integrated package from The Software Group; RBase 5000, a data base manager from Microrim; and Poly Windows, a desktop organizer from Polytron Corp.

The IRS will pay $1,100–$1,200 per machine, which is the winning bid put forward by Zenith. For the IRS, it's not a bad deal, considering that the units (without the internal modem and video adapter) retail at $2,399.

Industry analysts had all but given the contract to IBM. With Zenith the winner, they still think the contract will "legitimize" the laptop market.

February was a four-star month for Zenith. In addition to the IRS contract, the company won an order from the Department of Defense for delivery of 90,000 Z-200 personal computers to the Air Force, Navy, Army, and Marine Corps over the next three years.

Zenith Data Systems is now firmly entrenched as the number one supplier of PCs and laptops to the government, having won four of the last five major procurements since 1983. (Federal Data Corp. won the other contract.) IBM has yet to win its first microcomputer contract from Uncle Sam.

Tandyland

If you haven't visited your local Ra-dio Shack Computer Center recently, I'd advise stopping by to see if Tandy's unadvertised software sale is still going on. In February, Tandy began shipping software to the RSCCs for quick sale at bargain-basement prices.

The sale has created a yard-sale atmosphere at some of the computer centers, with system-specific software selling at incredibly low prices in bundled packages. One center I contacted was selling a Model 4 "grab box" containing Pascal 4, CBasic 4, Enhanced VisiCalc, and Spectaculator for $69.99. Together, the packages would normally retail for $449.95.

Though a boon for buyers, the sale has RSCC managers wondering what to do with their regular software stocks. The same software packages being offered on sale continue to sit on shelves at their regular prices.

Tandy says the unadvertised clearance is an effort to consolidate stock and relieve two of its Fort Worth warehouses of overstocked items. Tandy gave no indication at press time of how long the sale would continue.

You might also note the price of the Tandy 2000 the next time you visit a Radio Shack Computer Center. Fort Worth slashed the price to $999 for 30 days last winter in order to generate interest in the problem child of its MS-DOS family. The consensus among Tandy watchers at press time was that the company will continue to pare down the 2000's price tag after tallying the results of the 30-day "fire sale."

The success of Tandy's MS-DOS line is raising questions about the life expectancy of the Model 4D. In March, reports on CompuServe and in a TRS-80 newsletter fueled rumors that Tandy had stopped production of the 4D. Amy Arutt, Tandy's assistant manager of marketing information, assured me that the rumors were false. She reiterated the official statement of CEO John Roach that Tandy intends to sell and support the 4D "for at least two more years."

The Tandy 1000, meanwhile, continues to be the computer division's top seller in 1986, with monthly volume sales reaching the tens of thousands. The Tandy 3000 is also selling well, outperforming the company's expectations with monthly volume sales in the thousands.

Contrary to indications made to me at Comdex/Fall, it now appears that Tandy will not involve itself soon in the laser printer market. The company *is* working on a new line of printers, which we may see this summer, but a laser printer is not among the offerings. When asked about the possibility of Tandy introducing a laser printer in the near future, Ed Juge, Tandy's director of market planning, said, "As far as I know, we are not seriously considering a laser printer now. Six or eight months ago we decided that the price was too high and that it was still a niche market."

Tandy has big plans for Express Or-der Hardware (EOH), the companion project to its Express Order Software (EOS) system. Though still in its infancy,

EOH already includes almost a dozen products that you can order through Radio Shack Computer Centers. According to Amy Arutt, the company plans to add more expansion/memory boards and tape back-up systems to the current list of EOH products. The company also hopes to make alternate keyboards available through the EOH system, she said. Tandy plans to publish the list of EOH products in its summer catalog.

Third-party support of Tandy laptop computers has received a boost from Travelling Software Inc. The Seattle company recently introduced two new operating systems— TS-DOS and LAP-DOS—that permit information sharing between a Tandy laptop and an IBM PC or a compatible. TS-DOS, a disk operating system for the Models 100, 200, and 600, formats information on the laptops' 3½-inch portable disk drives so that it can be read by IBM PCs or compatibles running LAP-DOS.

Travelling Software sells both operating systems, a cable for connecting the Tandy portable disk drive to a PC, and a file-sharing program called Exchanger for $89.95. With the Exchanger program, files generated on a PC with Lotus 1-2-3, dBase III, or SideKick can be accessed with the company's Ultimate ROM II integrated software chip.

MicroTrends

Educational applications are the fastest-growing segment of the home-computing market, according to a report by Link Resources of New York. From 1984–1985, the percentage of home computer users who ran educational applications on their micros grew from 45 percent to 64 percent.

The application with the second most significant increase is information filing and storage. In 1985, 59 percent of home users ran data base applications compared to 49 percent in 1984. Link predicts that 69 percent of home users will run data base applications on their micros in 1986.

Interest in entertainment programs has declined in the home market, according to Link, with 64 percent of users running entertainment packages in 1985 compared to 66 percent in 1984. Interest in budgeting and financial management applications, on the other hand, is increasing. In 1985, 36 percent of home users ran financial applications

and Link expects use to increase by 33 percent in 1986. Word processing use increased from 30 percent in 1984 to 39 percent in 1985 (see the Figure).

Commenting on the study, Andy Bose, Link's director of microcomputer analysis, said, "Collectively, the popularity of the home productivity and home busi-
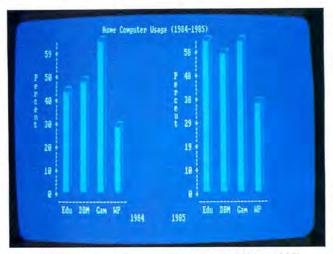
Figure. Home computer applications (1984–1985).

ness applications matches, if not exceeds, the education categories. This is not surprising, considering the heavy usage of the personal computer by heads of households."

Activision, a home software pro-ducer based in Mountain View, CA, and Infocom Inc., the number-one developer of interactive fiction, have signed a merger agreement to be finalized in June. Activision will exchange 2 million shares of its common stock for all outstanding shares of Infocom stock. The transaction has a cash value of approximately $7.5 million.

For Infocom, speculation that it was losing money trying to promote its Cornerstone data base program made the merger almost inevitable. At this time, Activision plans to allow Infocom to continue operating as a separate entity with product development and marketing activities based in Cambridge, MA.

As computer manufacturers and software publishers wonder what to do about soft sales, many are eyeing the educational market, where software sales are stronger than ever. During the 1985–1986 school year, software expenditures by America's schools (grades K–12) were $150 million, compared to $130 million in 1984–1985. The average number of new programs bought by schools went from 13 in 1984–1985 to 14 in 1985–1986, accounting for the 15 percent increase in sales.

According to Anne Wujcik, senior an-

alyst at New York's Talmis research firm, schools that have owned microcomputers for five years or more boosted their software purchases by as much as 25 percent during the past school year. Schools using micros for less than five years increased expenditures by 8 percent, while those with a year or less of computer experience decreased software budgets by 25 percent.

Answers to a Talmis survey sent to 5,000 junior and senior high schools across the country revealed a marked decrease in the numbers of entertainment and drill-and-practice programs being bought by schools. The move these days is towards applications programs in word processing, data base management, problem solving, and Logo programming.

Popular programs are Bank Street Writer from Broderbund Software Inc. and Scholastic Inc.; Apple Writer and Apple Logo from Apple Computer; PFS:File from Software Publishing Corp.; Master Type from Scarborough Systems; The Factory from Sunburst; Snooper Troops from Spinnaker Software; MECC Elementary Volumes from MECC Instructional Services; Typing Tutor from Microsoft; and Microzine from Scholastic Inc.

The results of the Talmis survey suggest that America's grade-school-level computer classes are beginning to move away from the superficial microcomputer applications that once predominated. Students today are being encouraged to use computers as learning tools that will help them after they leave school. But while teachers are increasingly leaning toward simulation and problem-solving software, the traditional drill-and-practice programs will no doubt continue to have a place in America's classrooms.

Face to Face

Tandy's president, John Roach, is slated as the keynote speaker at this year's PC Expo to be held in New York in July. According to the press release from PC Expo organizers, Roach will discuss PC options and directions applicable to corporate volume buyers and large resellers, and other industry trends. Says Roach, "I cannot promise a speech either extolling the best of times or lamenting the doldrums of doom, but [I do] hope to deliver an accurate and interesting commentary on the exciting and ever-changing industry." ■

PRINTER SALE

CANON
LASER PRINTER

$1999. *THE PRICE SAYS IT ALL.

LBP-8 A1

LBP-8 A2

THE STATE OF THE ART

The C.ITOH 3500 is fully IBM/Tandy 1000-1200-2000 compatible and FAST. Using bidirectional, logic-seeking printing across a full width of 13.2 inches, this little beauty prints at a speed of 350 characters per second (CPS). Letter quality printing is available at a rapid 87 CPS— more than twice the speed of most daisywheel printers. Couple that with a quick linefeed of 30 milliseconds per line and a standard buffer of 2K (16K optional, $50) and you have just about the fastest through-put around. Only 4.7″ high and a low noise level of 58 dBA.

Maximum versatility is offered for data and wordprocessing output, spreadsheets and business graphics by a variability of print densities, speeds, character sets and fonts, spacing, forms control and other attributes realized by operator and computer alterable functions, allowing the printer to be tailored to almost any application.

We could rave on and on about this fantastic printer but the proof is using this printer in your application. Try it on our 14 day money-back-if-not-satisfied plan. Don't be printer bound any longer. Call today.

350 CPS only $1295
(Please Specify Serial or Parallel)
(Shipping Included)

ORDER TODAY
CALL TOLL FREE **800-527-0347/800-442-1310**

MONTEZUMA MICRO

P.O. Box 32027
Redbird Airport, Hangar #8
Dallas, TX 75232
214-339-5104

MasterCard. VISA AMERICAN EXPRESS
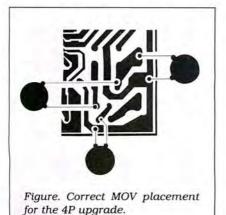
Making the Upgrade
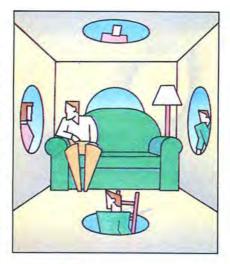
Two do-it-yourself modifications published in the January issue have prompted feedback. Author Joe Lindsey and reader Peter Nissen report that **souping up the MX-100** by replacing its 6 MHz and 8 MHz crystals with two 10 MHz crystals ("Speed Writing," p. 52) **can cause problems**. The printer works fine for normal printing, but in compressed and emphasized mode, it drops dots, creating characters of uneven density.

According to Tsun Tam ("On the Upgrade," p. 40), the **instructions for upgrading the 4P contained a few errors**. In Photos 8 and 9, the internal cables should show male, not female, connectors. On the external cable in Fig. 2, the pin labels for the two female edge connectors (right end of the diagram) were reversed.

In addition, the last physical drive should be terminated with a resistor pack so that the signal reaching the drive is the proper voltage. Finally, because blown MOVs short out, they should be installed after the fuse. The Figure below, which replaces Photo 6 of the original article, indicates proper MOV placement.

Although many 4P owners have successfully completed the upgrade, a few have had problems getting the disk drives to read double-sided disks. Tsun's upgrade requires that the drives be configured according to the industry standard—a standard that Radio Shack doesn't follow. To ensure proper operation of the drives, follow the directions for modifying the old cables carefully or purchase new internal (and external) cables.

Figure. Correct MOV placement for the 4P upgrade.

If you haven't found a source for the external drive extension cable mentioned in the article, try Alpha Products (79-04 Jamaica Ave., Woodhaven, NY 11421, 800-221-0916). They sell an 8-inch cable (catalog number C160) for $9.95.

From the Files

Model 4 owners with Epson printers who've been looking for a **screen dump routine to use with** *80 Micro's* **MacPaint-style graphics program** ("Drawing in Detail," September 1985, p. 56) are in luck. Dale Rogerson's routine for the Model III ("Rembrandt Redux," December 1985, p. 76) will run on the Model 4 without changes.

Raymond C. Boggs has passed on readers' **tips for using Finder** ("Basic Changes," November 1984, p. 56) **to manipulate Basic**. Dennis Lewis of Fresno, CA, tracked down the addresses required to use Finder with BasicG 1.1.0 (see the Program Listing).

One useful extra is Finder's ability to save a program if you accidentally return to TRSDOS with the System command. At TRSDOS Ready, type in FINDER; you will be returned to Basic with the program intact. Then type in Q at Finder's $ prompt. At the Basic Ready prompt, type in SAVE"file name:d". Immediately reboot the computer without trying to run your program. Just to be safe, you should carry out this procedure using a blank, formatted disk.

Boggs points out that TRSDOS 6.2 users are at an advantage when using Finder. They don't need to run the memory header printed in Reader Exchange in April 1985 (p.27). You can automatically insert a header by using MEMORY (HIGH = X'nnnn').

Setting the Record Straight

"Vital Statistics: Analyzing Data on Your Micro," listed the **wrong price for Walonick Associates' StatPac** statistics program (April 1986, p. 34). StatPac now sells for $495.

"Hard Decisions" (October 1985, p. 40) **contained an incorrect address for Western Digital**. The correct address is 2445 McCabe Way, Irvine, CA 92714, 714-863-7767.

Program Listing. BasicG 1.1.0 modifications for Finder.

```
****************BASIC ROUTINE ADDRESSES********************
ASCBIN   EQU    6011H            ;CONVERT ASCII TO BINARY
BINASC   EQU    3139H            ;BINARY TO ASCII AND PRINT
EDIT     EQU    3D2BH            ;BASIC LINE EDIT ROUTINE
EXPAND   EQU    7AA3H            ;BASIC LINE EXPAND ROUTINE
EXPAN1   EQU    7AA6H            ;SAME AS EXPAND, EXCEPT
                                 ;USER DEFINES BUFFER IN BC
KBLINE   EQU    974DH            ;KEYBOARD INPUT ROUTINE
PSTSRC   EQU    5AD6H            ;PST SEARCH ROUTINE
READY    EQU    5942H            ;1ST BASIC READY ADDRESS
READY1   EQU    595DH            ;2ND BASIC READY ADDRESS
VDLINE   EQU    7A9AH            ;PRINT LINE TERMINATED
                                 ;WITH BYTE OF 00H
;*************COMMUNICATIONS REGION ADDRESSES**************
BASBUF   EQU    70C5H            ;ADDRESS OF EXPANDED LINE
BASEND   EQU    722BH            ;END OF BASIC PROGRAM PTR
BASTRT   EQU    6F2AH            ;START ADDRESS OF PROGRAM
BINBUF   EQU    738BH            ;BINASC CONVERSION BUFFER
NOLINE   EQU    6C74H            ;NO SUCH LINE MESSAGE ADR
AUTOLIN  EQU    7214H            ;CURRENT AUTO LINE NUMBER
AUTOFLG  EQU    7213H            ;AUTO INCREMENT FLAG
AUTOINC  EQU    7216H            ;AUTO LINE INCREMENT
;*****ADDRESSES TO CHANGE TO TAKE CONTROL FROM BASIC*****
JUMP1    EQU    5985H
JUMP2    EQU    59B3H
OLDJMP   EQU    974DH            ;WHAT WAS IN THOSE AREAS
;*******************************************************
NUM      EQU    80H              ;CONSTANT FOR UNNEW
                                                        End
```

NEW!
CONV4TOPC
For TRS-80 Mod 4 users.
Plus other great utilities.

CONVERT
MOD I/III BASIC PROGRAMS and FILES
For Use On The
IBM PC, TANDY 1000, 1200HD, 2000

Here's time and money saving news for thousands of TRS-80 Mod I and Mod III owners who would love to move up to state-of-the-art hardware! EMSI's conversion package contains utilities to solve both problems facing those who want to upgrade:

PROBLEM 1—HOW DO I GET FROM HERE (Mod I/III) TO THERE (PC)? Do I need to retype everything, buy modems, RS232's, cables, and communications software?

ANSWER: None of the above! Use the HYPERCROSS utility included with our package! HYPERCROSS makes the entire disk transfer process very simple—and fast. All the work is done right on your Mod I/III. HYPERCROSS lets you format a diskette readable by all PC's in one drive and copy files directly to it from a Mod I/III diskette. After the transfer, take the PC diskette out of your Mod I/III and put it in your PC. Simple as that! (Mod I's need a doubler.)

PROBLEM 2—ONCE I'M THERE, HOW DO I CONVERT MY MOD I/III PROGRAM TO RUN ON A PC?

ANSWER: Use our CNV3TOPC utility to do 95% or more of the conversion for you. It automatically inserts all required spaces between keywords, replaces PRINT@'s (even those with variables) to LOCATE's, adjusts TAB addresses, corrects the exponentiation symbol, replaces the % symbol in USING statements with a backslash, removes down arrows, optionally removes REM's and flags and lists unresolved line numbers. It even allows for Mod I/III screen PEEKs and POKEs.

And, our thirty page user guide is packed with examples and hints showing how to make any manual program changes required after using CNV3TOPC.

"*. . .It's the best such program I've seen, well worth it's higher price over similar programs. . .The conversion program performed flawlessly.*"
Mr. Gary Shade, 80 MICRO, May 1985 (4½ stars).

"*Truly, a Superior Quality Software Package. Count me among your list of satisfied customers.*"
Waltham, MA

"*I would gladly recommend the package to anyone making the change to an IBM type machine.*"
Wauwatosa, WI

"*What a time saver. Thanks for a great product.*"
Denver, CO

"*Excellent! The manual alone. . .is worth the price.*"
Westport, CT

"*The fine points of conversion you cover in the manual are excellent.*"
Salem, OR

"*An excellent product. Thank you for the service.*"
Mapa, CA

"*Works like a charm! Congratulations.*"
Odenton, MD

CONV3TOPC V2.0 $139.95
(Package includes: HYPERCROSS and CNV3TOPC)
AVAILABLE WORLDWIDE through Radio Shack's Express Order Software (Cat.No. 90-0345)

Visit a R/S Computer Center and review the CONV3TOPC DEMO. It provides complete package details and will answer all your questions. If your store doesn't have the DEMO yet, ask them to order it thru ICST-FC# = FCO-925.

——— OTHER EMSI SOFTWARE ———
THE NORTON UTILITIES LIST $99.95, OUR PRICE $59.95

CONV3TOPC V2.0—(WITH Hypercross)	$139.95	**CONV3TOPC V2.0**—(WITHOUT Hypercross)	$119.95
CONV4TOPC V1.0—(WITH HYPERCROSS)	$139.95	**CONV4TOPC V1.0**—(WITHOUT HYPERCROSS)	$119.95
Same as CONV3TOPC V2.0, but		**CONV3TO4**—Mod I/III to Mod 4 BASIC	$49.95
specifically for mod 4 BASIC and FILES		**HYPERCROSS**—Mod 4, PC-DOS formats	$49.95
CONV3TOPC V2.0 DEMO (Runs on any PC)	$20.00	**ISAM ROUTINES**—Incorporate these routines	$69.95
HYPERCROSS—Mod I/III, PC-DOS format	$49.95	in your PC BASIC programs. They provide keyed	
CROSS REFERENCE—For PC BASIC	$24.95	access to random files & complete file maint.	
programs. Lists all referenced variables,		**RAMDISK**—Create superfast pseudo disk drive	$49.95
line numbers, etc.		(eg. create a 90K RAM drive C: and still have 60K for	
FASTSORT—Machine language SORT	$24.95	BASIC on 256K PC).	
callable from PC BASIC. Great PC subst.for Mod III		**INSIDE TRACK**—Over 60 PC utilities that	$44.95
BASIC's CMD"O" command.		compliment PEEKs 'n POKEs package. Too many	
PEEKs 'n POKEs—Over 50 utilities	$29.95	functions to mention.	
that enable PC BASIC programs to access and modify			
PC/MS-DOS system functions.			

800-922-0786
(NJ residents 201-879-5982)

EDUCATIONAL MICRO SYSTEMS, INC.
PO Box 471, Chester, New Jersey 07930

VISA

EMSI direct order terms: VISA, Mastercard, MO, check or COD. Add $3.00 shipping/handling. Add $1.90 for COD. Foreign or first class, add first class postage (package wt. 2¼ lbs.). NJ residents add 6% sales tax.

MasterCard

The Model 3000:
Tandy's Bundle of Muscle

by Dave Rowell

★ ★ ★ ★

The Tandy 3000 comes with 512K, a 1.2-megabyte disk drive, a built-in real-time clock, and a parallel/serial adapter. The 3000HD comes with an internal 20-megabyte hard disk. Tandy/Radio Shack, One Tandy Center, Fort Worth, TX 76102. The Model 3000 (catalog number 25-4001) is $2,599; the Model 3000HD (catalog number 25-4002) is $3,599.

The Tandy 3000 is, by all reports, a fully operational clone of the IBM PC AT. Tandy has trimmed a few nonessential AT features, speeded it up, and undercut the AT's premium price. The 3000 is clone enough that you don't have to buy software or hardware upgrades from Tandy when you can find a better deal elsewhere. You can have the best of both worlds: Tandy service and the IBM marketplace.

Tandy's made-in-Texas clone runs its 16-bit 80286 microprocessor at 8 MHz (megahertz), 33 percent faster than the AT's 6 MHz clock speed. The 3000's ROM BIOS is guaranteed 100 percent AT compatible by developer Phoenix Software. Tests done for Tandy by Future Computing back this claim; of 48 major PC software packages tested, four didn't run on the 3000. They don't run on the IBM PC AT, either.

The Equipment

The 3000 provides 10 expansion slots, seven full-length 16-bit AT-type slots and three (one of them half-length) 8-bit XT slots. That's one more of each type than on the AT. Like the AT, the 3000 comes with 512K RAM, a slim-line 1.2-megabyte 5¼-inch floppy disk drive, a serial/parallel port expansion board, a socket for an 80287 math coprocessor, and a battery-powered clock. The 3000's 84-key keyboard closely follows the AT layout. In addition, the 3000HD comes with a half-height, 20-megabyte internal hard drive.

The 3000 lacks the AT's lock-and-key mechanism, which physically secures the AT case and electronically locks up the keyboard. What may turn away some potential 3000HD prospects, however, is the relative slowness of the 20-megabyte hard drive. It's an IBM PC XT-type drive, not a voice-coil technology drive like the AT's. The older type of drive runs much faster on the 3000 than it would on an IBM XT, but not as fast as the AT's more expensive hard drive.

For most users, the 3000HD will feel exceedingly fast. The hard disk slow-down will be noticeable only with a few applications requiring lots of random file access. For handling large data bases or running the Unix operating system, Tandy will be coming out with a very fast 40-megabyte hard drive. And a few third-party, half-height, AT-fast hard

drives are available. If a fast hard drive really matters to you, add $500 to the 3000HD's price when comparing it to the AT. It may still seem a bargain.

Outside In

The hefty 3000 is solidly constructed of steel except for its stiff plastic front panel. The large main unit measures 6¹/₈ inches high and 19 inches wide by 18 inches deep. Tandy stayed with their usual off-white color for the 3000's textured case and keyboard. It's not an unattractive computer.

The slotted air intake occupies most of the front panel. Behind it, the fan draws cooling air through a filter. The square, orange reset button sits to the left, easy to reach but safely recessed. Below the reset button, two red lights indicate power and hard drive activity.

Openings for two floppy drives occupy the sunken area to the right of the intake grille. The 3000's 1.2-megabyte drive A occupies the top slot. Reach around the right side, towards the rear, and you'll find the power switch—just like on an IBM. After passing through the power supply, cooling air leaves the 3000 through two louvered areas at the right rear. The ac power cord plugs in below the louvers, as does the power cord for the 3000's monitor. You need only flip the 3000's power switch to turn on both computer and monitor. Covered expansion slot openings, numbered 1–10, fill the other half of the metal backside. The keyboard plugs into the center back slot.

The plastic keyboard is sturdy, but light enough for lap work. Two legs fold out easily and click into place to angle the keyboard up. Rubber pads at the bottom front prevent sliding. The six-foot coiled cord is thick and strong, with a straight section reaching out from the plug around to the side of the computer—a nice touch.

The 3000 uses the sensible AT key layout: 10 function keys in two columns on the left, and a numeric keypad to the

Photographs by Frank Cordelle

right with all the calculator operation keys except "/", which isn't far off. The arrow keys on 2, 4, 6, and 8 control the cursor until you toggle to number mode with the Num Lock key. Large enter and shift keys are within easy reach of the home row. You must stretch to reach the "|", "\", and backspace keys above the enter key, however.

LEDs (light-emitting diodes) indicate the state of Caps Lock, Num Lock, and Scroll Lock right on the keys rather than in a bank of lights above the numeric keypad as on the AT. The mysterious Sys Req key in the top row of the numeric keypad will have uses under Unix (it has a dedicated interrupt vector).

My 10 fingers agree that the 3000's keyboard possesses an easy, positive key action that brings out their typing abilities like no other keyboard they've used.

The 3000's power and video plugs and expansion slots.

Uncovered

Remove three screws from the back panel and the 3000's case easily slides forward from its steel frame, revealing space for a rank of full-sized expansion boards. A large, perforated metal cage at right rear houses the 175-watt power supply, rated at 2.1 amps maximum current drain. The 3000 gets by with less power than the AT's 192 watts partly because it uses 18 power-conserving 256K memory chips to achieve 512K, while the AT uses 72 64K chips. The 3000 can be set to use 120-volt or 240-volt power sources.

The metal drive compartment occupies the area in front of the power supply. There are arrangements for three internal half-height drives. The top bunk holds the 1.2-megabyte floppy drive. The 3000HD's half-height hard drive sits in the bottom. The middle compartment can hold either a floppy or hard drive.

To the left of the drive cage, a large 6.8-volt lithium battery attaches by velcro to the inner front metal panel behind the plastic facade. When the computer is off, the battery powers the 3000's CMOS RAM, which holds not only current date
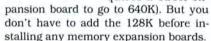
and time, but also hardware configuration information. When you add memory or hard drives, or change monitor type or the date and time, you must run the 3000's Setup program to update the CMOS RAM.

Over the right front area rests a clear plastic airflow covering, which channels the airflow through the power supply compartment.

The powerful (and fairly loud) fan sits at left front in a cast metal housing. A thermostat throws it into high speed at very high temperatures, but most users will never witness this phenomenon. A removable metal-fiber air filter slides up in front of the fan. It slides out the bottom of the frame when you undo its velcro fastener. The 3000's speaker occupies space to the left of the fan.

The motherboard under the fan holds 512K of RAM in two banks of nine 256K chips. The ninth chip in each bank implies that the 3000 uses the same parity memory-checking scheme as IBM computers. Behind the system RAM are two more banks of nine-chip sockets. They're for the 18 64K chips that bring RAM to 640K (the AT requires a 128K expansion board to go to 640K). But you don't have to add the 128K before installing any memory expansion boards.

Centered below the expansion board area is the central processing unit (CPU), the 80286 microprocessor. With a 16-bit data bus, improved instruction processing, and faster clock speed, the 3000 leaves the Tandy 1000/1200's 8088 CPU in a cloud of dust. The 80286 can also address 16 megabytes of RAM, while the 8088 is limited to 1 megabyte.

The empty socket in front of the 80286 is reserved for a 5-MHz 80287 math coprocessor (Tandy sells one for $400). The 3000's ROM is contained in two chips to the right of the 80286. Like the AT, the 3000 has two extra ROM sockets for future enhancements.

All 10 expansion slots have the 62-pin connector used for XT-type cards with 8-bit data bus. You can stick an XT expansion board (if AT compatible) in any of the 3000's 10 slots. The seven AT-type slots have an additional 36-pin plug-in, providing access to the full 16-bit data bus and 16-megabyte RAM addressing.

The half-length XT-type slot 10 is occupied with the serial/parallel adapter board. The parallel port is a female 25-

pin submini-D connector and accepts IBM printer cables. The pin assignments are pure IBM. You can use any printer designed for an IBM, or a Tandy printer that has an IBM mode. The 3000's serial port, like the AT's, is a 9-pin connector. The drive adapter board is in slot 9, and the video board (an XT-type) should go in slot 1 or 3.

Drive Options

The 3000HD drive controller card (Western Digital WD1002-WA2) supports two disk drives and two hard drives. The 3000 comes with a controller card that handles two floppies only. If you add a Tandy hard drive later, you must buy both a hard drive controller card ($500) and a drive ($800). That's $300 more than the price difference between the 3000 and 3000HD.

The 3000HD takes three internal drives, but all must be half-height units. You can, for instance, mount two internal hard drives in addition to the 1.2-megabyte floppy. Maximally configured, the 3000HD can run two internal floppies, an internal hard drive, and an external hard drive.

The 3000's Mitsubishi 1.2-megabyte floppy drive uses special 5¼-inch high-density disks ($59.95 for 10 from Tandy). The disk format is 80 track, double-sided, but with 15-sector tracks. At 360 rpm, the capacious floppy runs faster than the 360K drives (300 rpm), and its average access time of 91 milliseconds (ms) and track-to-track interval of 3 ms is quicker than some hard drives.

The high-density floppy drive can read and write in 360K format, but there are some restrictions. First, even if formatted to 360K, high-density disks can't be read by 360K drives because the magnetic signal strength is too high. Second, the 1.2-megabyte drive can read disks created by 360K drives with no problem, but you must take precautions writing to 360K disks with a 1.2-megabyte drive so they can be read by a 360K drive. The disk should be new or bulk-erased before formatting and the 360K drive must be well aligned.

The current 3000 version of MS-DOS 3.1 can't read or write to the 720K floppy format used by the Tandy 2000 (you can take a directory, but that's all). The next release of the DOS will probably add this capability (along with the ability to treat large hard drives as two logical drives).

The 3000HD's 20-megabyte hard drive, also made by Mitsubishi (MR522), uses four heads (two platters) to read 614 cylinders with 17 sectors per track. The 85 ms average access time and 18 ms track-to-track time are in the same class as an XT drive, but slower than the 52 ms and 14 ms ratings of an AT hard drive.

I ran several hard drive tests to get a feeling for the hard drive's capability. The Doran test (running Norton Utilities' Disktest program with the /D parameter) reads the whole hard drive sequentially, testing for read errors. The test took 328 seconds to read 20 megabytes with 16 disk buffers available—a rate of 63 kilobytes per sec (kb/s). An AT tests out around 70 kb/s. The PC XT 10-megabyte hard drive reads only 44 kb/s. The fast 80286 CPU allows the 3000 to use an interleave of three (i.e., the drive reads every third sector on a track and takes three disk revolutions to read a whole track). The XT uses an interleave of six, taking twice as long to read a track sequentially.

I also ran two test programs from *PC Magazine*. The first, called Bench01. EXE, performs random reads and writes and sequential reads on a large test file. The 3000's hard drive performed random writing and sequential reading at an equal or faster rate than an AT drive. Random reading revealed the slower head movement of the 3000's hard drive; it took three times as long as an AT with 512-byte records, but only 23 percent longer with 578-byte records.

The Core International test for drive speed (Bench06.EXE on *PC Magazine* benchmark disk) determines drive access speeds by averaging repeated read/write head movements. As many people have discovered, the 3000's hard drive fails miserably with the Core test, giving inconsistent access times much higher than the drive's rated speed. The cumulative test time for random access was so long as to overflow the test program's storage capacity, producing an obviously incorrect "30 percent faster than AT" rating.

It turns out that this stressful test uncovered a minor bug in the 3000 BIOS. According to Tandy, the bug shows only with the Core test and doesn't affect normal drive operation. The other hard disk tests bear this out. The BIOS ROM has been fixed and worked into current production.

What does all this mean? Benchmarks aside, the speed of the 3000's hard drive access depends on what you're using it for. For pure sequential file reading, the 3000 compares favorably with an AT. With pure random file I/O, it's slow. Most applications use sequential file I/O or use random access only sporadically, so drive access time isn't usually important. For large data bases doing sorts on disk or a Unix system using the virtual memory capabilities of the 80286, the 3000's hard drive becomes the limiting factor in getting the job done. It's for these uses that Tandy is bringing out a fast 40-megabyte hard drive.

Because the AT uses full-height drives, fast half-height drives are not common; in fact, they're just becoming available. The Tandon 755, for instance, is a new half-height, voice-coil, 40-megabyte drive that you can purchase by mail for $1,395 from BT Enterprises (10 Carlough Road, Bohemia, NY 11716, 800-645-1165). If you start with the single-floppy 3000, you'll need a new controller board, too. The Tandy card is $500. Because all ATs (even the floppy-only model) come with hard drive adapters, there isn't a big third-party market for them.

More Memory

The 80286 can address up to 16 megabytes of RAM, but not under MS-DOS 3.1, which normally can address only 640K. MS-DOS uses the 80286 compatibility mode so that you can run software written for a PC's 8088 chip. Any expansion above 640K can be used only as a RAM disk unless you buy a third-party memory board that follows one of the expanded memory standards. For instance, Intel's Above Board follows the Lotus/Intel/Microsoft extended memory standard and allows some software to address the extra RAM.

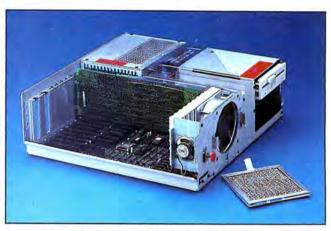This year, Tandy will come out with Microsoft's Xenix 5.0, their version of the Unix System V. This operating system runs the 80286 in its protected mode, allowing multitasking, use of virtual memory (1 gigabyte), and addressing the maximum 16 megabytes of RAM. Tandy plans to introduce a 2-megabyte expansion board in the second quarter of this year to make use of this capability. With the disk controller taking one 16-bit slot, six AT slots are left for 2-megabyte boards, so you can run 12.6 megabytes of RAM. The board comes with 512K for $499, and additional RAM from Tandy will come in sets of 256K (25-3062) for $220—an exorbitant $1,060 for 2 megabytes.

Video Electives

One decision you make when buying a Tandy 3000 is what type of monitor to buy. You can run anything from a TV to a high-resolution color graphics monitor. You must also purchase an adapter appropriate to your chosen display. Tandy has tested a wide range of PC and AT video adapters, and they're confident that the 3000 is totally IBM compatible.

The 3000HD I reviewed came with Tandy's Deluxe Graphics Display Adapter

($500) and their top-of-the-line CM-1 monitor ($599). Although capable of displaying 640- by 400-pixel *graphics, it* displays standard PC graphics (640 by 200) by scanning 200 lines twice. Text is as sharp as a good monochrome display, and the eight-color palette (16 is standard) is more than adequate for most applications software (missing are the paler colors produced with the intensity bit on RGBI monitors). The combination of clear text and sharp, bright color graphics explains why Tandy is selling so many 3000s with this monitor. This adapter, made by STB, also drives the VM-1 monochrome monitor ($100), too.

Tandy's Deluxe Text Display adapter ($250) drives the CM-1 and VM-1 monitors, producing high-resolution text, but

Inside: AT compatibility and expansions galore.

color only (on the CM-1) in text mode (colored text characters and background). In addition, the Tandy 1200's three adapter boards (mono, graphics, and master graphics) can be used in the 3000 (one must be patched when used with the VM-3 high-resolution monochrome monitor). These boards can be used with the monochrome and RGB monitors Tandy sells for the 1000 and 1200. With a fast machine like the 3000, you may want to run Digital Research's GEM or Microsoft's Windows in color. You'll need a monitor and a board that can handle the IBM's Enhanced Graphics Adapter (EGA) and get the 640 by 350 resolution in 16 colors. EGA boards and monitors are expensive; IBM equipment totals $1,500. Tandy is pushing to get their own EGA board out for the 3000.

Performance

The 3000 seems to perform in two modes: fast and instantaneous. I ran some computing benchmarks with known results on the AT. The Norton Utilities system information program (SYSINFO) rates the 3000's computing performance at 7.3 (times the IBM PC's). The IBM PC AT gets a 5.9. The 3000's advantage is 24 percent, not the 33 per-

The 3000 seems to perform in two modes: fast and instantaneous.

cent you'd predict just from the clock speed increase. A prime number calculation (Bench02.EXE from *PC Magazine*) ran only slightly faster on the 3000 (*14 seconds* versus the AT's 15 seconds).

I also ran some Basic speed tests (from *Byte* magazine, Vol.10:11). Two tested sequential file access by writing and reading 128K records to a 64K file. The 3000 read the file in the same time as the AT (24 seconds), and wrote the file slightly faster (25 seconds compared to the AT's 26). The 3000 performed the standard Sieve of Eratosthenes prime number program at the same speed as the AT (80 seconds), but it performed a single-precision calculation test quite a bit faster (24 seconds compared to the AT's 27). Keep in mind that differences between the two Basics may affect the comparison.

You should be able to go right out and buy software designed for the AT, although Tandy versions of MS-DOS software provide support for Tandy printers (and remove support for non-Tandy printers). Old programs tied to the PC's 4.77 MHz clock tick (usually copy-protected programs) won't run on an AT or 3000, and a few rare programs tied to the AT's 6 MHz clock might not run on the 3000. The 3000 doesn't have a 6 MHz mode like some AT clones.

Manuals

When you buy the 3000, you get a small *Installation and Operations Manual* and a utilities disk. The manual contains everything you need to know about setting up your new 3000, including the hard drive. Lots of well-drafted diagrams show clearly the appropriate circuit board placement of any jumpers used in hardware modifications. A jumper on the motherboard must be set for color if you use a color monitor, and another set when you bring total RAM on the motherboard to 640K. There are also two jumpers on the serial/parallel expansion board; one selects serial port 1 or 2, and the other switches the baud rate generator between USA and international settings.

The self-booting utilities disk provides a menu-driven set of programs for setting up the 3000. The disk contains a

low-level formatter for the hard drive and a Setup utility to program the CMOS RAM. Also provided are the Shiptrak program to park the hard drive read/write heads safely for transport, and simple format and disk-copy programs for the floppy drive.

Like the AT, the 3000 doesn't come with operating system software. You have to buy the MS-DOS 3.1/Basic/ DeskMate package from Tandy ($100).

The resident MS-DOS system files take 37K of RAM (not including Command. COM, which yields its space in high RAM to other programs if necessary). DOS 3.1 offers better support of hard drives than 2.11, which comes with the Model 1000. You can, for instance, run a program in another directory if you specify its full path name. DOS 3.1 also supplies a RAM disk (VDISK), a program to change disk volume names, and the ATTRIB command to change file attributes (read-only, hidden, and system). DOS 3.1, like 2.11, however, addresses only 1 megabyte of RAM, 640K of which is available for program use. VDISK can use any RAM above 640K.

Basic 3.11.00 (Tandy version) runs fast with an 80286 microprocessor. Functionally, it's the same as the AT's Basic (or BasicA), but isn't ROM-based. Several small programs I wrote using the graphics commands ran nicely on the CM-1 monitor and were extremely fast and sharp. In screen mode 1, the CM-1 limits you to eight colors; colors 8–15 come out the same as 0–7.

Unix Soon

Microsoft's Xenix 5.0 will soon be available for the 3000 from Tandy. One Tandy change will be the addition of device drivers for their Tape and Disk Cartridge Systems. DeskMate will also be part of the package.

Tandy has an expansion card in the works that will allow up to six people to use the 3000 under Xenix. It will provide four more serial ports and a Z80 microprocessor for improved performance. The AT, as IBM supports it, can handle three users (console and two serial ports). However, Xenix will not be a great system for very large applications because it must work around the 80286's 64K memory segmentation.

Conclusion

The Model 3000 is an excellent machine, and with a fast hard drive, it will be a great machine. Tandy has put more into this computer than any other: quality of construction, attractive design, and most of all, function. Tandy has made great strides over the last year with their MS-DOS machines, and the 3000 crowns Tandy's comeback. ∎

AFM: Freedom Of Information

by Harry Bee

★ ★ ★

AFM: Automatic File Manager runs on the Model III (48K) and requires two disk drives. PowerSoft Products, 17060 Dallas Pkwy., Suite 114, Dallas, TX 75248, 214-733-4475. $99.95.

Many data base managers are too strict. They force you into rigid data structures and demand that you modify your data to fit. AFM, however, is a free-wheeling data base manager that imposes no such restrictions. It's a filing system designed for those of us whose desire to get organized exceeds our discipline.

AFM is more difficult to learn than it should be and it displays some inconsistencies. Its help screens are less than illuminating and one command didn't work. In spite of all this, I like the program.

Easy Entry

To set up a data base under AFM, you supply names (called tabs) for the fields you think your records will contain. You don't have to decide how long each field will be or what sort of information you'll put into it. A field (for example, "phone:") may occur several times within a record, or not at all. The program allows 60 field names and can retrieve records by 26 of them.

An individual record is just that: individual. Rather than a cubbyhole of predetermined size and shape, it is a collection of related items as large or as small as necessary. The limit is a generous 4,096 characters per record.

AFM uses a "clipboard" of 64 continuous lines as a work area for most functions, and provides an adequate text editor. To build a data base, you type your data on the clipboard (including tabs) and use the editor to play with the format. When you're satisfied with a record, pressing the break key gets you out of the editor and another single-key command writes the record to disk. The program removes your data from the clipboard and leaves the field names in place as a form for the next record. But you're not locked into that format. You can add, delete, and rearrange fields for each record and maintain a library of data-entry forms to call as needed.

The program can also read ASCII text created by other data base programs or word processors directly into your files. I found this feature handy for gathering information from remote data bases by modem.

When you retrieve a record by enter-

ing a field name, it appears on the clipboard the way you wrote it. The format, though, is not set in cement. You are as free to edit as when you first created the record. As you page through records, options allow output to the printer or another disk file, and deletion. The Run command permits retrieval by fairly complex search criteria for continuous review, output, editing, or deleting.

A Disk Miser

With all this flexibility, you'd think AFM would gobble up your disk space. But the opposite is true: AFM stores only what each record contains, not unused fields or portions of fields.

For comparison, I transferred to AFM an address book I'd been keeping with Profile III+. In creating the AFM file, I was able to include all my data for each individual without having to make separate records to catch the overflow. I'd left room on Profile for only two phone numbers and one address in each record; with AFM it didn't matter. In addition, I was able to attach liberal notes to the AFM records (the names of children and dogs, birthdays, anniversaries) instead of being limited to the 32 bytes I'd allowed with Profile. AFM used about 20 percent less disk space, yet I felt I had more (and more useful) information in the new file.

The Puzzle Book

Forget that the 70-page user's guide is insensitive (librarians are "prim young ladies behind the desk"), puerile ("Isn't that nice?"), and noisy (!). It's inadequate. The flag-waving prose covers installation, setup, data entry, and simple retrieval well enough, but once you're ready for the program's more complex and powerful features, all you'll find are intriguing possibilities never explained. Nine commands listed in the manual's appendix do not appear in the manual proper.

The demonstration program supplied with the package is a hands-on guide to the same material in the manual and leaves the same puzzles. AFM's help screens are reminders only and shed little light.

Hidden Power

The real power of a filing system is its ability to return your data, sorted into an intelligible format. This is where AFM shines. There seems no end to the variety of reports it will produce or to the extent of the criteria you can search by.

To produce a columnar report, you enter a series of commands to establish its physical parameters, define the fields to include, set up your search mask, and run the report. Once you've puzzled out what everything is for, you can easily

run a report directly from the keyboard. However, it's no more trouble to use the template supplied with the program to create report generators, keep them on file, and run them when needed. You can also use AFM by itself to create serviceable form letters.

Although the syntax takes getting used to, you can set up separate search criteria for each field in your file, and AFM searches by all of them at once. Its Lookup feature uses data from one file to search another.

One operator deserves special attention. Preceding a search term with an apostrophe (') tells the program to search the entire text of a field for the given string. It's not perfect. The term '''commu'' *will* pull up ''communication'' or ''community,'' but also finds things like *covers modal music*. The possibilities of this feature, however, make AFM stand out among Model III data bases.

No Fatal Flaws, But. . .

The first time I ran AFM, I was greeted with an error message. In fact, there was no real error, and the condition disappeared after I set the program to execute automatically on bootup. This is typical of the anomalies that compound the manual's deficiencies. Nothing I tried crashed the program, but this sort of thing kept me wincing.

More serious are the inconsistencies. The search term ''>B<B'', used one way, will find items beginning with B. Used elsewhere, the term finds nothing; you have to change it to >B<C to get the same results. Another quirk shows up in sorting. The program is supposed to provide three distinct sorts. I found two, there being no discernible difference between the Standard and ASCII options. These two are more interesting than the Numeric sort, though. They shuffle numbers into the alphabet: ''2'' winds up between ''Q'' and ''R.''

The most serious flaw is the *SCAN command, which is supposed to let you recover otherwise unrecoverable records. *SCAN didn't work.

Conclusion

If you have the experience and the time to work around the documentation's shortcomings and the program's eccentricities, AFM turns out to be surprisingly easy to use. If the information you need to organize doesn't come in uniform packages and you feel as though regular data bases are just too difficult to work with, then AFM was made for you. Its flexibility and casual style, combined with its judicious use of disk space, allow applications that are not possible with other Model III data bases. ■

A Laptop Media Event

by Thomas L. Quindry

★ ★ ★ ★

The Portable Disk Drive works with the Models 100 and 200 and requires at least 8K RAM (more recommended). Tandy/Radio Shack, One Tandy Plaza, Fort Worth, TX 76102. Catalog number 26-3808. $199.95.

Finally, affordable data storage for the Models 100 and 200. Tandy's Portable Disk Drive is inexpensive, light (less than half the weight of a Model 100), and convenient to use. The drive will easily fit into a briefcase with a Model 100 or 200, a supply of disks, and an optional power supply.

The Facts

The Portable Disk Drive is a single-sided, 40-track, $3^{1}/_{2}$-inch drive that transfers data at 19,200 baud through the RS-232 port, rather than through the 40-pin expansion bus. It is powered by four AA alkaline batteries or an optional power supply. (The power supply is the same type that can power a Model 100 or 200 and costs only $5.95.)

While manufacturers of other disk drives for the 100 and 200 have chosen to use higher-capacity drives of 360K per disk, Tandy's smaller drive holds only 101K. But the difference in price makes up for the storage disadvantage.

Setting up the drive is easy. You should have at least 8K of RAM in your computer. Actually, 16K should probably be the minimum, since an 8K Model 100 only has about 5K of free RAM. When you use the drive for the first time, its four DIP switches have to be set to the on position—the drives should be delivered this way. You type in a three-line Basic program that loads the key program, Floppy.CO, from the supplied utility disk into your computer. After Floppy.CO is loaded, you set the four switches to the off position so that the file management program can be run. I did have trouble determining the on/off position of the switches since they aren't clearly marked.

The disk drive isn't much of a power grabber. A set of four alkaline batteries that would last 20 hours in a Model 100 will last about 10 hours in standby mode with the disk drive. The batteries will power about two hours of actual read/write operations. But two hours is a long time for disk input/output (I/O). It's rarely necessary to keep the drive on for more than a few seconds at a time; you

More Software from the Creators of Super Utility™

>> All products shipped from stock within 24 working hours! <<

SuperCross/XT

Easily Move Files from the TRS-80™
to MS-DOS™ or CP/M™ and back!

80-MICRO summed it up;"The value of this program far exceeds its price... if you work with different computers, it's a must."

See the 4-STAR Review in the Sept. 1985 issue

HUNDREDS OF ENTHUSIASTIC COMPLIMENTS ON FILE!
Don't be confused by competitors that are really BASIC translators with very limited file transfer capabilities!

SuperCROSS/XT will allow you to MOVE files back and forth between different operating systems, including PC/MS-DOS 1.x, 2.x/3.x (single or double-sided), CP/M+, or CP/M 2.2 on your TRS-80™ Model 4/4P, III, or I/DD. You can do this with your existing hardware and *SuperCROSS/XT*, eliminating modems, cables, and terminal program transfers. *SuperCROSS/XT* runs as a /CMD file under your TRS-80 operating system. Data files, spreadsheet files, and text files can be moved between machines. For example, years of Visicalc™ files, business letters, legal drafts, text files, PFS files, or medical records can now be utilized on your new machine!

Comments and letters on-file from registered users are unusually enthusiastic about this product and its ease of use. *It WILL do what you think it will do, it's easy, and it WORKS GREAT!* New features in SuperCROSS/XT include "tagging" files for multiple COPY's or KILL's to eliminate *many* unnecessary keystrokes!

CNVBASIC/CMD, available separately, "preps" your BASIC programs before sending over with SuperCROSS/XT. It will make most of the syntax and spacing changes required for converting Model I/III BASIC programs for use on GW-BASIC, CP/M BASIC, or Model 4 BASIC. Complex or commercial business packages written in BASIC probably will not convert 100% over by our *or any other* BASIC translator.

Important! Specify Model I/DD, III, or 4 computer type required.

SuperCross/XT (70 different disk formats)	$ 99.95
CNVBASIC/CMD alone	$ 29.95
*BOTH! • COMBO Special! - Save $10!	$119.95

* * * * *

Tandy 1000 and other PC compatible users:

Do you have problems with copy-protected disks on your Tandy 1000, 1200, 3000, IBM, or other PC compatible? *We don't mean trading software,* but making archival backups of your purchased MASTER disks and running your protected business applications from your hard disk. There is a program called *COPY II PC* that will handle these tasks for you. *COPY II PC* will backup most protected PC software. It also makes using your hard disk as convenient as it was designed to be! No longer will you need to use a "key disk" on most software (applications - not games). You can run directly from your hard disk with *NOKEY*, included with the *COPY II PC* package. Insert NOKEY in your BATCH file for fast, easy, automatic execution of your "protected" applications. This is sold for making archival backups of your own purchased software for your own use or running from your own hard drive only. *TANDY 1000 requires memory card because of needed DMA chip that resides there.* >> *Not for Tandy 2000* <<
Minimum requirements: TANDY 1000 (memory board required),1200, 3000, IBM™ PC™, XT™, AT™, or other "real" compatibles including Leading Edge, AT&T 6300, Compaq, etc. One or two disk drives and at least 128K required on most machines.

COPYII PC with Nokey - only $39.95

* * * * *

PowerSCRIPT 4.2 for Model 4, III, & I

Do you really like SCRIPSIT™, but wish it did a whole lot more? You'd like a lot of new features without having to relearn a whole new system or spending much additional money? Our PowerScript 4.2 is just the answer. *PowerScript 4.2* adds two categories of additions to SCRIPSIT™. Directory/File functions (DIR, FREE, KILL CHAIN, LINK), and embedded printer control functions. With all the smarter printers out there, it doesn't make any sense to use a "dumb" word-processor anymore. In this case, "dumb" means it can't really control your printer to its fullest capacity. If you notice, there are MANY word processors out there for the TRS-80™ besides the ones from Tandy. We could recommend any one of these, *IF YOU WANTED TO BUY SOMETHING NEW.* One of the problems, however, is that you need to learn all new commands. The second problem (maybe the main one) is that you need to shell out a lot more money to buy it! PowerScript 4.2 gives you many new features, while retaining the commands you already know - at a very low price! All your previous files will, of course, be compatible. *PowerScript 4.2* received a 4 1/2 star rating in *80-MICRO's* July '85 issue. Supports Mod I, III, and 4 versions of SCRIPSIT™ (Ownership of SCRIPSIT required).

Add all this power to your program for only $39.95!

Reference Manual for Scripsit™

New! Contains easy to follow instructions on getting the most from SCRIPSIT™ for Mod I, III, or 4! Explains all the mysteries and ends confusion on commands you never understood! Meant to supplement your manual, but will stand alone in its completeness. Contains extensive Index to the Tandy I/III manual. *only $7.95!*

* * * * *

SUPER UTILITY™ Combo Specials

Our ever-popular book *INSIDE SU+ 3.x* has been revised and renamed to now include all the changes that have occurred in the past two years since it was last revised. This perfect-bound, large format, slick 100 page+ book explains all the ins and outs of using these powerful utilities to the fullest of their capabilities. If you own *SU+ 3.x, SU4/4P,* or *PowerTOOL* this book will really add to your knowledge and maybe explain some things you didn't understand before. Lots of tips, hints, and suggestions are included, as well as helpful information for the novice as well as the "pro". Disk theory is explained as well. The book is now shipping, and a real value at only $19.95, its list price, but this month we're offering it for only $15! *Save $5!* If you are a new Model 4 owner and a new SU4 owner as well, then *NOW* is the time to get some great info!

Don't own *Super Utility* yet?? Buy our world-famous *SUPER UTILITY+ 3.2* or *SUPER UTILITY 4/4P* and add our new edition of *USING SUPER UTILITY* for only $5 extra! *Save $15! This program will amaze you. There is NO WAY to tell you in this space how powerful SUPER UTILITY really is, but literally thousands all over the world have written their praises of this program. Don't delay! You wouldn't believe how many of these we Federal Express to people who need it "yesterday"! Think of SU as data insurance. You can't go wrong.*

Super Utility+ 3.2 - for the TRS-80™ I, III, 4(III)	$79.95
Super Utility 4 - for the TRS-80™ Model 4/4P/4D	$79.95
USING SUPER UTILITY Book Sale price	$15.00

Special combo! SU with new book for only $84.95! Save $15!

Super Utility for the TRS-80 is "protected". It comes with two copies of the program, however. Upon registering, you may order an unprotected copy for $20 (U.S.) at your option.

Super Utility/PC for PC/MS-DOS - unprotected -	$89.95

* * * * *

Do you use a hard drive on your TRS-80?

We have a complete line of *Supreme Hard Disk Driver* packages that offer *much greater flexibility and superior performance* over the standard drivers supplied by Tandy or other hard drive manufacturers for the TRS-80 Mod 4, III, or I. They allow you to split up your hard drive into partitions of differing sizes, and also allow you to define granule sizes and other optimizing operating characteristics to achieve maximum use from your drive. The drivers (once relocated) are very small; less than 256 bytes. Most hard drive companies either sell or recommend our drivers. These drivers can also be used to allow a hard drive to be shared between LDOS 5.1 and TRSDOS™ 6 for Model 4 owners! If you have the 15 meg or 35 meg Tandy drive, you can now format it for full useage (Mod I, III or 4 mode)! *If you have a 4P, you can boot directly off the Tandy hard drive in Mod 4 mode without a disk!* Call or write for complete details. *only $99*

Whether you use our software driver or not, you NEED to use BACK/REST! Shame on you if you're not backing up your data!
>> See the 5-Star review in the Oct'85 issue! <<

If you've invested good money into a hard drive system, *it doesn't make any sense not to have a good BACKUP routine.* BACK/REST makes that job MUCH easier and saves hours. TIME IS MONEY. *Don't delay any longer!* BACK/REST will work with all hard drives that use LDOS™ or TRSDOS 6™ (all versions including 6.2.1).

Hard Drive COMBO Specials! Save Money!

BACK/REST 1.3 *for TRSDOS 6™ & LDOS*	$ 99.95
LDOS ToolBox	$ 49.95
Hard Disk Repair & Recovery Tools for LDOS - Mod I/III	
Model 4 ToolBelt	$ 49.95
Hard Disk Repair & Recover Tools for TRSDOS 6 - Mod 4, 4P, 4D	
Mod 4 ToolBelt and LDOS ToolBox Combo	$ 75.00
useful if your hard drive is split Mod 4 and III	
BACK/REST and your choice of TOOLBOX	
for LDOS or MOD 4 TOOLBELT	$139.95
BACK/REST and BOTH sets of tools	$159.95
BACK/REST and our *Supreme Rigid Driver* * (save $50!)	$149.95
*add either LDOS TOOLBOX or Model 4 ToolBelt for only $29.95 more!	

* * * * *

Fine print: Visa or MasterCard gladly accepted. Flat rate shipping on any item or combination of items from this ad is only $3 (UPS ground). Blue Label or overnight available at extra cost ($3 and $13.50). COD orders will have $2 added to the total. Canadian airmail is $4 and other foreign airmail will be actual cost. Foreign customers, please use charge card ONLY. Checks and orders not drawn on US banks will be returned. *Specials end 6-30-86.* Texas residents must add appropriate sales tax.
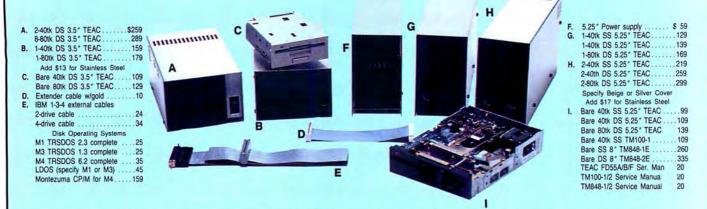
POWERSOFT

17060 Dallas Parkway, Suite 114
Dallas, TX 75248
(214) 733-4475

MasterCard   VISA

NEW FLOPPY DISK DRIVES
For Mod 1-3-4-1000, IBM

Aerocomp continues to lead the way to the BEST value in disk drives and related peripheral products for your computer. 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 selection and call our toll-free order number with your selection now. If you are not sure of what you need just call our technical assistance number and we'll help you out. All drives are new—not factory blems, seconds, closeouts or defunct manufacturers surplus (MPI, Qume, Shugart, etc). Instruction manuals are included at no extra cost and service manuals are available. We appreciate your business and will do our very best to support you.

Add $4 shipping for non-drive items; $6 for single drives; $10 for dual drives.
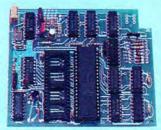
INCREASED DISK STORAGE FOR YOUR MODEL I
Add 80% more capacity to your disk drives with our Double Density Controller (DDC).

Add double density to your TRS-80 Model I by installing our DDC in your expansion interface. Lets talk about density. The Model I was designed to store data on diskettes in single density. Single density refers to the method used to write data to the disk. Your diskette is organized into tracks and sectors. Early Model I's had 35 track drives while later models, and most aftermarket drives, had 40 tracks. In single density the tracks on the diskette surface are divided into 10 sectors. Each sector contains 256 bytes of data for a total 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, allows each track to be divided into 18 sectors. As in single density each sector contains 256 bytes but now there are 18 sectors instead of 10 giving an new storage capacity of 4,608 bytes or 4.5k per track. The result is 80% more data in the same space. You may wonder why Radio Shack did not choose to use double density in the beginning. The reason is simple. It costs more money. Double density disk storage techniques were more expensive to implement back then.

Reliable double density operation required a better disk drive than Radio Shack was furnishing in addition to better quality components and diskettes. Therefore, no double density for the Model I. We went to work and came up with a design that allowed reliable double density operation on the Model I. In fact, our DDC had a higher probability of data recovery than any other disk controller on the market then or since. Our analog design phase lock loop data separator has a wider capture window than the digital types others use. This allows high resolution data centering. Our "DDC" analog circuit allows infinitely variable tuning with optimum attack and settling times. The oft-stated fears of adjustment problems rumored by digital dilettantes have been proved groundless by thousands of satisfied users the world over. The bottom line is state-of-the-art reliability and performance. 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 operation with far fewer disk I/O errors; single density compatability; simple plug-in operation. You will need a disk operating system that has the necessary double density software driver. All the popular DOS's (except TRSDOS) have the software driver. We have a special combination offer that saves you $10 and includes the LDOS operating system in the event you do not already have a DOS.

DDC by itself .... **$ 99**
DDC including the latest version LDOS **139**
Please add $5 shipping

LOW COST HARD DISK DRIVES FOR YOUR TRS-80

$ 495 & up
COMPLETE
5 Megabyte
Primary
Add a Secondary
for only
Add $20 shipping **$395** Add $30 for Stainless Steel

STAINLESS STEEL!

Our hard disk systems provide you with the latest integrated designs, proper controllers and pre-tested drives to ensure no unwelcome surprises. Our systems are engineered to provide years of trouble free service. These systems come complete ready to go to work. There is nothing else to purchase. Your choice of either CP/M, LDOS or TRSDOS software drivers at no charge. Additional drivers may be purchased for $30.

Each unit is guaranteed for one full year, parts and labor, at no additional cost. We provide the little things that are so important to a long troublefree life. Things like state-

of-the-art continuous duty switching power supplies; buffered seek drives; plated media; filtered forced-air ventilation for cool operation and extended life; power line EMI filter; solid steel construction; gold plated connectors; front panel LED indicators for Power—Ready—Select—Read—Write; built-in diagnostics; automatic error correction; provision to add a secondary drive; plus attention to details and a dedication to provide quality service that is unequaled. We are so sure you will be satisfied that we offer a 14 day

free trial. If you are unhappy with your hard drive, for any reason, just let us know within 14 days of your receipt and we will promptly refund your purchase price (less shipping). You can't go wrong. Start enjoying the real power and speed of your computer with one of our hard disk drives. Do it today! Use our toll-free ordering lines now.

ADDITIONAL SIZES AVAILABLE

	PRIMARY	SECONDARY
10 Megabyte	$699	$599
20 Megabyte	849	749
30 Megabyte	999	899

SAVE YOUR MONEY WITH OUR NEW INTERNAL HARD DISK DRIVES

When you shop and compare hard disk drives for your Tandy 1000 or IBM you will find the best values here. Our hard drive specialists put together a hard disk system that you can count on to give you years of dependable service.

These systems allow you to boot directly from the hard drive. The controller automatically goes into a self test diagnostics mode at power up that verifies optimum system performance. The latest in error checking and correction is built-in allowing extremely reliable handling of your data. Our installation kit comes complete with everything you need to install it into your computer. Complete illustrated instructions as well as all cables and software needed are included. All connectors and card edges are gold plated and keyed so correct assembly is assured with no possibility of damage to your computer, the controller or hard disk drive. These kits are warranted for one year including parts and labor. Our half-high drives have low power requirements making it unnecessary to modify or change your factory power supply. Model 1000 computers require a memory board (Tandy or other) with DMA. Use your standard PC or MS-DOS version 2.1 or later.

We promise your satisfaction with a 14 day money back guarantee. If, for any reason, you are unhappy with your hard drive just let us know within 14 days of receipt and we will arrange the return and a full refund of your purchase price (less shipping). Why wait? Call us now toll-free.

20 Megabyte . . . . $ 549
30 Megabyte 749
Add $10 shipping

ADD DISK DRIVES TO YOUR MODEL 3/4

Convert your cassette Model 3 or 4 to disk operation with one of our easy to install kits. Detailed illustrated instructions are included. All you need is a screwdriver and pair of pliers. Included is our own advanced controller with gold contacts capable of 4-drive operation; plated steel mounting towers complete with RFI shield; power supply plus all the cables and hardware. Choose a 1 or 2 drive system or the basic kit and pick the drives you want and your disk operating system from the list on the opposite page. Give us a call. We are ready to help with the answers to your questions.

Complete system, less drives . . . $ 159
1-Drive system 259
2-Drive system 349
Substitute DS drives for only $10 each.
Add $10 shipping.
Disk Controller Only 110
RS-232 Board complete 69   Add $4 shipping.

Pulling Strings

by Lewis Rosenfelder

One way to make your programs friendlier, faster, and more compact is to take advantage of Basic's Instring (INSTR) function. INSTR finds the position of one string within another, which can help you decode and validate keyboard input, filter or change unwanted data, perform program branching, and conduct high-speed searches of disk files. Once you master a few techniques for using INSTR, you'll find it to be a powerful and versatile tool.

Covering the Basics

Let's begin with a review of how INSTR works. Try typing in this command in Basic:

PRINT INSTR("MICROCOMPUTER","R")

When executed, this returns an answer of 4, since the first occurrence of the letter R is in the fourth position of the string MICROCOMPUTER. The string contains another R; to check for it, Basic must conduct a second search beginning one position beyond the first match:

PRINT INSTR(4 + 1,"MICROCOMPUTER","R")

The optional first argument (4 + 1) tells Basic to begin searching at position 5, resulting in an answer of 13.

Now try typing in:

PRINT INSTR(0,"MICROCOMPUTER","C")

Basic's versatile Instring function encourages program code that is fast, friendly, and concise.

This is an illegal function; the optional argument must be greater than zero.

Here's another example:

PRINT INSTR(13 + 1,"MICROCOMPUTER","R")

When executed, this command returns an answer of zero, Basic's response when it cannot find the search string. The search failed because there are no occurrences of R past the 13th position. Indeed, there are only 13 positions in the string.

If you were to enter the command:

PRINT INSTR("MICROCOMPUTER","X")

Basic would again answer with a zero, as there is no X in the searched string.

You can use INSTR to search for more than one character. Type in:

PRINT INSTR("MICROCOMPUTER", "COMPUTER")

In this case, Basic returns an answer of 6.

If a search string is null (" "), as in this example:

PRINT INSTR("MICROCOMPUTER"," ")

it "slips through the crack" at the first byte. Thus, the answer is 1.

Validating Input

Testing for valid keyboard input is the most common application of the Instring function. Type in and run these two lines:

10 LINE INPUT"Do you want to continue? ";A$
20 IF INSTR("YyYESyes",A$) THEN PRINT "Let's go!"

As desired, the If condition in line 20 tests true if a user types in a Y, y, YES, or yes, or if the enter key is pressed. However, it also tests true if the user types in an E, the letter sequence ES, or any other sequence that can be derived from the letters in the string.

To eliminate the possibility of Basic accepting invalid input, rewrite line 20 like this:

20 IF INSTR("/Y/y/YES/yes/","/" + A$ + "/")
    THEN PRINT "Let's go!"

Bracketing each valid option with slashes tells Basic exactly which letters and letter sequences to search for. If a user types in ES, INSTR searches for /ES/, which, of course, it won't find.

You can use the same string to show users which options are valid and to validate the choice. Try this technique:

10 VC$ = "RED,GREEN,BLUE"
20 PRINT "Valid choices are: ";VC$
30 LINE INPUT "Choose a color: ";A$
40 IF INSTR("," + VC$ + ",","," + A$ + ",") = 0
    THEN 20
50 PRINT A$;" is a good selection."

System Requirements

All systems
Basic

To change the options, all you need to do is modify line 10. Notice that I used commas instead of slashes to separate the options.

Option Strings

INSTR is also useful for programming yes/no choices. You can avoid the tedium of coding in questions like, "Do you want to do this?" or "Do you want to do that?" by assigning letter values to actions, as in Program Listing 1.

You might conceivably make this screen appear before a receivables printout. The user could obtain a double-spaced listing of accounts that are active but past due by typing in DAP, a single-spaced listing of inactive accounts by typing in I, and so on.

Use the Instring function to test for options. For example, to handle the possibility of a double-space request after the report line is printed, you might write the following:

230 IF INSTR(OP$,"D") THEN LPRINT " "

Here's how you would check whether the user wants active accounts only:

210 IF INSTR(OP$,"A") > 0 and B# = 0 THEN 240

If the condition tests true and the balance is zero, the program jumps to line 240. (When using the AND/OR operators with INSTR, be sure to use equal to or greater than zero, as I did in the test for A in line 210. This forces a true (– 1) or a false (zero) result, rather than a position.)

Converting to Numbers

INSTR works well for converting string responses to numbers, which computers handle more easily. Suppose you have a string of branch offices, coded ATL, BOS, CHI, CLE, DAL, DEN, DET. It might make your coding easier if you could refer to the branch offices as numbers 1–7. With the help of INSTR and a bit of math, Basic can make the conversions (see Program Listing 2).

The secret is keeping the codes the same length—four characters (including the comma) in this case. Let's say a user enters the branch office code CHI. Line 40 adds commas to both sides of the list, changes the search key to ",CHI," and finds a match in the ninth position. Line 50 subtracts 1 from 9 (to get 8), divides 8 by 4 (to get 2) and adds 1 to produce the answer, 3.

Another application of INSTR is for creating menu choices and program branching. For this application, use INSTR with an On. . .GOTO statement as shown in Program Listing 3. If the user types in A, control goes to line 100; W forces a branch to line 200; P forces a branch to line 300. If the operator presses any other key, including enter, the message "Bad choice" appears on-screen, and the program loops back to line 10. (Pressing enter results in

an empty quote or null string (" "), which, as I mentioned earlier, returns an answer of 1. A question mark will also return a 1.)

Simplifying Number Codes

INSTR also converts nonconsecutive numbers into consecutive ones for easier handling. An error-recovery routine is a perfect example. Suppose you anticipate four disk errors when writing a file:

    61 Disk full
    67 Too many files
    70 Disk write protect
    72 Disk media error

(These are error codes for the Tandy 1000, 1200, 2000, and 3000.) If one of these errors occurs, the error routine is called. You can convert the error code contained in the ERR variable to a number from 1–4 or a zero if it isn't one of the four expected. An On. . .GOTO statement then makes the program branch accordingly:

1200 A% = INSTR(" 61, 67, 70, 72,",STR$
    (ERR) + ",")
1210 IF A% THEN ON (A% – 1)/4 + 1 GOTO
    1310, 1320, 1330, 1340

Though I've used two lines for clarity, you can combine them. The STR$ function converts the error code to a string. Since error codes are always positive, a blank is always in the sign position. Accordingly, blanks appear to the left of each number in the string. For the math in line 1210 to work, each number, including its trailing comma, must take exactly four positions.

If ERR is 70, INSTR searches for " 70," and finds it at the ninth position. Line 1210 converts 9 to 3, forcing control to line 1330. An error code not in the list sends the program down to the next line, presumably 1220.

Filtering Data

Programmers who sort records alphabetically often store last names first in disk files. But what if you want first names to appear first in printouts and screen displays? Type in the following program to see how this might be done:

10 LINE INPUT "Last name, first: ";A$
20 A% = INSTR(A$,", ")
30 IF A% THEN A$ = MID$(A$,A% + 2) + " " +
    LEFT$(A$,A% – 1)
40 PRINT A$
50 END

Run the program and then type in JOHNSON, JERRY. INSTR looks for the comma and the space after it. Then line 30 changes the string to JERRY JOHNSON. If you enter ABC SUPPLY, it remains unchanged, since the name doesn't contain a comma and space together.

Sometimes you might want to filter characters from a string. Maybe your printer goes berserk when it receives a particular code, or maybe you want to alter data in a file you've downloaded over the phone line. The example in Program

Listing 4 reads a file called Oldfile and creates a new one called Newfile, in which all slashes are changed to dashes. Notice how INSTR reexecutes in line 140 if a given line contains more than one slash.

Indexing Disk Files

At some point or other, you've probably tried accessing random disk files in Basic and encountered obstacles. To access a record, Basic must know the record number. If you have an inventory file in which Widgets is the 340th record, you can't type in WIDGETS and expect Basic to find it. You must either search the entire file or perform some fancy programming to convert the search key to a record number.

Deleting records is another problem. Let's say you have an inventory file in which Buggy Whips is the 42nd record. Your company no longer manufactures buggy whips; therefore, you'd like to delete the record. Doing so creates a hole, which, ideally, you would fill with the next new product in the inventory file.

Keeping track of record deletions and available records at the end of a file can be a very confusing and slow process. Again, INSTR comes to the rescue. Index.BAS (see Program Listing 5) creates an index in Basic for accessing and managing random disk files.

All the search keys are stored in a string array. Instead of one string for each key, the program stores multiple keys in each string of up to 255 bytes. This means that if 42 6-byte keys are stored in each string, only nine Instring searches are required to find the 378th key!

The subroutine that begins at line 1000 dimensions the KA$ array according to the key length and index capacity specified in lines 1010 and 1020. It also contains a function that pads or truncates search keys that are shorter or longer than the indicated key length.

The subroutine at line 2000 performs the search. If you put the key in K$, it returns the record number in K%. If K% is zero, the key was not found. Otherwise, you can use K% with a Get statement to access your random disk file.

When INSTR (line 2020) finds a match, KP% is the position in the string. Line 2050 checks to see if the match is on an even key boundary. If it is, line 2070 converts the position to a key number. If the match isn't on an even key boundary, the search resumes at the next key.

If a key is found, check for duplicates by making K% zero, adding 1 to KP%, and performing a GOSUB to line 2020. K% returns the record number if a subsequent match is found.

To add a key to the index, put it in K$ and GOSUB 3000. On return, K% contains the record number the subroutine has chosen. If K% is not zero, use it with a Put statement to write the record to disk.

If K% is zero, the index is full and the record cannot be added.

To find the position for adding a new key, the subroutine at line 3000 searches for a blank position created by a previous deletion. If one is available, the routine at 3050 changes it to a new key. Otherwise, the key is added at the end of the index. The program does not check to see if the new key already exists; if you want all keys to be unique, do a GOSUB to line 2000 first. Then, if K% is zero, you can add it by performing a GOSUB to line 3000.

To delete a key, you must find it first by performing a GOSUB to line 2000. Then make K$ equal an empty quote (" ") and GOSUB to 3050. To change a key, you must first find it and then put the new value in K$ and GOSUB 3050.

I provided subroutine 4000 in case you have a record number and you need the key that goes with it. Put the record number in K% and GOSUB 4000. On return, K$ has the key. If K$ is null, the record number is too high. If K$ is all blanks, the record number represents a deleted key.

I've included a demonstration at the beginning of the program. Since it doesn't write anything to disk, you'll need to add disk routines for applications. After initializing the variables with GOSUB 1000,

load the index from a sequential file. For example:

```
700  OPEN "I",1,"ACCTINDX"
710  KI% = 0
720  IF EOF(1) THEN 740 ELSE LINE INPUT
     #1,KA$(KI%)
730  KI% = KI% + 1:GOTO 720
740  CLOSE 1:RETURN
```

Then open the random file to be indexed. Note that lines 700–740 assume the index already exists, so you'll need to create an empty file before the first run. Here's how you'd do it:

```
OPEN "O",1,"ACCTINDX":CLOSE 1
```

Before ending a program in which you've updated an index, be sure to close the random file and rewrite the index to disk as follows:

```
800  OPEN "O",1,"ACCTINDX"
810  KI% = 0
820  IF KA$(KI%) = " " THEN 840 ELSE PRINT
     #1,KA$(KI%)
830  KI% = KI% + 1:IF KI% < = KA% THEN 820
840  CLOSE 1:RETURN
```
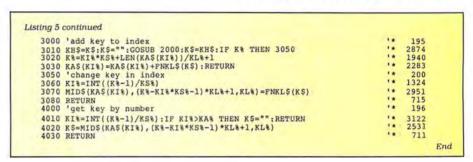
As you can see from these examples, knowing the position of one string is useful in a variety of applications. I hope the techniques I've covered will help you write better Basic programs. ∎

Lewis Rosenfelder is a software developer, author, and vice president of Tradewind Software, Honolulu, HI. His published books include Advanced BASIC Faster & Better *(Walnut, CA: Blue Cat Inc., 1985). You can reach him at 3026 Edwards Place, Riverside, CA 92503.*

```
Listing 5 continued
3000  'add key to index                                              '*   195
3010  KH$=K$:K$="":GOSUB 2000:K$=KH$:IF K% THEN 3050                  '*  2874
3020  K%=KI%*KS%+LEN(KA$(KI%))/KL%+1                                  '*  1940
3030  KA$(KI%)=KA$(KI%)+FNKL$(K$):RETURN                              '*  2283
3050  'change key in index                                           '*   200
3060  KI%=INT((K%-1)/KS%)                                             '*  1324
3070  MID$(KA$(KI%),(K%-KI%*KS%-1)*KL%+1,KL%)=FNKL$(K$)               '*  2951
3080  RETURN                                                         '*   715
4000  'get key by number                                             '*   196
4010  KI%=INT((K%-1)/KS%):IF KI%>KA% THEN K$="":RETURN                '*  3122
4020  K$=MID$(KA$(KI%),(K%-KI%*KS%-1)*KL%+1,KL%)                      '*  2531
4030  RETURN                                                         '*   711
                                                                        End
```

Truth AND Consequences

by Hardin Brothers

One feature of Pascal that Basic programmers often find attractive is its use of Boolean operators and functions. You don't have to learn Pascal, however, to take advantage of Boolean operations. Basic has most of the same logic capabilities as Pascal and other structured languages, though they are poorly documented and rarely discussed in books and articles.

Advanced programmers make frequent use of Boolean operations to simplify their programs, no matter what language they are written in. In this article, I'll explain how Boolean operations work in Basic and provide examples that I hope will lead you to more sophisticated programs.

First Principles

To begin, let's explore how Basic assigns true and false values. In Basic, type in the command:

PRINT (3 < 4)

This asks Basic to evaluate the truth of the statement "three is less than four." Basic responds by printing "−1" to the screen. Try typing in the command:

PRINT (3 > 4)

This time Basic responds by printing a zero to the screen.

The integers – 1 and zero are Basic's internal machine-level representations of true and false. When you ask Basic to evaluate the truth of a statement, it does so by assigning a value of – 1 or zero. In an If. . .Then or a While statement, Basic always evaluates the conditional portion first and assigns either a true (– 1) or false (zero) value before deciding what action to take.

As another experiment, type in the command:

IF – 1 THEN PRINT "True"

Basic responds (as you probably guessed) by printing "True." If you substitute a zero for the – 1, Basic prints nothing. Try other numbers after the conditional If and you'll discover another fact: Basic interprets any nonzero value as true, even though it always generates a – 1 when asked to evaluate a true statement.

The designers of Basic chose – 1 and zero to represent true and false because of the way computers store integers. Every integer requires 2 bytes (16 bits) of internal storage. Not surprisingly, zero is stored as 16 bits of zero. Its exact opposite, 16 bits of 1, is the internal representation of – 1. It makes sense to us to say "Not false is true." Basic and your computer agree with that logic. If you type in:

PRINT NOT 0

Basic returns the true value, – 1.

So What?

With this background, you are ready to put truth values (also called logical values or Boolean values) to work in a program. There will look at only a few.

Programs that handle sequential disk files often include routines that read an entire file into an array in memory. The Basic function EOF( ) returns a – 1 (true) when your program reaches the end of a file and a zero (false) when there is still more to read. The following routine is a simple method for reading an entire sequential file into the array A$:

```
WHILE NOT EOF(1)
    COUNT = COUNT + 1
    LINE INPUT#1, A$(COUNT)
WEND
```

(Some of the examples in this article, including the above, must be modified for use with Model I/III Basic. See the sidebar on p. 45 for changes.)

The expression NOT EOF(1) returns a true value as long as your program hasn't reached the end of the file. When it has, the value changes to false, and the program falls through to the line following Wend.

Another common routine is one that waits for a user to press one of a specified set of keys. For example, a menu might have

five choices numbered 1–5. The following routine makes the program wait for legitimate input after the menu is displayed:

```
OKAY = 0
OKAY.KEYS$ = "12345"
WHILE NOT OKAY
    KEY$ = INPUT$(1)
    OKAY = INSTR(OKAY.KEYS$,KEY$)
WEND
```

As the program fragments show, Boolean statements like While Not Okay and While Not EOF( ) are easy to read and understand; used correctly, they can make a program easier to read and debug.

Logical Functions

Boolean values are especially powerful when included in user-defined functions. I find this technique useful in data-oriented programs for testing the validity of user input.

Suppose, for example, you are writing a program that develops a list of phone numbers, which will be the basis of an automatic dialing program. The input loop for a phone number might look like this:

```
PHONE$ = " "
WHILE NOT FN VALID.PHONE(PHONE$)
    display a prompt
    get input in PHONE$
WEND
```

Notice that nothing in the five lines detracts from the main logic of the program, which is to keep asking for a phone number until the user types something the program considers valid.

You might decide that a valid phone number can have any of three forms: 555-1234, 800-555-1234, or (800) 555-1234. In this case, you must write a function— we'll call it FN Valid.Phone—that returns a true value only if a string conforms to one of the three forms. This is easy to do if you break the task into small steps.

First, call the three phone number forms Phone1, Phone2, and Phone3. Then write the function like this:

DEF FN VALID.PHONE(X$) = FN VALID. PHONE1(X$) OR FN VALID.PHONE2(X$) OR FN VALID.PHONE3(X$)

This defines a string as valid if it fits one of three special forms. If FN Valid.Phone1 or FN Valid.Phone2 or FN Valid.Phone3 returns a true value, then FN Valid.Phone will also be true.

The first format looks the easiest, so let's tackle it first:

DEF FN VALID.PHONE1(X$) = (LEN(X$) = 8) AND FN THREE.DIGITS(LEFT$(X$,3)) AND (MID$(X$,4,1) = "-") AND FN FOUR.DIGITS (RIGHT$(X$,4))

This function says that a string fits the first format if (and only if) it is eight characters long, the first three characters are numerals, the fourth character is a hyphen, and the last four characters are nu-

merals. The conditions are connected with AND operators because you want all four to be valid.

Next you'll want to check whether you have three or four digits in a row. One way to do this is to write three related functions as shown in Program Listing 1. Once you've written the code to test for one phone number format, you should have no trouble writing the code for the other two. Program Listing 2 shows how everything would fit together in a program.

If you look carefully at the code we've written so far, you may notice a way to save a great deal of space. FN Four.Digit could check one character and then use FN Three.Digit to check the other three. Both FN Valid.Phone2 and FN Valid.Phone3 could use FN Valid.Phone1 to check the last seven characters of a phone number.

Having functions call one another like this requires less typing and might improve program flow. Even so, I do not recommend this technique, for two reasons. First, Basic takes longer to evaluate multiple-linked functions. On a Model 4, the program in Listing 2 can evaluate the validity of a phone number in about a quarter of a second. It will take almost twice as long if the functions are more tightly linked.

Second, Basic uses a fair amount of stack space to store the partial results of one function while it calls another. If you link the functions in Listing 2 as tightly as possible, the Model 4 will send an "Out of memory" error message. You can overcome this problem by using the Clear command to allocate more stack space (CLEAR,,600 seems to work well), but you'll then run into a bug in the Basic interpreter. In addition to using more memory space than it should, the Clear command uses additional memory space every time you invoke it; it won't release that space until you leave Basic.

I originally wrote Listing 2 using tightly linked functions and added the command CLEAR,,600 in line 15. Then I began to correct typographical errors and test the program. After running it only a few times, I found that I had less than 5,000 bytes of memory left. To reclaim the memory confiscated by Clear, I had to save the program, return to TRSDOS, and reenter Basic. So much for programming elegance.

You should have little trouble adapting the programming techniques in Listing 2 to create routines that validate the format

System Requirements

All systems
Basic

In Basic, the operator NOT changes every bit in an integer into its opposite.

of Social Security numbers, dates, times, and other standard input formats. You can use FN Is.Digit as a guide to writing similar low-level functions that will check for alphabetic characters, punctuation, control characters, and so on. Higher functions can use these to determine whether user input fits the formats needed by a particular program.

Tied Up in NOTs

A technique I use often in Basic programs is to define a string of characters with an If. . .INSTR( ). . .Then construction. Let's say I have a string of characters called Special$. For checking user input, I'd include two lines similar to these:

```
KEY$ = INPUT$(1)
IF INSTR(SPECIAL$,KEY$) THEN . . .
```

The If. . .Then statement can branch to a help screen, error message, or any kind of special routine.

The advantage of using this construction is that it is easy to understand and it saves a small amount of processing time. The functionally equivalent statement If. . .INSTR( )<>0. . .Then forces Basic to convert the zero to binary form and then perform a comparison every time the line is interpreted. Since Basic interprets every nonzero value as true, the comparison is a waste of Basic's time and resources.
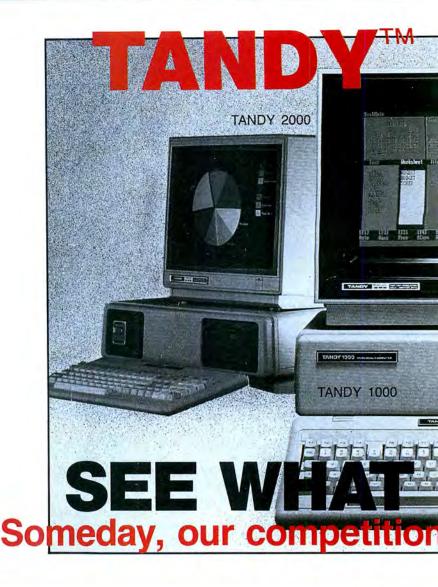
You might be tempted to use an opposite and related expression:

```
IF NOT INSTR(SPECIAL$,KEY$) THEN . . .
```

This will fail, however. Though the statement looks as if it should produce the opposite effect of the previous one, it will, in fact, always test true. Nothing a user types in will make Basic skip the Then clause.

```
A = B = 0
A = B = C = 0
A = B = C = D = 0
A = B = C = D = E = 0
A = B = C = D = E = F = 0
```

Figure. Test what you know about logic programming by determining the value of A in each expression.

The NOT operator is behind this paradox. Some languages (C, for example) have two NOT operators: one produces a logical negative and the other generates a bit-by-bit complement of an integer. In Basic, the NOT operator appears to be a logical negative, since it turns false values into true ones. It does so, however, only because of the bit patterns that make up the values for zero and -1.

In Basic, the NOT operator changes every bit in an integer into its opposite. False is composed of 16 bits of zero; true is composed of 16 bits of 1. Basic recognizes all nonzero values, including -1, as true. In the expression given above, if Key\$ is not contained in Special\$, then the value of the INSTR( ) function will be zero and the result of the expression NOT INSTR( ) will be the complement of zero, or -1.

On the other hand, if Key\$ is contained in Special\$, the result of the INSTR( ) function will be a number between one and the length of Special\$. The NOT of every positive integer is a negative integer, so NOT INSTR( ) produces a negative integer. Basic sees the negative integer as a nonzero (or true) value and therefore branches to the statement in the Then clause.

The moral of this is to be careful when using operators such as AND, OR, and NOT in logical expressions. All are really bit-wise Boolean operators. They do what you expect if you limit their operands to

Model I/III Modifications

Several examples on p. 44 of this article contain While. . .Wend statements, which Model I/III users must convert to If. . .Then statements. Also, Model I/III users must limit variable names to one or two characters.

The routine for reading a sequential file into array A\$ should appear as follows:

```
10 IF NOT EOF(1) THEN C = C + 1:
   LINE INPUT#1, A$(C):GOTO 10
```

The routine that waits for legitimate user input should be coded like this:

```
10 OK = 0: OK$ = "12345"
20 I$ = INKEY$:IF I$ = " " THEN 20
30 OK = INSTR(OK$,I$)
40 IF NOT OK THEN 20
```

And the input routine for a phone number should look like this:

```
10 P$ = " "
20 IF NOT FNV(P$)THEN
      display a prompt
      :get input in P$
      :GOTO 20
```

—Beverly Woodbury
80 Micro Technical Editor

zero and − 1. If you use them to manipulate other true values, however, your program might operate erratically.

On Your Own

At first, incorporating Basic's logic functions into everyday programs may require extra effort and thought. But if you keep at it and debug carefully to catch mistakes early, you'll eventually be able to program faster and move from Basic to other languages with less difficulty.

You'll also find that your finished programs require less debugging and the process will be easier. Logic functions are not a programmer's panacea, but they can help solve some difficult programming problems.

To test what you've learned about logic programming, I'll leave you with the puzzle in the Figure. See if you can determine the value of A in each of the expressions. Assume that all variables have been set to zero with a Clear command before each line.

I won't tell you the answers (your computer can do that for you), but you should see a pattern develop that will help you understand both this and other logic problems.■

You can contact Hardin Brothers through CompuServe. GoPCS-117 to the Writers' and Editors' SIG (WESIG). Or write to Hardin at 280 N. Campus Ave., Upland, CA 91786. Enclose a stamped, self-addressed envelope for a reply.

Program Listing 1. Three functions to test whether you have three or four digits in a row .

```
DEF FN IS.DIGIT(X$) = (X$=>"0") AND (X$<="9")

DEF FN THREE.DIGITS(X$) = (LEN(X$)=3) AND
    FN IS.DIGIT(LEFT$(X$,1)) AND
    FN IS.DIGIT(MID$(X$,2,1)) AND
    FN IS.DIGIT(RIGHT$(X$,1))

DEF FN FOUR.DIGITS(X$) = (LEN(X$)=4) AND
    FN IS.DIGIT(LEFT$(X$,1)) AND
    FN IS.DIGIT(MID$(X$,2,1)) AND
    FN IS.DIGIT(MID$(X$,3,1)) AND
    FN IS.DIGIT(MID$(X$,4,1))
```
                                                                        End

Program Listing 2. Program to test for valid phone numbers.

```
10 DEFINT A - Z
20 '
30 '        Phone number functions
40 '
45 '     Test for one digit
50 DEF FN IS.DIGIT(X$) = (X$=>"0") AND (X$<="9")
53 '
55 '     Test for 3 consecutive digits
60 DEF FN THREE.DIGITS(X$) = (LEN(X$) = 3) AND
    FN IS.DIGIT(LEFT$(X$,1)) AND FN IS.DIGIT(MID$(X$,2,1)) AND
    FN IS.DIGIT(RIGHT$(X$,1))
63 '
65 '     Test for 4 consecutive digits
70 DEF FN FOUR.DIGITS(X$) = (LEN(X$) = 4)  AND
    FN IS.DIGIT(LEFT$(X$,1)) AND FN IS.DIGIT(MID$(X$,2,1)) AND
    FN IS.DIGIT(MID$(X$,3,1)) AND FN IS.DIGIT(MID$(X$,4,1))
73 '
75 '     Test for format: 555-1234
80 DEF FN VALID.PHONE1(X$) = (LEN(X$) = 8) AND
    FN THREE.DIGITS(LEFT$(X$,3)) AND (MID$(X$,4,1) = "-") AND
    FN FOUR.DIGITS(RIGHT$(X$,4))
83 '
85 '     Test for format: 800-555-1234
90 DEF FN VALID.PHONE2(X$) = (LEN(X$) = 12) AND
    FN THREE.DIGITS(LEFT$(X$,3)) AND (MID$(X$,4,1) = "-") AND
    FN THREE.DIGITS(MID$(X$,5,3)) AND (MID$(X$,8,1) = "-") AND
    FN FOUR.DIGITS(RIGHT$(X$,4))
93 '
95 '     Test for format: (800) 555-1234
100 DEF FN VALID.PHONE3(X$) = (LEN(X$) = 14) AND
    (LEFT$(X$,1)="(") AND FN THREE.DIGITS(MID$(X$,2,3)) AND
    (MID$(X$,5,2) = ") ") AND FN THREE.DIGITS(MID$(X$,7,3)) AND
    (MID$(X$,10,1) = "-") AND FN FOUR.DIGITS(RIGHT$(X$,4))
103 '
105 '     Test for any valid format
110 DEF FN VALID.PHONE(X$) = FN VALID.PHONE1(X$) OR
    FN VALID.PHONE2(X$) OR FN VALID.PHONE3(X$)
120 '
990 '     Test loop for phone functions
991 '     NOTE: If Basic reports an error in one of the following
992 '           lines, the error may actually be in one of the
993 '           above functions.
994 '
1000 CLS
1010 PHONE$ = ""
1020 WHILE PHONE$ <> "END"
1030 INPUT "Type in a phone number to test ==> ", PHONE$
1040 IF FN VALID.PHONE(PHONE$) THEN PRINT "VALID" ELSE PRINT "INVALID"
1050 WEND
```
                                                                        End

Chain Links

by Bruce Tonkin

f you program on a Model 4 or one of Tandy's MS-DOS computers, you probably feel cramped in the skimpy space Basic allows for your programs. On an MS-DOS machine with 640K of memory, you'd expect to have more than 60K for your programs. If you use a 64K Model 4 or a 128K Model 1000, you're likely to have less than 30K. What happened to the good old days when Basic took up only 8K?

Life would be much easier and your programs better documented if you had more elbow room. It would be nice, for instance, if you didn't have to worry about the space remarks take up. The more remarks, the easier a program is to debug or enhance six months later. For the same reason, it would help to have long, descriptive variable names and error messages that make sense. It would be even better to write an application as one enormous program composed of many modules.

But if you're like most programmers, you're convinced you can't do these things: there's just not enough memory. So, you write programs with tiny error messages, or none at all. You remove all remarks or put them into a separate file for reference. (Your separate file, of course, is never up-to-date. If you make the idle mistake of renumbering the original program, your documentation turns into garbage.)

You might also use compression routines to squeeze air out of your programs. The result is generally unreadable, sometimes unprintable, and often unlistable. When it crashes, the disaster is completely unmanageable.

You might use cross-reference programs to look for variable names and replace every long one with a single letter. You trace the program to see which variables are needed where, eliminate new ones when possible, and reuse the old ones. This makes for new disasters when you run the result. But you may save as many as a hundred bytes!

In short, the programs you write are typical microcomputer Basic programs: difficult to maintain, hard to debug, and limited in features by available memory. Structurally, they resemble a pile of jackstraws held together with string, paste, and spit.

Just as garlic oil is supposed to repel vampires, so computer science instructors believe a little oil of Pascal will repel programs like this. Fortunately, there's a better solution.

Chain Gang

If you use the Microsoft Basic that comes with all Tandy MS-DOS computers, the Model 4, and with CP/M, you may be familiar with the Chain command. The purpose of the command is simple: it allows you to run another program, saving variables in memory. Files remain open for use in the program to which you've chained. All you stand to lose are user-defined functions and type definitions (unless you use the Merge option). The syntax of the Chain command is:

CHAIN file name [,line][,ALL]

The file name can be any legal Basic program name. (For use under the interpreter, however, the program chained to must not be saved as an ASCII text file unless you use the Merge option.) The optional line number can be a variable, and the line chained to must exist in the program to which you are chaining. The All parameter specifies that all variables will be retained.

Like subroutines, chaining gives programs structure and makes them easier to debug. If you've tried chaining, however, you know it is different from running a subroutine in at least one important respect: once you've chained from the main program to another, you can't truly return without using the Merge command. Merges let you overlay the lines of the chained program on the main program, maintaining user-defined functions and type definitions. There are penalties involved, however; you must reserve certain line numbers throughout your programs, your program must be saved as an ASCII text file, and you can't compile.

But though Basic has no Chain return, there is a way to create one. The secret lies in the Chain statement itself: the fact that the line number can be a variable means you can save it, compute it, or massage it in a number of interesting ways.

Let's say you were somehow able to save the current line number and program name. You could then chain to a subprogram and return to that line number or one before or after. By subprogram, I mean one that runs under the control of the main program while existing separately on disk or in memory. It functions much like a subroutine, except that it has all of your Basic memory space available to it.

All that prevents you from chaining to a subprogram is a means of saving and using the current line number. I know of only one way to determine the current line number in Basic: force an error with the Error command. This works regardless of how the program is later renumbered. The Basic interpreter saves the current line number and the current error code in the special reserved variables ERL and ERR. You can save these and use them to control how and where to chain.

Program Listing 1 shows how sample code might look in the main program (listings begin on p. 51). X% is the number of an available file buffer and Current% is a flag variable. The name of the current program (up to 14 characters in length) is saved in the variable Program.Name$. The name of the subprogram to execute is saved in the variable Go.Program$. The variable XX holds the current line number.

The subroutine uses PN$ and LN$ to save the name of the current program and the name of the line number to execute. These values are saved in the file called TEMP. The line number is saved as a single-precision value, since program lines can exceed 32767.

The subprogram begins executing at line 10, as you can see in line 50060. To return, you'd execute code similar to that in Program Listing 2. The subprogram returns to the same line of the main program that was last executed (before the routine at line 50000). In this example, the program returns to line 10010. However, the subprogram has changed the value of the variable Current% to zero. That means line 10020 will execute, since the conditional If in line 10010 of the main program no longer tests true.

In effect, the chaining program creates a "stack" on the disk, which saves the name of the program that originated the subprogram call and the line number to which program control should transfer. You can manipulate the stack and alter the record length in various ways. On MS-DOS machines, you may want to use more than 14 characters to save the complete

name of the Basic program. MS-DOS names may include a path specification of almost arbitrary length. (In practice, even under MS-DOS, a path specification length of 25 or 30 characters should be adequate.) You may also use a variable to indicate which line number to chain to in the subprogram.

Program Listings 3 and 4 are sample programs that you should be able to run on your computer. Listing 3 contains a main program, ThisCode.BAS. Listing 4 contains a subprogram saved as That-Code.BAS. (Model 4 file names would be written with a slash instead of a period before the extension.) I've changed the number of the file buffer from 4 to 1 and added a small amount of executable code so you can watch the program execute.

You may be surprised to see just how rapidly these programs run. If you don't care about power outages, you can usually put one or more of the programs on a RAM disk for even greater speed. Overall, the performance penalty is probably far less than you'd expect.

Basic Advantages

The chaining technique I've described offers numerous advantages. Here is a partial list:
- All programs and subprograms are reentrant. A program can chain to itself at any line.
- You can name chained-to programs by function; in effect, you're able to replace line numbers with labels up to 11 characters long.
- You can save commonly performed functions as modules and use them with many programs.
- You can alter the normal return sequence under program control. Unlike a normal GOSUB/Return, which can't be abnormally exited without leaving unnecessary values on the stack, this technique places the stack under the programmer's control.
- You can save variables in another disk file. The current program state (line executing and program name) can be saved at the top of the stack at any point. Therefore, you can halt a program and resume at another time.

If all programs are controlled through a master program, a minor modification allows the master program to determine which program was last executing. Thus a self-booting disk could resume processing without loss of data if you were to lose power while a program is executing.

System Requirements

Model 4, 1000, 1200, 2000, 3000, or CP/M-based computer Basic

---

- You can save variables selectively by using a Common statement and not using the All option in the Chain command. This allows you to have local and global variables in the subprograms. Variables placed in a Common statement will be global; those not placed in a Common statement will be local. Even without using the disk file to save the program name, a program could chain to itself at any point and wipe out variables no longer needed. Under the compiler, chaining with a Common statement is necessary because the All option is not supported.
- The technique does not work well with poorly structured code, sometimes called spaghetti code. Thus, it forces you into the good habit of writing modular programs.

Even if the subprogram contains many modules, you can use self-explanatory variable names and values to simplify debugging and make program functions more clear. You might, for example, write something like the code in Program Listing 5. By controlling the line number at which the subprogram is entered, line 10010 precisely describes the action anticipated in the subprogram. The general subroutine is the same as in Listing 1, except for line 50060, which now directs control to the line specified by the variable Action Code.

Chaining eliminates virtually all memory limitations due to program size. If you use large arrays, you may still be limited by memory, but any reasonably modular program can be run, given sufficient disk space. You can use as many remarks as you like and variable names of whatever length you like. You will not need to compress programs. Contrast this with a typical microcomputer Basic program!■

Bruce Tonkin's column, The Art of Programming, debuted in the May issue of 80 Micro. *You can reach Bruce at 34069 Hainesville Road, Round Lake, IL 60073 312-223-8595.*

Yes, but...

All programming techniques have shortcomings, and chaining is no exception. Even so, many of the disadvantages you'll encounter with chaining are easy to overcome. And the overall benefits outweigh the small extra effort you'll put in to make the technique work well.

Following are some common questions, comments, and concerns about the chaining technique and my responses.

With chaining, won't you have a far more difficult job of cross-referencing an entire program and controlling all variables used?

This may be true, but you'll be able to control all the variables used in each module. That's more important, especially when you have many modules shared among a number of programs. In time, you'll develop a standardized set of variable names for equivalent purposes in all your programs.

Doesn't chaining make programs run more slowly?

Which would you rather have: a program that runs slowly, or one that doesn't run at all? Besides, "slowly" is a state of mind. If your modules are about the right size, it will take only several seconds to chain from one to another. It's likely that it will take far more time to execute the code in the module than it will to chain from module to module. With planning, you can minimize the difference in running times between modular programs and equivalent ones that don't use chaining.

Since one file buffer must always be reserved for controlling the disk "stack" allocation, won't you lose a file buffer?

The extra buffer is lost only briefly while it is being used. The only real penalty is that the particular buffer used to control the stack must not be used for another file when it is needed to control the stack.

If you want to increase the speed of moving from one program to another, you can leave the file buffer open at all times; this puts more restrictions on all the programs and subprograms, however. For one thing, you won't be able to write unconditional Close statements.

Also, it doesn't really matter which file buffer is open for use: it might be buffer number 1 one time, and buffer number 2 another. One way to lessen the penalty is to write the routine so that it uses the first available buffer instead of a fixed one. The Program Listing contains some sample code using this method. Lines 50010–50030 form a loop. The variable X% is incremented until line 50020 executes successfully or until X% becomes too large.

The only real problem with this method is that too few file buffers or a bad file name will make the routine "fall off the end." In the sample code I've given, I made the program print the message "Darn it!" when this happens.

Won't you lose your defined functions when you return from the subprogram?

Yes, but all you need to do is put the user-defined functions in a block at the beginning of each program or subprogram. Make these definitions occur as a subroutine, and call them whenever you return from the subprogram or enter a subprogram for the first time.

This method makes your program more modular and more compatible with the IBM or Microsoft Basic compiler. The compiler doesn't recognize function or type definitions until it encounters them in the program. If you try to compile a program for which a user-defined function appears before it is defined in the source code, you'll get an error. Putting all functions at the beginning ensures that this will never happen.

—*Bruce Tonkin*

Program Listing. Routine to use first available buffer.

```
10000 CURRENT%=1:PROGRAM.NAME$="THISCODE.BAS":ON ERROR GOTO
50000
10010 IF CURRENT% THEN GO.PROGRAM$="THATCODE.BAS":ERROR 255
10020 'More executable code appears here.
 .
 .
50000 X%=1:XX=ERL:RESUME 50010
50010 ON ERROR GOTO 50030
50020 OPEN"R",X%,"TEMP",18:GOTO 50040
50030 X%=X%+1:IF X%<15 THEN RESUME 50010:ELSE PRINT"Darn
it!":END
50040 FIELD X%,14 AS PN$,4 AS LN$
50050 GET X%,1:REC=VAL(PN$):IF REC=0 THEN REC=1
50060 REC=REC+1:LSET PN$=STR$(REC):PUT X%,1
50070 LSET PN$=PROGRAM.NAME$:LSET LN$=MKS$(XX):PUT
X%,REC:CLOSE X%
50080 CHAIN GO.PROGRAM$,10,ALL
```

End

Program Listing 1. Sample main program code.

```
10000 X%=4:CURRENT%=1:PROGRAM.NAME$="THISCODE.BAS":ON ERROR
GOTO 50000
10010 IF CURRENT% THEN GO.PROGRAM$="THATCODE.BAS":ERROR 255
10020 'More code follows; will be executed following the
return
  .
  .
  .
50000 XX=ERL:RESUME 50010
50010 OPEN"R",X%,"TEMP",18
50020 FIELD X%,14 AS PN$,4 AS LN$
50030 GET X%,1:REC=VAL(PN$):IF REC=0 THEN REC=1
50040 REC=REC+1:LSET PN$=STR$(REC):PUT X%,1
50050 LSET PN$=PROGRAM.NAME$:LSET LN$=MKS$(XX):PUT
X%,REC:CLOSE X%
50060 CHAIN GO.PROGRAM$,10,ALL
```
                                                                    End

Program Listing 2. Sample code to return from the subprogram.

```
10000 X%=4:CURRENT%=0
10010 OPEN"R",X%,"TEMP",18
10020 FIELD X%,14 AS PN$,4 AS LN$,1
10030 GET X%,1:REC=VAL(PN$):REC=REC-1:LSET PN$=STR$(REC):PUT
X%,1
10040 GET X%,REC+1:XX=CVS(LN$):PN$=PN$+"":CLOSE X%
10050 CHAIN PN$,XX,ALL
```
                                                                    End

Program Listing 3. ThisCode.BAS.

```
10000 X%=1:CURRENT%=1:PROGRAM.NAME$="THISCODE.BAS":ON ERROR
GOTO 50000
10010 IF CURRENT% THEN GO.PROGRAM$="THATCODE.BAS":ERROR 255
10020 COUNT=COUNT+1
10030 PRINT"Returned from subprogram; loop execution
number";COUNT
10040 GOTO 10000
50000 XX=ERL:RESUME 50010
50010 OPEN"R",X%,"TEMP",18
50020 FIELD X%,14 AS PN$,4 AS LN$
50030 GET X%,1:REC=VAL(PN$):IF REC=0 THEN REC=1
50040 REC=REC+1:LSET PN$=STR$(REC):PUT X%,1
50050 LSET PN$=PROGRAM.NAME$:LSET LN$=MKS$(XX):PUT
X%,REC:CLOSE X%
50060 CHAIN GO.PROGRAM$,10,ALL
```
                                                                    End

Program Listing 4. ThatCode.BAS.

```
10 PRINT"Now in the subprogram."
10000 X%=1:CURRENT%=0
10010 OPEN"R",X%,"TEMP",18
10020 FIELD X%,14 AS PN$,4 AS LN$
10030 GET X%,1:REC=VAL(PN$):REC=REC-1:LSET PN$=STR$(REC):PUT
X%,1
10040 GET X%,REC+1:XX=CVS(LN$):PN$=PN$+"":CLOSE X%
10050 CHAIN PN$,XX,ALL
```
                                                                    End

Program Listing 5. Sample code using self-explanatory variable names.

```
100 UPDATE.RECORD=10:DELETE.RECORD=1000:DISPLAY.RECORD=2000
  .
  .
  .
10000 X%=4:CURRENT%=1:PROGRAM.NAME$="THISCODE.BAS":ON ERROR
GOTO 50000
10010 ACTION.CODE=DISPLAY.RECORD
10020 IF CURRENT% THEN GO.PROGRAM$="THATCODE.BAS":ERROR 255
10030 'More code follows
  .
  .
  .
50000 XX=ERL:RESUME 50010
50010 OPEN"R",X%,"TEMP",18
50020 FIELD X%,14 AS PN$,4 AS LN$
50030 GET X%,1:REC=VAL(PN$):IF REC=0 THEN REC=1
50040 REC=REC+1:LSET PN$=STR$(REC):PUT X%,1
50050 LSET PN$=PROGRAM.NAME$:LSET LN$=MKS$(XX):PUT
X%,REC:CLOSE X%
50060 CHAIN GO.PROGRAM$,ACTION.CODE,ALL
```
                                                                    End

Photograph by Edward Judice

Internal Affairs

by Robert D. Covington

What you need to know about the 8088's architecture and machine-language instruction set.

Sooner or later, every Basic programmer thinks about moving to a faster, more powerful language. The last straw could be a sort that takes too long, a keyboard input routine that can't keep up with the typist, or a DOS feature that can't be accessed in Basic.

But why learn another interpreted language, like C, Pascal, or Forth, when you could learn to program in machine language instead? Interpreted languages stand between you and your computer; machine language lets you communicate directly with your computer. It's the alphabet upon which all programming languages depend.

In this article, I'll help you get started with machine-language programming on an MS-DOS computer. I'll begin by introducing the architecture of Intel's 8088 microprocessor, the central-processing unit in the Tandy 1000 and 1200. Then, I'll cover a number of the instructions used to program the 8088. You'll need an understanding of terms like "bytes," "bits," and "memory." While I'll limit the discussion to the 8088, many of the concepts I'll discuss also apply to the 80186 and 80286 microprocessors.

Memory Organization

One of the most important functions of a CPU is transferring data to and from internal memory or external devices connected to it. In most cases, the CPU transfers data from one memory location to another. An 8-bit CPU like the Z80, 8085, or 6809 has a 16-bit address bus, which can address 65,535 (2^{16}) bytes of memory at a time. To access more than 64K of memory, 8-bit systems have a feature called bank switching, which allows the CPU to switch between two or more banks of memory. The Model 4 and Color Computer commonly use the ROM (read-only memory) that holds the Basic interpreter as a memory bank. When ROM is not in use, the CPU replaces its memory addresses with RAM (random-access memory).

Since a 16-bit CPU like the 8088 can address up to 1 megabyte (1,048,576 bytes) of RAM at a time, bank switching is not a usual feature. Instead, the CPU uses a 20-bit word to describe the address of each byte in the system's memory. The CPU's software derives the 20-bit address by converting two 16-bit values—one describing a 64K memory segment and the other describing an offset address inside the memory segment—to 20-bit numbers and then adding them. To convert the values to 20-bit numbers, the CPU adds 4 zero bits to the right of the segment value and 4 zero bits to the left of the offset value.

For example, suppose the segment value is 5678 hexadecimal (hex) and the offset value is 1234 hex. Adding 4 zero bits to the right of the segment value produces the 20-bit number 56780 hex. Adding 4 zero bits to the left of the offset value produces 01234 hex. Adding the two creates the 20-bit address 579B4 hex.

Though more complicated than standard linear memory addressing, memory segmentation has several advantages. Since the segment offset uses a 16-bit register, programs written for 8-bit machines with 16-bit address spaces can be made to run in one segment on the 8088. Also, if a program modifies only the segment offset when it operates, you can relocate the program to any 16-byte (2^4) boundary in the computer's memory by changing the initial segment value.

Machine-language programs typically hold the segment portion of the address constant, which means memory references in the current segment can be addressed with 16 bits instead of 20, generating more compact code. This is another important advantage.

The main disadvantage of memory segmentation is that the CPU can't easily address the entire 1 megabyte of available memory in a linear fashion. For it to address data that occupies more than 65,535 bytes, the program would have to modify both the segment and offset registers. This reduces the program's speed, since it has to calculate both the segment and offset values for an address. This is one reason why the creators of GW-Basic, Turbo Pascal, and many text editors limited the amount of data the programs can access to less than 64K.

8088 Registers

In addition to the 1 megabyte of memory it can address, the 8088 contains 28 bytes of RAM divided into 14 named registers (see Table 1). Registers can be addressed faster than normal memory because they are stored internally in the CPU.

Each register is 16 bits wide. The AX, BX, CX, and DX registers can be divided into eight 8-bit registers: AH, AL, BH, BL, CH, CL, DH, and DL. (The letters "H" and "L" indicate the high and low bytes in the register.)

Though most registers are used for holding and manipulating data, each also has specific functions that it can perform in machine-language programs. The AX register, also called the accumulator, holds operands for math and logic functions. It may also be the source or destination of data moved between memory and input and output (I/O) ports.

The BX, BP, SI, and DI registers address data in the system's memory. I'll describe these registers in more detail later.

CX is the counter register for loops. The CPU decrements this register after each execution of a looping instruction. When the register is equal to zero, the loop falls through. In Basic, you would write the

code for this type of loop as follows:

FOR CX = 50 TO 1 STEP – 1:NEXT CX

The CPU may use the DX register for addressing I/O ports, but as port I/O is not common in machine-language programs, the register is usually reserved for storing data.

The registers discussed so far can be used either for their specific function or for storing and manipulating data. The following registers are not as flexible; they perform only their specific system functions.

The IP (sometimes called PC) register is a program counter. It works in conjunction with the CS (code segment) register to create a 20-bit address that points to the program instruction currently executing. When the contents of the IP register change, the program starts executing at the new address specified by the register. Since the IP register is only 16 bits wide, you have to change the CS segment register to access data beyond its 65,536-byte range.

The SP and SS registers also work in conjunction to point to the general-purpose data area called the stack. Applications programs save the contents of a register or memory address temporarily in the stack. The CPU also uses it to store important return execution points. I'll describe the stack in more detail later.

The status register indicates the status of the processor after it executes certain operations. The 8088 uses nine status bits or flags for this purpose (see Table 2).

CS, DS, SS, and ES are the four segment registers; they indicate the location of the 64K memory segment. The DS and ES registers, respectively, provide the segment portion for addressing program data. Having two segments makes it easy to work with two groups of data at the same time (when comparing strings, for example). The code, stack, data, and extra data segments can occupy the same, overlapping, contiguous, or separate areas.

Moving Data

Now that you are familiar with the 8088's architecture, you can begin experimenting with the microprocessor's machine-language instruction set. Let's start with the simplest and most common instruction: MOV.

The MOV instruction moves data from one area of memory to another. The syntax in Assembly language is:

MOV   destination,source

where the destination and source may be registers, data, the contents of a memory address, or a mem/reg. A mem/reg is the 8088's global-addressing mode.

The simplest application of a MOV instruction is for moving data into a register. This is sometimes called "immediate" addressing. For example, the instruction:

MOV   DX,32767

moves the value 32767 into the DX register.

A slightly more complex application is for moving data between the accumulator (either the AX or AL register) and the contents of a memory address. If you use the

AL register, you can transfer only 8 bits of data; if you use the AX register, you can transfer 16 bits.

A memory address, remember, has two parts: the 16-bit segment offset and a 16-bit segment register. You must specify the 16-bit offset in the MOV instruction, enclosing it in brackets to signify that it is a memory address, not an immediate value to load into the accumulator.

In most cases, the CPU defaults to the 16-bit segment register DS to calculate the 20-bit address of the move. It then adds the segment register value and the segment offset value to create the 20-bit address. Let's say, for example, that the DS segment register value is 5678 hex. In this case, the instruction:

        MOV AX,[1234H]

loads the AX register with the 16-bit value stored at 579B4 hex (56780 hex + 01234 hex = 579B4 hex).

If, for some reason, you don't want to use the DS register as the segment register, you can change the default with the SEG (segment override) instruction. For example:

        SEGES
        MOVAX,[1234H]

loads the AX register with the combined address, calculated by adding the 16-bit value in the ES register and the value 1234 hex. The segment override option works only with the DS, ES, CS, and SS segment registers.

When assembling a program with either the IBM or Microsoft macro assembler, you must type in a segment register followed by a colon before the memory descriptor in brackets. For example, with the Microsoft compiler, or any of its compatibles, the segment override I just gave would be coded:

        MOV AX,ES:[1234H]
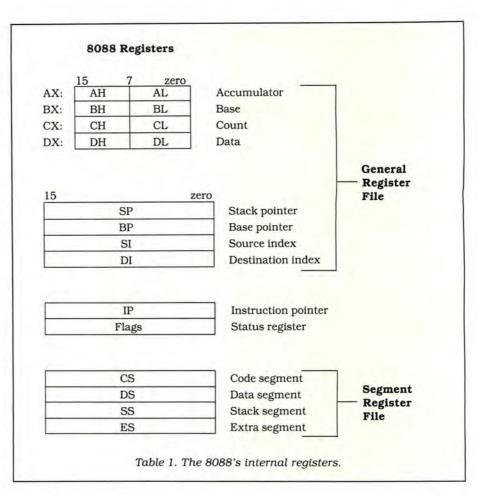
The most complex applications of the MOV instruction incorporate the 8088's global-addressing mode, traditionally called a mem/reg. A mem/reg instruction usually holds two parts: the source and the destination of an operation. The simplest data that a mem/reg instruction can describe as a source or destination is a register. If the operation uses 16-bit numbers, the available registers are AX, BX, CX, DX, SP, BP, SI, or DI. For 8-bit operations, the available registers are AL, BL, CL, DL, AH, BH, CH, or DH.

A mem/reg instruction can also describe the contents of a memory location. In this mode, you must enclose the address with brackets. For example, the instruction:

        MOV BX,[1234H]

loads the 16-bit word at 1234 hex in the current data segment into the BX register.

In other modes, a mem/reg instruction can address the contents of a memory location by referencing the contents of the BX, BP, SI, or DI register with or without

8088 Registers

	15	7	zero	
AX:	AH	AL		Accumulator
BX:	BH	BL		Base
CX:	CH	CL		Count
DX:	DH	DL		Data

General Register File

15	zero	
SP		Stack pointer
BP		Base pointer
SI		Source index
DI		Destination index

IP	Instruction pointer
Flags	Status register

CS	Code segment
DS	Data segment
SS	Stack segment
ES	Extra segment

Segment Register File

Table 1. The 8088's internal registers.

Flag name	Status bit	Function
Carry	C	Indicates if there is a carry bit from bit 7 in an addition or a borrow bit in a subtraction. CMP and SUB set this flag if the second parameter of the comparison is greater than the first.
Parity	P	Indicates when the number of set bits in the destination of a logical operation is equal to an even number (even parity).
Half carry or Auxiliary carry	A	Indicates when there is a carry from bit 3. Usually used in BCD math or nibble operations.
Zero	Z	Indicates when the result of an arithmetic or logic instruction is equal to zero. CMP sets this flag when two items being compared are equal.
Sign	S	Indicates the current condition of the most significant bit used in the operation (bit 7 for bytes, bit 15 for words). In numerical data standards, the most significant usually reflects the sign of the number.
Trap	T	When set, the 8088 performs an INT 1 after each instruction. This "single-step interrupt" is used primarily for debugging machine-language programs.
Interrupt	I	Disables all maskable enable interrupts if reset.
Direction	D	Indicates if a string instruction should auto-decrement (set) or auto-increment (reset).
Overflow	O	Indicates when the result of the operation causes an overflow in a signed number.

Table 2. The 8088's status bits.

How People With Common Interests Find An Interesting Common Ground.

Presenting CompuServe Forums. Where people from all over get together, without even leaving home.

Now thanks to CompuServe Forums, computer owners are sharing common interests by talking to each other through their computer keyboards. *Software users, computer enthusiasts, ham operators, french cooks, fire fighters, science fiction lovers and other special interest groups* are already in touch, online.

Because when you subscribe to CompuServe, you're able to reach people who want to talk about the things you do. As many people as you like. For as long as you like. Whenever you wish.

Join a conversation already in progress or start one on your own. Ask questions. And get answers.

All it takes is a modem, most any personal computer and CompuServe.

Forum members across the country are as close as a local phone call.

You can go online with just a local call in most major metropolitan areas. And normal usage fees for weekday nights and weekends are just 10¢ a minute

Of special interest to all Forum participants is software that's FREE for the taking.

Public domain software. For all sorts of activities, from games to business programs. And it's just as easy to copy a piece of software as it is to participate in a Forum.

Become a CompuServe subscriber and get a $25 Usage Credit to start you off.

Becoming a subscriber is as easy as contacting your local computer dealer. Or you can call us and order direct. Suggested retail price is $39.95.

And if you'd want more information about CompuServe, we'll be happy to send you a free brochure. Because with all that CompuServe offers—we think it's in your best interest.

CompuServe®

Information Services, P.O. Box 20212,
5000 Arlington Centre Blvd., Columbus, OH 43220
800-848-8199
In Ohio, call 614-457-0802

An H&R Block Company

an 8-bit or 16-bit offset added. Generally, the syntax for this is:

[base register + index register + displacement]

where BX or BP is the base register, SI or DI is the index register, and the displacement is an 8-bit or a 16-bit value. The displacement can assume a negative value, allowing a displacement to be subtracted from the other register. Because of this, 8-bit displacements can be in the range of –128 to 127 and 16-bit displacements in the range of –32768 to 32767. For example, the program:

```
MOVBX,124
MOVSI,76
MOVCX,[BX+SI+100]
MOVDX,[BX+SI–100]
```

loads CX with the 16-bit number at location 300 in the current data segment and loads DX with the 16-bit number at location 100.

The 8088 also supports an abridged version of this addressing mode, which allows you to omit index registers from the address calculation. Some examples of this are:

```
MOVCX,[BX]
MOVCX,[BP+1234]
MOVCX,[SI+1234]
MOVCX,[BP+SI]
MOVCX,ES:[SI]
```

Moving Strings
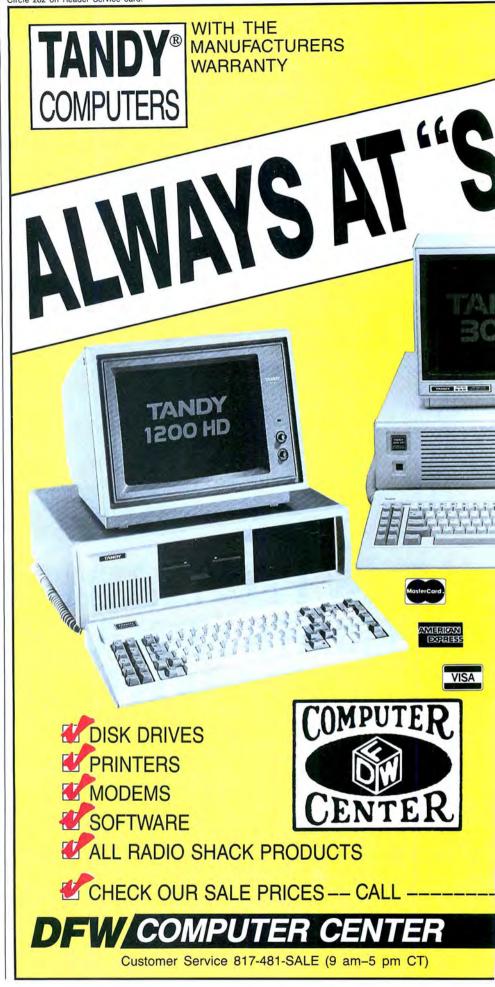
The 8088 supports five string-oriented instructions: MOVSx, LODSx, STOSx, SCASx, and CMPSx. I'll discuss only a few of these in this article.

In string instructions, the SI register points to the source offset address and DI points to the destination offset address. The offset in SI is referenced from the segment in the DS register. The offset in DI is referenced from the segment in the ES register.

When a string instruction executes, the CPU automatically increments or decrements the SI and/or DI registers. This allows the next instruction to access the next byte (or word) in the string. To set auto-increment mode, you must reset the D (direction) status flag by executing a CLD instruction. To set auto-decrement mode, use the STD instruction to set the direction flag. The CPU double-increments or double-decrements the SI and/or DI registers if the operation is a 16-bit word.

The string-movement instruction MOVSx moves the data pointed to by DS and SI to the address pointed to by SS and DI. If the operation transfers a byte (8-bit), then replace the x variable with a B (MOVESB). If it transfers a word (16-bit), replace the x variable with a W (MOVSW). For example, the program:

```
CLD
PUSH DS
POP ES
MOVSI,0
MOVDI,100
MOVSW
```

moves the word pointed to by the register pair DS and SI to the word pointed to by ES and DI. In this example, ES and DS are the same segment. The CPU double-increments (SI = 2, DI = 102) the SI and DI registers so that the next 2 bytes will transfer when you execute another MOVSW.

To make MOVSx and other string-based instructions more useful, the 8088 supports a repeat (REP) option, which causes the instruction to continue executing until the CX register is equal to zero. With each repetition of the string instruction, the CPU increments or decrements the index registers and decrements the CX register by 1. The program:

```
CLD
MOVCX,50
MOVSI,0
MOVDI,100
REPMOVSW
```

copies the memory from zero to 100 in the DS segment to the memory from 100–200 in the ES segment.

You can terminate the repeat option by setting or resetting the zero flag with REPZ and REPNZ, respectively. These instructions are generally used only in string comparisons (CMPS and SCAS).

The instructions LODSx and STOSx move data between the accumulator and memory. LODSB and LODSW move the byte or word pointed to by DS:SI into AL or AX; STOSB and STOSW load the AL or AX register into the memory pointed to by ES:DI. You might use these instructions to manipulate data—for example, to convert a string to uppercase—between transfers. Because of the processing, these instructions are rarely used with the repeat option.

Stack-based Moves

In addition to its ability to move data between memory and its registers, the 8088 can access an area of memory called the stack. The stack is a last-in, first-out (LIFO) buffer that temporarily stores data used by the processor or a machine-language program.

You might think of the stack as a box that holds many sheets of paper. As you look down at the box, only the top sheet is visible. When you stack paper in the box, the last sheet you put in will be the first you can later take out. To get at a sheet of paper in the middle of the stack, you must first remove all of the sheets on top.

Instead of paper, the 8088's stack is filled with 16-bit values. The 8088 stores the address of the last entry in the stack in the two 16-bit registers SS and SP. When a new 16-bit value enters the stack, the stack pointer (the SP register) is decremented twice (once for each byte) to keep pointing to the top of the stack. To simplify this task, the 8088 supports the PUSH and PUSHF instructions, which save 16-bit values on the stack. The syn-

tax of the PUSH instruction is simply:

PUSH source

where the source can be either a register (PUSHAX, for example) or a mem/reg (PUSH[BX], for example).

The PUSHF instruction saves the current status register status on the stack. As there is only one status register, the PUSHF instruction has no arguments.

For retrieving information from the stack, the 8088 has two complementary instructions: POP and POPF. The syntax of a POP instruction is:

POP destination

where the destination may be either a register or a mem/reg. A POP instruction loads the 16-bit value pointed to by the SS and SP registers into the register or mem/reg specified and then increments the SP register twice.

Interestingly, PUSHes and POPs do not store the register names along with the data. This means you can push a register onto the stack and then pop the value into another register. For example, to load the flags register into the accumulator, you could perform the following operation:

PUSHF
POPAX

When using these instructions, you must make sure to manage the stack properly. You can have only one PUSH for every POP in your machine-language program; otherwise, the stack may become unbalanced, resulting in erroneous data from a POP or, in some cases, a system crash.

You must also be careful not to overwrite the stack by pushing more data into it than it can store. Though the 8088's stack can store as many as 32,767 16-bit values (64K), most programs start the stack pointer at a few thousand bytes.

The stack used by MS-DOS, for example, generally can store no more than 200–300 16-bit values. In most applications programs, you can store 50–100 16-bit values on the DOS stack without running into problems.

You can create a larger stack by reserving more memory for it in your machine-language program. Be sure, however, to write the address of your program's stack in the SS and SP registers if you decide to do this.

Math Functions

Though moving data is an important function, processing and manipulating information is the microprocessor's primary job. The 8088 supports an extensive set of arithmetic and logic-based instructions. The four most common are ADD, SUB, MUL, and DIV.

The syntax for ADD and SUB is:

Operation destination,source

where the source is an immediate value or a mem/reg and the destination is an ac-

cumulator or a mem/reg. The 8088 supports both 8- and 16-bit math operations, depending on the precision of the source and destination.

The 8088 performs the math on the source and destination data and places the result in the destination operand. For example, the program:

MOV AX,1000
ADD AX,500

produces the sum 1500 in the AX register.

INC (increment) and DEC (decrement) are two additional types of add and subtract instructions supported by the 8088. INC increments the operand by 1; DEC decrements the operand by 1. Each instruction has only one operand, which can be a register or a mem/reg. Examples of legal INC and DEC instructions are:

INC AX
INC [2333]
DEC [BX]

ADC and SBC are add and subtract functions that include the carry flag. The instructions add/subtract the source, destination, and carry flag, and place the result in the destination operand. For example, if the carry flag were set, the program:

MOV AX,1000
ADC AX,1

would place the value 1002 in the AX register.

After each math operation (except for INC and DEC) the 8088 automatically updates the math-oriented flags in the status register (C,P,A,Z,S, and O). The INC and DEC instructions update all status flags, except for the carry (C) flag.

Many times you may want to check the

status bits of a math operation without changing any of the data. To find out if the AX register is equal to 1234, you could execute a SUB AX,1234 and check if the zero (Z) flag is set. If it is, you know that AX is equal to 1234, since it's the only number that will produce a zero when subtracted from 1234.

The only problem with SUB AX is that it destroys the original contents of the AX register. One way to prevent this is to push the AX register, subtract the number, and pop the AX register.

An easier method is to use the CMP (compare) instruction, which compares the operations by subtracting the two numbers and setting the status flags. Unlike SUB, CMP does not modify the original contents of the compared items.

The 8088 supports both signed and unsigned multiplication and division with the MUL and DIV instructions, which multiply/divide a mem/reg operand to the accumulator. An 8-bit multiplication places the result in the AX register. A 16-bit multiplication places the result in the 32-bit integer generated by combining the AX register with the most significant 16 bits in the DX register.

An 8-bit division places the quotient in the AL register and the remainder in the AH register. Likewise, a 16-bit division places the quotient in the AX register and the remainder in the DX register. The following are examples of legal MUL and DIV instructions:

MOV AL,2
MOV BL,5
MUL BL
DIV BL

The IMUL and IDIV instructions multi-

Instruction	Condition to jump on	Required flag status
JA	Jump if above	C and Z flag reset
JAE or JNC	Jump if above or equal	C flag reset
JB or JC	Jump if below	C flag set
JBE	Jump if below or equal	C flag or Z flag set
JCXZ	Jump if CX register = 0	
JE or JZ	Jump if equal	Z flag set

Table 3. Unsigned conditional jumps.

Instruction	Condition to jump on	Required flag status
JG	Jump if greater than	Z flag reset and S flag equals O flag
JGE	Jump if greater than or equal	S flag equals O flag
JL	Jump if less than	S flag does not equal O flag
JLE	Jump if less than or equal	Z flag set and S flag does not equal O flag

Table 4. Conditional jumps.

ply/divide signed integers and handle overflows into the 15th bit. On the 8088, the most significant bit (7 or 15) contains the sign of the value (reset is a positive integer, set is a negative integer). Signed integers can thus represent 8-bit numbers from −128 through +127 and 16-bit numbers from −32768 to +32767. IMUL and IDIV have the same operand configuration and exit conditions as the MUL and DIV instructions.

Logic Functions

AND, OR, XOR, and NOT are logic instructions that operate like their Basic counterparts. The syntax of all four is:

Operation destination,source

where the source is an immediate value or a mem/reg and the destination is an accumulator or a mem/reg. Like math operations, logic operations place results in the destination operand. Examples of legal logic instructions are:

OR AX,1234
AND BL,[1234]
XOR BX,345H

These instructions set the P, Z, and S flags in the status register to their proper values; the C and O flags are reset after each instruction executes.

You can use the TEST instruction instead of AND to modify status flags without changing data, just as you can use CMP insead of SUB. TEST has the same entry/exit conditions and the same parameters as AND.

Absolute Jumps

The 8088 supports a family of JMP (jump) instructions for controlling program flow. In machine-language programs, jumps are either absolute or conditional. An absolute jump changes a program's flow to another address whenever the instruction executes. This type is the same as Basic's GOTO instruction, except that it uses memory addresses instead of line numbers to specify where a program should begin reexecuting.

The syntax for an absolute jump is:

JMP address

where the address can be a value or a mem/reg. Absolute jumps can be intersegment, intrasegment, or short. (Most assemblers automatically determine which type of jump to use if you jump to an address inside a machine-language program and you use a label.)

With an intersegment jump (sometimes called a long jump), a program can begin reexecuting at any address in the system's memory. The location has a 32-bit address descriptor, which is defined by merging the CS segment register and the IP offset register.

With an intrasegment jump (sometimes called a near jump), a program begins reexecuting at any address in the current segment. The address descriptor is 16 bits wide so that the IP register can be changed to any address in the segment. Instead of loading the address descriptor into the IP register, however, the CPU adds it, making it possible for a program to jump to a section of code relative to the current location. This means that no matter where you load the program in memory, it can execute the same JMP instruction. Unlike a long jump, a near jump is relocatable.

A short jump is also relative to the current address, but it uses 1 byte instead of 2 to describe the address location. With a short jump, programs can begin reexecuting ±128 bytes from the byte after the instruction.

CALL and RET are two additional absolute jumps supported by the 8088. Like Basic's GOSUB and Return instructions, they permit the use of large subroutine libraries inside machine-language programs.

CALL is the same as JMP except that it saves the 32-bit address descriptor (CS:PC) of the next instruction on the stack. An intrasegment call saves only the

16-bit offset of the next instruction in the current segment. A return (RET) pops a 16- or 32-bit value off the stack and jumps to it. This instruction can be used with both intersegment and intrasegment calls. Some assemblers use the instruction RETF to designate an intersegment return and RET to designate an intrasegment one; others are supposedly smart enough to know which type you need. I advise consulting your assembler's manual to find out how it supports RET instructions.

INT is a special instruction that performs an intersegment call to an address specified in a lookup table in the first 1,024 bytes of the computer's memory. This table contains 256 32-bit addresses for subroutines in the system's memory. The parameter following INT can be a number from zero to 255 specifying which 32-bit entry you want to call.

INT is actually a forced interrupt, which performs a few other functions in the process of the call. Most newcomers to machine language use the instruction for accessing MS-DOS functions. For this purpose, it is equivalent to an intersegment call.

Conditional Jumps

Absolute jumps undoubtedly have their uses. Conditional jumps, however, add decision-making capability based on the condition indicated by the flags in the status register. For example, the program:

```
            MOV AX,20
LOOP  INC AX
            CMP AX,30
            JC LOOP
. . . . . .
```

causes the computer to loop nine times and then resume execution after the JC LOOP instruction. Table 3 lists the conditional jumps supported by the 8088 for unsigned values. Table 4 lists the conditional jumps supported for signed values.

All conditional jumps compensate for values that use the high-order bit for the sign of the number. Thus, if you compared −1 (65,535 unsigned) to 1 (1 unsigned), the JL instruction would cause a jump to the specified address, since −1 is less than 1. The jump would not execute if you used JB, since the computer reads the −1 as 65,535. Short address descriptors are the only limitation of conditional jumps. You have to use an additional near or long JMP instruction to jump nearer/farther than 128 bytes from the original instruction.

Three other conditional jumps—LOOP, LOOPZ, and LOOPNZ—emulate Basic's For. . .Next loop. These instructions decrement the CX register and jump to an address only if CX is greater than zero. (The LOOP instruction does not change the sta-

tus registers when it checks to see if CX is zero.) LOOPZ and LOOPNZ (sometimes referred to as LOOPE and LOOPNE) decrement CX and jump only if CX is greater than zero and if the Z flag is set or reset.

The following is an example of a legal LOOP instruction:
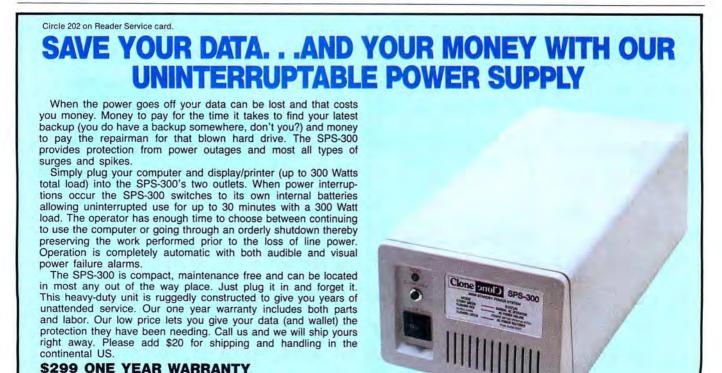
```
            MOV CX,30
AXINC  INC AX
            LOOP AXINC
. . . . .
```

When assembled and executed, this program increments the AX register by 30 and makes the CX register equal to zero.

Parting Words

The 8088 supports a number of instructions in addition to those described in this article. I suggest skimming the entire 8088 instruction set in your assembler manual before striking out on your own. Once you master the concepts and instructions I have covered, however, you'll be able to start writing your own machine-language programs. ∎

Robert D. Covington is a technical consultant with the Regional Consortium of Educational Technology in St. Louis. You can reach him at P.O. Box 3007, St. Louis, MO 63146.

Have a Nice Data

Build Data statements automatically for fast entry of machine-language code.

The average Basic programmer reacts to machine language the way most of us do to cod liver oil. Everyone knows it's powerful stuff, but few find it easy to take. The standard methods for entering machine code into a computer are all unpalatable. I find the cost of an editor/assembler unacceptable for simply entering programs. DOS Debug utilities, on the other hand, are awkward to use and, unless you know what you're doing, dangerous.

Perhaps the best alternative is to put the decimal values of machine-language code into Data statements, which a Basic program then POKEs into memory or writes to disk. This is a tiresome process, however; you have to type in a line number, then the word DATA, a decimal number, a comma, the next decimal number, and so on. The fact that the numeric keypad lacks a comma makes the process all the more cumbersome.

But what if you could automate this procedure? MakeData, a utility for the Models I/III/4 and the Tandy 1000, lets you do just that. You supply the decimal values and the program does the rest, producing an ASCII disk file of line-numbered Data statements.

On Cue

When you run MakeData (see the Program Listing), the program prompts you for a starting line number, a line number increment, the number of data items you want to insert on each line, and a file name. Type in this information and press the enter key. The program displays the opening portion of your first line and a prompt to begin entering data items.

Suppose you specify 10 as the first line of your file, a line increment of 10, and a maximum of five data items per line. After opening your file, MakeData displays the following:

```
10 DATA
Enter data item 1 = >
```

LOAD 80

System Requirements

Model I, III, 4, or 1000
Disk Basic or GW-Basic

Program Listing. MakeData.BAS

```
1 CLS:CLEAR 200:GOTO 200   ' TRANSFER TO START OF PROGRAM           '*   1451
2 '                                                                 '*    121
3 ' **MAKEDATA/BAS**                                                '*     51
4 ' By Pat Anderson                                                 '*     52
5 '                                                                 '*    124
10 ' **INPUT SUBROUTINE**                                           '*     97
15 '                                                                 '*    173
20 PR$=""                                                           '*    457
30 PRINT @ FN LO(R,C),"=>"; ' INPUT PROMPT                          '*   1559
40 IN$=INKEY$:PRINT @ FN LO(R,C)+LEN(PR$)+2, CHR$(95);: IF IN$="
   " THEN PRINT @ FN LO(R,C)+LEN(PR$)+2," ";:GOTO 40 ' DISPLAY C    '*   6639
50 IF IN$=CHR$(13) THEN PRINT @ FN LO(R,C)+LEN(PR$)+2, " ";: RET
   URN ' RETURN ON <ENTER> KEY                                      '*   3938
55 IF IN$=CHR$(8) THEN PR$=LEFT$(PR$,LEN(PR$)-1):PRINT @ FN LO(R
   ,C)+LEN(PR$)+2,"         ";:GOTO 40 ' BACKSPACE (LEFT ARROW)     '*   5531
60 PR$=PR$+IN$ ' ADD INPUT CHARACTER TO EXISTING LINE               '*    821
70 PRINT @ FN LO(R,C)+2,PR$; " ";:GOTO 40                          '*   2453
150 '                                                               '*    221
200 ' **START OF PROGRAM**                                          '*    146
210 '                                                               '*    218
220 ' **INITIALIZE VARIABLES, DEFINE PROMPTS & MESSAGES**           '*    148
230 '                                                               '*    220
240 DEFINT A-Z                                                      '*    856
250 I=1:K=1 ' I=TOTAL ELEMENTS, K=ELEMENTS ON CURRENT LINE          '*    609
260 WI=64 ' SCREEN WIDTH, CHANGE TO 80 FOR MOD IV AND TANDY 1000    '*    511
265 ' FUNCTION TO SIMULATE LOCATE STATEMENT, R=ROW & C=COLUMN       '*    157
270 DEF FN LO(R,C)=(R-1)*WI+C-1 ' DELETE FOR TANDY 1000             '*   1757
275 'FUNCTION TO GET COLUMN NUMBER TO CENTER CE$                    '*    158
280 DEF FN CE(CE$)=WI/2-LEN(CE$)/2                                  '*   1930
290 DT$=" DATA "                                                    '*    850
300 MS$(1)="MAKEDATA/BAS":MS$(2)="By Pat Anderson"                  '*   3347
310 MS$(3)="For easy entry of elements in DATA statements"          '*   4819
320 MS$(4)="Creates an ASCII file of numbered lines of DATA"        '*   4687
330 MS$(5)="statements to merge with BASIC program"                 '*   4210
340 P$(1)="First line number? ":P$(2)="Line increment? "           '*   4228
350 P$(3)="Number of data elements per line? "                      '*   3668
360 FP$="File name (incl extension and drive) for output"           '*   4878
370 EP$="Type END at data prompt to end"                            '*   3160
380 CT$="Press <ENTER> to continue "                                '*   2756
390 NL$="Press <ENTER> to start new line "                          '*   3267
400 EM$(1)=" CREATED FOR MERGE WITH BASIC PROGRAM":EM$(2)="CONTA
    INING ":EM$(3)=" DATA ELEMENTS"                                 '*   5852
410 '                                                               '*    220
430 ' **INTRO SCREEN**                                              '*    151
440 '                                                               '*    223
450 CLS                                                             '*    411
460 R=1:X=1                                                         '*    634
470 CE$=MS$(X):C=FN CE(CE$)                                         '*   1540
480 PRINT @ FN LO(R,C),CE$                                          '*   1538
490 IF X=5 THEN GOTO 510   ELSE R=R+1:X=X+1:GOTO 470                '*   3147
500 PRINT                                                           '*    578
510 R=15:C=1:PRINT @ FN LO(R,C),CT$;                               '*   2144
520 C=C+LEN(CT$):GOSUB 10   :CLS                                    '*   1831
530 '                                                               '*    223
550 ' **GET PARAMETERS**                                            '*    154
560 '                                                               '*    226
570 FOR X=1 TO 4:PRINT P$(X):NEXT X                                 '*   2197
580 PRINT                                                           '*    586
590 PRINT FP$                                                       '*    805
600 R=1:C=44:GOSUB 10 :LN=VAL(PR$) ' LINE NUMBER                    '*   2046
610 R=2:GOSUB 10 :LI=VAL(PR$) ' LINE INC'ENT                        '*   1753
620 R=3:GOSUB 10 :ME=VAL(PR$) ' MAXIMUM ELEMENTS ON LINE            '*   1752
630 R=7:C=1:GOSUB 10:F$=PR$ ' OUTPUT FILE NAME                      '*   1612
640 OPEN "O",1,F$                                                   '*    914
650 R=15:C=1:PRINT @ FN LO(R,C),CT$:C=C+LEN(CT$):GOSUB 10   :CLS    '*   3796
660 '                                                               '*    227
680 ' **DATA ENTRY**                                                '*    158
690 '                                                               '*    230
700 ST$=STR$(LN)+DT$ ' START WITH LINE NUMBER AND WORD "DATA"       '*   1198
710 R=1:C=1:PRINT @ FN LO(R,C),EP$;:IP$="Enter data item "+STR$(
    I)                                                              '*   4396
720 R=5:PRINT @ FN LO(R,C),ST$                                      '*   1820
730 R=8: PRINT @ FN LO(R,C),IP$                                     '*   1842
740 R=8:C=LEN(IP$)+1: PRINT @ FN LO(R,C),"            "             '*   2909
750 GOSUB 10:CE$=PR$                                                '*   1190
760 IF CE$="END" AND K=1 THEN CLOSE:CLS:PRINT F$;EM$(1):PRINT EM
    $(2); I-1; EM$(3):END                                          '*   5065
770 IF CE$="END" AND K<>1  THEN PRINT #1,LEFT$(CL$,LEN(CL$)-1):C
    LOSE:CLS:PRINT F$;EM$(1):PRINT EM$(2); I-1; EM$(3):END         '*   6990
780 IF K<ME THEN GOSUB 820:GOTO 710 ' BUILD LINE                    '*   2139
790 IF K=ME THEN GOSUB 910:GOTO 710 ' SAVE AND START NEW LINE       '*   2141
800 '                                                               '*    223
```

Listing continued

This is all it takes to get Tandy talking to IBM, Apple, Macintosh and Compaq.

And it's simple and inexpensive.

Now a powerful **GridNet**™ Local Computer Communications Network is as close as your nearest electrical outlet.

Thanks to GridComm's Communicators, virtually any office can create its own high speed, error-free **GridNet**™... instantly! There's no need to purchase special system software. And there's none of the expense

and inconvenience of installing additional wiring. Because a **GridNet**™ utilizes your existing electrical grid as a highly-effective data transmission medium.

And a **GridNet**™ will support IBM PC, XT, AT, and all IBM compatibles, Apple and Macintosh, Hewlett-Packard, Tandy, AT&T, Epson, NEC...all the most popular micros and peripherals, in the

same network. Freely sharing data, with complete connectivity and effective compatibility.

The cost? As much as 60% below that of specially-wired systems.

Find out just how easy it is to increase your office's productivity. See your local dealer, or contact "The Great Communicator" for more information.

GRIDCOMM

THE GREAT COMMUNICATOR

Circle 271 on Reader Service card.

P.O. Box 2779, Danbury, Connecticut 06813-2779. For information call 203-790-9077

Trademarks: IBM, PC, XT, AT/International Business Machines Corp.—Apple, Macintosh/Apple Computer Inc.—Tandy/Tandy Corp.—
Compaq/Compaq Computer Corp.—Hewlett-Packard/Hewlett-Packard Co.—Epson/Epson America Inc.—AT&T/American Telephone and Telegraph Co.—NEC/NEC Home Electronics (U.S.A.) Inc.

```
820 ' **BUILD CURRENT LINE**                              '*   154
830 '                                                     '*   226
840 AD$=AD$+CE$+CHR$(44) ' ADD CURRENT ELEMENT AND COMMA  '*  1287
850 PR$=ST$+AD$ ' BUILD LINE TO DISPLAY                   '*   863
860 R=5:C=1 :PRINT @ FN LO(R,C),PR$; ' DISPLAY LINE       '*  2146
870 CL$=PR$                                               '*   629
880 K=K+1:I=I+1:RETURN ' ADJUST COUNTERS                  '*  1390
890 '                                                     '*   232
910 ' **SAVE CURRENT LINE AND START NEW LINE**            '*   154
920 '                                                     '*   226
930 AD$=AD$+CE$ ' ADD CURRENT ELEMENT                     '*   802
940 PR$=ST$+AD$ ' BUILD LINE TO DISPLAY                   '*   863
945 'DISPLAY FINISHED LINE AND SAVE IT TO DISK            '*   162
950 R=5:C=1:PRINT @ FN LO(R,C),PR$;:PRINT #1, PR$         '*  2959
960 R=15:PRINT @ FN LO(R,C), NL$;:C=C+LEN(NL$):GOSUB 10   '*  3410
965 'INCREASE LINE NUMBER BY LINE INCREMENT AND ADJUST COUNTERS '* 164
970 LN=LN+LI:K=1:I=I+1:AD$="":ST$=STR$(LN)+DT$            '*  2782
980 PR$=STRING$(64," "):R=15:C=1:PRINT @ FN LO(R,C), PR$; :R=5:P
    RINT @ FN LO(R,C), PR$;:PRINT @ FN LO(R,C), ST$;:RETURN '*  7232
```

End

After the prompt, type in a data item (I'll use the number 12 as an example) and press the enter key:

> 10 DATA 12,
> Enter data item 2 = >

Continue typing in data items, pressing the enter key after each one. After you have typed in the maximum number (five, in this example), MakeData saves the line to disk and waits for you to press the enter key again. When you do, it moves on to the next line, which it numbers automatically:

> 20 DATA
> Enter data item 6 = >

Type in END when you finish entering data items for the file.

Stringing Along

Functionally, MakeData is an easy program to follow. Lines 10–70 contain an IN-KEY$ input subroutine. The program builds lines (PR$) one character at a time from each IN$.

When the test in line 50 detects that you have pressed the enter key, control returns to the main program. Line 55 provides for limited editing of the input line with the backspace (left-arrow) key.

Lines 200–400 initialize variables and define string constants. The program defines numeric variables as integers; that is more than enough precision for the simple math involved.

The variable WI (line 260) defines the width of the screen for the function in lines 270 and 280 that simulates GW-Basic's Locate statement. For the Models I and III, the screen width is 64; for the Model 4 and the Tandy 1000, change it to 80.

The function that simulates the Locate statement replaces ordinary Print statement syntax on the Models I/III/4. This allows Model 1000 users to substitute the statement:

> LOCATE R,C : PRINT . . .;

wherever they see:

> PRINT @ FN LO (R,C), . . .;

Model 1000 users will also want to delete line 270, which defines the simulation function. Except for these Model 1000 modifications, the program works the same on the Models I/III/4 and the Tandy 1000.

Lines 550–650 fetch the parameters. The input routine returns strings for the starting line number, line increment, and elements on each line, which are converted to numbers with the VAL function. In order to display prompts and operator input, the row (R) and column (C) are defined for the simulated Locate statement

before each call to the input subroutine. This section also opens an output file.

The STR$ function in the data entry section (lines 680–790) converts the line number (LN) back into a string and inserts the word DATA at the beginning of every line. After you type in a data item, the program tests whether the maximum number (ME) has been reached for the current line. If it has not, the program branches to lines 820–880 and tacks on the new data item and a comma to the end of the display line.

If the maximum number of items has been entered, lines 910–980 save the line to disk. After you press the enter key, the program increases the line number (LN) by the line increment (LI) and starts a new line.

Lines 760–770 test to see if you have typed in END. If an End statement is detected as the first item on a new line, the program closes the output file and ends the program.

If an End statement is not the first line item, then the program saves the current line (CL$), closes the file, and terminates. In either case, the screen will display both the file name and the number of data items saved in it.

Easy to Swallow

The file that MakeData creates is a straight ASCII text file. As the program includes no provision for editing input after you press the enter key, you will have to use either your word processor or the Basic line editor if you want to go back and edit a MakeData file.

Use a Basic program with a For. . .Next loop to POKE the data items into memory. Afterwards, write the program to disk as a /CMD file with the TRSDOS Dump command or by executing the following:

```
10  OPEN "O",1,"file name/CMD"
20  FOR K = 1 TO N:READ A:PRINT#1,
    CHR$(A);:NEXT
30  CLOSE:END
```

where N = number of data items (see Crosscheck program lines 200–220 in "A Basic Programmer's Best Friend," September 1985, p. 66).

MakeData accepts data of any type or length. Some Data statement formats contain integers that are longer than the numbers used with the POKE technique, for instance. Also, some POKE routines use data in hexadecimal format. If you wanted, however, you could easily modify the program to limit acceptable input to numbers, the enter key, and the backspace (left-arrow) key (see "Restricted Entry," May 1985, p. 70). ■

A lawyer by vocation, Pat Anderson has made computing his hobby since 1982. You can reach him through CompuServe (user number 72027,1055) or by writing to him at 36631 S.E. Fall City Road, Fall City, WA 98024.

A Hard Bargain

The edifying story of one Tandy 1000 owner's efforts to install a bargain-priced 10-megabyte hard disk system.

One of the factors I weighed when considering the purchase of a Tandy 1000 was the ease with which I could upgrade the system. In the past two years, the aftermarket for MS-DOS computers has produced a great many add-ons that users can install themselves. Most that are built on 10-inch boards (or smaller ones) are compatible with the Tandy 1000.

This article describes the steps I followed in adding an external 10-megabyte hard disk drive to my Tandy 1000. Although the process wasn't as trouble-free as I might have wished, along the way I learned a lot about how the 1000 operates. More importantly, I increased the storage capacity of my system while saving a couple hundred dollars by doing it myself. Maybe you can, too.

Catching the Bug

About a month after I bought my 1000, I was bitten by the add-on bug. While scouring the computer offerings at a local hamfest (a flea market for amateur radio operators where one often finds bargains on computer-related components), I encountered a vendor selling an unused Western Digital hard disk controller board, along with a half-height, 5¼-inch 10-megabyte drive, for under $400 (see Photos 1 and 2). In spite of having no assurance that it would work with the Tandy 1000, I decided to purchase the package and install it myself.

My bargain came complete with drive, 10-inch controller board, front panel, and cables. In lieu of documentation, the vendor at the hamfest wrote out the following installation procedures:

1. Place a jumper on SW1-4 (switch 1, row 4).
2. Boot the system from DOS in drive A.
3. At the A>prompt, type DEBUG and press the enter key.
4. At the – prompt, type G = C800:5.
5. At the next prompt, type Y.
6. After a couple of minutes, the monitor will display the message "Format successful."
7. At the A>prompt, type FDISK.
8. Next prompt, press the enter key.
9. Next prompt, press the enter key.
10. Next prompt, press the enter key.
11. At the A>prompt, type FORMAT C:/S and press the enter key.
12. At the next prompt, press any key.

He assumed I was installing the drive in an IBM PC, and I made no indication to the contrary.

Photo 1. Western Digital hard disk controller.

Photo 2. Half-height, 5¼-inch 10-megabyte drive.

Failed Attempts

At home, I removed the cover from the 1000's central processing unit (CPU), detached the cables from drive B, and attached the four-conductor plug in the computer to the mate in the hard drive. Setting the drive temporarily on top of the floppy drive shield in the CPU, I made the other two cable connections from the hard disk controller card to the hard drive, carefully observing polarity on each end of the cables. I then installed the jumper on SW1-4 as the hamfest vendor had instructed. After one last check of the connections, I was ready to boot up.

System Requirements

Model 1000

I placed the MS-DOS disk in drive A and turned on the power switch. Halfway through the boot, the drive select light for the unformatted hard disk came on—and stayed on. The boot never completed.

I began to think my purchase wasn't a bargain after all. The sight of the Tandy 1000 with its dislocated monitor and exposed wires gave me a sick feeling. I decided to clean up the mess, pack away the hard drive, and reassemble the 1000.

The following day, I called Western Digital about my problem. I learned that the controller board had an installation manual, which they would send in the mail. Within 48 hours, it was at my door, postpaid.

A study of the Western Digital manual and the Tandy 1000 technical manual produced an important discovery: I learned that the IBM PC uses interrupt number 5 to service the hard disk, while Tandy uses interrupt number 2. The controller board defaults to the former. With some minor tinkering, however, I could change the default. The manual explained exactly how. All it amounted to was cutting a trace and moving a jumper.

I made the change and again opened the 1000 and disconnected the cables from drive B. Then I installed the controller board and made the connections to the hard drive. Anxiously, I turned the power on. Again the system hung up halfway through the boot. It was back to the drawing board.

A New Lead

I placed another call to Western Digital and spoke with an engineer who told me that there were two known problems installing the board in the Tandy 1000. One is the interrupt handler default (which I had already solved); the other involves the BIOS (basic input/output system) ROM on the controller board. Apparently, Western Digital had developed several ROMs for the board, but only one could be used on the Tandy 1000.

The engineer managed to track down the correct BIOS ROM and offered to mail it to me free of charge. In return, I mailed him the BIOS ROM that was contained in the board I owned.

The mailman arrived with the new ROM BIOS a few days later. For the third time, I disassembled the 1000 and made

Tips for Do-It-Yourselfers

If you are considering installing a hard disk drive in your Tandy 1000, make sure you first have the appropriate hardware and software. Here are some steps you should follow:

► Determine which BIOS version your computer has. When you boot the system, the version number is displayed on the first line of the screen after the memory check.

► If you have BIOS version 01.00.00, arrange with your dealer or with a repair center to install version 01.01.00. If you have a PBJ MFB-1000 expansion board in your computer, you may need to obtain a different DMA controller from PBJ Inc.

► Buy the MS-DOS hard disk utilities from your local Radio Shack. The dealer may have to order them from the factory.

► When you purchase your 10-megabyte or 20-megabyte hard disk (I recommend buying from an established vendor), be sure that it has a BIOS ROM that works with the Tandy 1000.

► Set up the hard disk controller board (using interrupt number 2) for operation with the particular make and model of the drive. The vendor can tell you which jumpers or switches to set.

► If you install an external drive, purchase a case and power supply for it. Although most of the newer hard drive assemblies will operate from the power supply in the Tandy 1000, they push the supply to the limit.

► If you are installing a Western Digital controller board, follow the format procedure outlined on p. 68.

—Jim Creasy

the necessary drive connections. This time when I turned on the power, a miracle happened: the computer booted the MS-DOS disk in drive A. I was up and running.

I ran the first format from Debug and, sure enough, the drive select light came on for the hard drive. I heard noises on the drive and noticed that the head stepper motor was moving at a slow, but precise, rate. In about three minutes, the stepper rewound and the message "Format successful" appeared on-screen.

Sure that I had cleared all hurdles, I began to partition the disk. I typed in the FDISK command and selected option 1 to create a DOS partition. Almost immediately, the system returned the message "Error reading fixed disk." Thinking I had done something wrong, I reentered the FDISK command. The error message appeared again. I began to think about giving up.

One More Try

Before resigning myself to ownership of a $400 paperweight, I decided to call Tandy. My investigation yielded another crucial tidbit: the hard disk system was designed to work with direct-memory access (DMA), a standard feature on other MS-DOS computers. Although Tandy 1000s with 256K or more have direct-memory access, the feature is not programmed properly by version 01.00.00 of the BIOS. I learned, however, that Tandy had addressed the problem by producing a revised BIOS (version 01.01.00) for the 1000.

I took the 1000 to a Radio Shack repair center to have the upgrade installed. The procedure took less than an hour. Because installing the new BIOS version is considered an upgrade, not a repair, I was charged $20.

At last, I had everything I needed to run my Tandy 1000 with a 10-megabyte hard disk. When I returned home, I installed the controller board for the fourth time. This time, the system booted from the hard drive and the C> prompt appeared on-screen. Within an hour, I created nine directories; after another hour, the hard disk was 70 percent full.

In spite of all the effort, I am convinced I now own one of the best and least expensive MS-DOS personal computers on the market. I also had the opportunity to speak with some very helpful and nice people at Western Digital (Enhanced Peripherals Division, 2445 McCabe Way, Irvine, CA 92714, 714-863-7767) and Tandy. Figuring in phone calls to Tandy and Western Digital, travel costs to the repair center, and the purchase of a *Tandy 1000 Technical Reference Manual*, the total cost for installing an external 10-megabyte hard disk drive was $497—a bargain by any standard. ■

You can reach Jim Creasy at 517 Reeves Drive, Phoenixville, PA 19460.

Different Strokes

Two convenient utilities for the Model III keyboard.

The keyboard is the computer world's great equalizer. Not everybody writes device driver routines, but everybody types. ReadyKey, my Model III keyboard routine, combines two utilities that save you typing time by cutting down on keystrokes (see the Program Listing). The first utility copies text from one part of the screen to another. The second, a macro key utility, lets you enter Basic key words by pressing shifted letter keys; you can customize it with up to 26 character strings of your choice, including DOS commands or other frequently used strings.

Do You Copy?

To use ReadyKey, type in the Listing and assemble it to disk. Type in the routine's /CMD file name from TRSDOS Ready to run it.

You can copy text when you're at DOS level, in Basic, or from a program using the following procedure: press the clear key, and a new, fast-blinking cursor appears on-screen.

Position this cursor at the beginning of the text you want to copy, press the @ key, and then hold it down. ReadyKey copies the text at the fast-blinking cursor's location to the original cursor's location. Release the @ key to stop copying; otherwise, ReadyKey will continue copying a line to its end.

If you copy more text than you intended, you have to use the minus key to delete extraneous characters, since ReadyKey disables the left-arrow key.

Take a Letter

To use the macro keys, you simply press a shifted letter key while in Caps Lock mode and the corresponding Basic key word appears as if you had typed it in from the keyboard. The Table shows the definitions I used for the keys; some letter

LOAD 80

System Requirements

Model III
TRSDOS 1.3
Assembly language
Editor/assembler

Program Listing. ReadyKey.

```
00010           ORG     0FA00H
00020 ORG       DEFL    0FA00H;START OF THIS PROGRAM
00030           LD      HL,ORG
00040 HIMEM     DEFL    4411H;ADDRESS TO STORE HIMEM VALUE
00050 BHIMEM    DEFL    40B1H;ADRESS OF BASIC HIMEM POINTER
00060           LD      (BHIMEM),HL
00070           LD      (HIMEM),HL
00080           LD      HL,NAME
00090           CALL    021BH
00100           LD      HL,MESS
00110           CALL    021BH
00120           LD      HL,MESS1
00130           CALL    021BH
00140           LD      HL,MESS2
00150           CALL    021BH
00160           LD      HL,MESS3
00170           CALL    021BH
00180           LD      HL,MESS4
00190           CALL    021BH
00200 BEGIN     LD      HL,(4016H)
00210           LD      (BUFF),HL
00220           LD      HL,ROUTE
00230           LD      (4016H),HL
00240           LD      A,0
00250           LD      (BUFF4),A
00260           RET
00270 ROUTE     POP     DE
00280           LD      (BUFF2),DE
00290           LD      DE,(MCADD);KEY STRING BEING SENT?
00300           LD      A,D
00310           OR      E
00320           JR      Z,GET
00330           INC     DE
00340           LD      A,(DE)
00350           OR      A
00360           JR      NZ,MRE
00370           LD      DE,0
00380 MRE       LD      (MCADD),DE
00390           JR      EXIT1
00400 GET       LD      DE,REST
00410           PUSH    DE
00420           LD      HL,(BUFF)
00430           JP      (HL)
00440 REST      PUSH    BC
00450           PUSH    HL
00460           CP      13;CARRIAGE RETURN
00470           JR      Z,FIN
00480           CP      2;IF CNTRL-B PRESSED SHUT OFF
00490           JP      Z,FIX
00500           LD      D,A
00510           LD      A,(BUFF4)
00520           CP      0
00530           JR      NZ,CURSE
00540           LD      A,D
00550           CP      31;CLEAR
00560           JR      Z,FOUND
00570           LD      A,(3880H)
00580           AND     3
00590           LD      H,A;ADJUSTS FOR THE RIGHT SHIFT KEY
00600           SRL     H
00610           OR      H
00620           LD      HL,(4019H);IS CAPS LOCK IF NON 0
00630           AND     L
00640           LD      A,D
00650           JP      NZ,MACRO
00660 FIN       LD      D,A
00670           LD      A,0
00680           LD      (BUFF4),A
00690           LD      A,D
00700 EXIT      POP     HL
00710           POP     BC
00720 EXIT1     LD      DE,(BUFF2)
00730           PUSH    DE
00740           RET
00750 LEAVE     LD      B,20
00760 LEAV1     LD      D,0
00770 NXT       DEC     D
00780           JR      NZ,NXT
00790           DJNZ    NXT
```

Listing continued

Radio Shack has the best
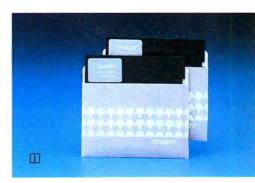
...of everything.

Accessories can make the difference between just a computer and an *efficient* computer system. That's why Radio Shack offers a wide selection of accessories that make the most of your computer.

Save valuable space with our Deluxe Printer Platform (A, $49.95). It elevates your wide-bed 132-column printer so paper feeds from below. And the Universal Printer Supports (B, $19.95) elevate your printer, so paper can be placed underneath.

If your computer workspace is cramped, then our beautifully styled Computer Workcenter (C, $79.95) is just what you need. There's plenty of room for your computer, printer, accessories . . . and your elbows.

To keep your 5¼" disk drives in tip-top shape we've got a Disk Drive Head Cleaning Kit (D, $7.95). Then store diskettes safely in our Disk Library Box (E, $4.95). It stores like a book when it's closed, and you can flip through up to 10 disks when it's open. If you need to store up to 50 disks, then choose a Diskette Storage Box (F, $14.95).

To protect your computer from power spikes, we offer a Power Line Filter (G, $49.95). It filters transients and line noise from appliances and reduces interactions between peripherals. And of course we offer a full selection of fanfold printer paper and high-quality diskettes (H, I).

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

```
00800          LD      A,(BUFF4)
00810          JR      CURSE
00820 FOUND    LD      HL,(4020H);LD HL WITH ORIGINAL CURSOR ADDRESS
00830          LD      (BUFF3),HL
00840          LD      A,1
00850          LD      (BUFF4),A
00860          LD      D,0
00870 CURSE    LD      HL,(BUFF3)
00880          CP      4
00890          JP      Z,COPY
00900          LD      A,(HL)
00910          LD      (BUFF5),A
00920          LD      B,0
00930 LIT      LD      A,191
00940          LD      (HL),A
00950          DJNZ    LIT
00960          LD      A,(BUFF5)
00970          LD      (HL),A
00980          LD      A,D
00990          CP      '-'
01000          JR      NZ,OTHER
01010          LD      A,8;BACKSPACE AND DELETE
01020          JR      EXIT
01030 OTHER    CP      31;IF CLEAR PRESSED NO MORE CURSOR COPY
01040          JR      NZ,OK
01050          LD      A,0
01060          JR      FIN
01070 OK       LD      A,(3840H)
01080          BIT     3,A;UP ARROW
01090          JR      Z,NO
01100          SCF
01110          CCF
01120          LD      BC,64
01130          SBC     HL,BC
01140          LD      (BUFF3),HL
01150          LD      B,80
01160          JR      LEAV1
01170 NO       BIT     4,A;DOWN ARROW
01180          JR      Z,NO1
01190          LD      BC,64
01200          ADD     HL,BC
01210          LD      (BUFF3),HL
01220          LD      B,80;LONGER DELAY FOR UP AND DOWN ARROW
01230          JR      LEAV1
01240 NO1      BIT     6,A;RIGHT ARROW
01250          JR      Z,NO2
01260          INC     HL
01270          LD      (BUFF3),HL
01280          JR      LEAVE
01290 NO2      AND     32;LEFT ARROW
01300          JR      Z,NO3
01310          DEC     HL
01320          LD      (BUFF3),HL
01330          JR      LEAVE
01340 NO3      LD      A,D
01350          CP      64;HIT @ TO COPY
01360          JP      NZ,EXIT
01370          LD      A,4
01380          LD      (BUFF4),A
01390 COPY     LD      A,(HL)
01400          INC     HL
01410          LD      C,2
01420          LD      D,0
01430          LD      B,A
01440 MORE     LD      (BUFF3),HL
01450          LD      A,(3801H)
01460          AND     1
01470          JR      NZ,GO
01480          LD      A,1
01490          LD      (BUFF4),A
01500          LD      A,(4021H)
01510 GO       DEC     D
01520          JR      NZ,MORE
01530          DEC     C
01540          JR      NZ,MORE
01550          LD      A,(4021H)
01560          CP      63
01570          JR      NZ,EX
01580          LD      A,(4020H)
01590          CP      255
01600          JR      NZ,EX
01610          LD      A,B
01620          SCF
01630          CCF
01640          LD      BC,64
01650          SBC     HL,BC
01660          LD      (BUFF3),HL
01670          JP      EXIT
01680 MACRO    OR      A
01690          JP      Z,EXIT
01700          LD      HL,KEYS
01710          LD      BC,TEXT
01720 NCHR     CP      (HL)
01730          JR      Z,MAC
01740          INC     HL
01750          SCF
01760          CCF
01770          PUSH    HL
01780          SBC     HL,BC
01790          POP     HL
01800          JR      NZ,NCHR
```

Key	Definition	Key	Definition
A	AND	N	NEXT
B	None	O	OPEN
C	CHR$(	P	POKE
D	DATA	Q	None
E	ELSE	R	RETURN
F	FOR	S	STRING$(
G	GOSUB	T	THEN
H	None	U	USING
I	INPUT	V	VAL(
J	None	W	None
K	None	X	None
L	LEFT$(	Y	None
M	MID$(	Z	None

Table. ReadyKey's key assignments.

keys are undefined and will not produce a key word when shifted.

To substitute your own strings for those in the Table, you need to modify lines 2230–2580 of the Listing. You can use any ASCII sequence you want up to 255 characters. However, you can't use zeros in your strings as they mark the beginning and end of a definition.

You must begin your key definitions with a DEFM statement followed by a list of the keys you're defining. For example, if you want to define only the A, C, and D keys, you use the statement DEFM 'ACD'. The list need not be in alphabetical order.

Next, set up your list of key words (or other strings). The first string corresponds to the first letter key in your list of defined keys, and so on. Mark the end of each key definition with a DEFB 0. For example, the definition for A is DEFM 'AND' DEFB 0.

You don't need a DEFB 0 before the first definition, since the one at the end of the list of defined keys doubles as a leading zero for the first key definition.

Et Cetera

Because the Model III ROM's key scan routine calls ReadyKey, it might not work with all programs. The program resides in high memory and tells TRSDOS 1.3 not to overwrite it. However, other DOSes might overwrite the program, making your computer crash. For this reason, you can turn off the keyboard routine by pressing control-B (shift/down-arrow/B). ■

You can write to Chris Joy at Aiglon College, 1885 Chesiers-Villars, Switzerland.

Related Articles

Archer Jr., Rowland, "Uni-Key," September 1980, p. 76. A Model I utility that lets you enter Basic key words with one keystroke.

Rigg, Don, "Autokey," December 1982, p. 280. A Model III conversion of Uni-Key.

Listing continued

```
01810           JP      EXIT
01820 MAC       LD      BC,KEYS
01830           SBC     HL,BC
01840           LD      E,L
01850           INC     E
01860           CALL    FINDE
01870           LD      A,(HL)
01880           LD      (MCADD),HL
01890           JP      EXIT
01900 EX        LD      A,B
01910           JP      EXIT
01920 FINDE     LD      BC,400H
01930           LD      HL,TEXT
01940           XOR     A
01950 TIMES     CPIR
01960           DEC     E
01970           JR      NZ,TIMES
01980           RET
01990 FIX       LD      HL,(BUFF)
02000           LD      (4016H),HL
02010           JP      EXIT
02020 NAME      DEFM    'ARROW KEY EDITOR BY CHRIS JOY'
02030           DEFB    13
02040 MESS      DEFM    'PRESS THE CLEAR KEY TO ENTER AND
EXIT.'
02050           DEFB    13
02060 MESS1     DEFM    'PRESS ARROW KEYS TO MOVE CURSOR.'
02070           DEFB    13
02080 MESS2     DEFM    'PRESS @ TO COPY.'
02090           DEFB    13
02100 MESS3     DEFM    'PRESS SHIFT AND ANY KEY FROM A-Z,
WHILE IN UPPERCASE MODE, TO OBTAIN BASIC COMMAND.'
02110           DEFB    13
02120 MESS4     DEFM    'PRESS SHIFT,DOWN ARROW,B TO DELETE
PROGRAM.'
02130           DEFB    13
02140 MCADD     NOP
02150           NOP
02160 BUFF      DEFS    2
02170 BUFF2     DEFS    2
02180 BUFF3     DEFS    2
02190 BUFF4     DEFS    1
02200 BUFF5     DEFS    1
```

```
02210 INPUT     DEFS    19
02220           NOP
02230 KEYS      DEFM    'ACDEFGILMNOPRSTUV'
02240 TEXT      DEFB    0
02250           DEFM    'AND'
02260           DEFB    0
02270           DEFM    'CHR$('
02280           DEFB    0
02290           DEFM    'DATA'
02300           DEFB    0
02310           DEFM    'ELSE'
02320           DEFB    0
02330           DEFM    'FOR'
02340           DEFB    0
02350           DEFM    'GOSUB'
02360           DEFB    0
02370           DEFM    'INPUT'
02380           DEFB    0
02390           DEFM    'LEFT$('
02400           DEFB    0
02410           DEFM    'MID$('
02420           DEFB    0
02430           DEFM    'NEXT'
02440           DEFB    0
02450           DEFM    'OPEN'
02460           DEFB    0
02470           DEFM    'POKE'
02480           DEFB    0
02490           DEFM    'RETURN'
02500           DEFB    0
02510           DEFM    'STRING$('
02520           DEFB    0
02530           DEFM    'THEN'
02540           DEFB    0
02550           DEFM    'USING'
02560           DEFB    0
02570           DEFM    'VAL('
02580           DEFB    0
02590           END     0FA00H
```

End

Transfer Points

Build your own null modem adapter
for moving files between computers.

Photo. Null modem adapter.

Many owners of Tandy MS-DOS computers face the job of transferring Model I/III/4 files to their new machines. Some transfer methods require extra hardware or software that can cost serious money. But for a few dollars, we built a null modem device, which allowed us to move files from a Model 4 to a Tandy 1000 via their serial ports (see the Photo). Since TRS-DOS 6 has a COMM utility and the 1000 comes with DeskMate, we didn't need additional communications software. You could, of course, use a similar setup to move files between any two computers.

Making the Connection

To port files, your Tandy 1000 must have the RS-232C upgrade sold by Radio Shack (stock number 25-1006, $99).

The null modem is an adapter that you connect to one of the computers—it crosses send and receive lines so the two machines can communicate. To build your own, you'll need male and female 25-pin D-sub-miniature connectors (part numbers 276-1547 and 276-1548 respectively) or the equivalent. The Table shows the wiring configuration. We tied pins 6, 8, and 20 together and the device still worked.

System Requirements

All systems

Male Connector		Female Connector	
1	Protective ground	1	Protective ground
2	Transmit data	3	Transmit data
3	Receive data	2	Receive data
4	Request to send	5	Clear to send
5	Clear to send	4	Request to send
6	Data set ready	20	Data terminal ready
7	Signal ground	7	Signal ground
8	Carrier detect	20	Data terminal ready
20	Data terminal ready	8	Carrier detect
20	Data terminal ready	6	Data set ready

{ other pins not required for this function }

Table. Pin configurations.

Current Status:

Autodial Modem	No
BAUD Rate	300
Data Word Length	8 BITS
Parity	NONE
Number of Stop Bits	1 BIT
XON/XOFF Flow Control	ON
ASCII Character filter	OFF
Line Feed Filer	OFF
Echo (Half Duplex)	OFF
Redial (# of Retries)	0

Figure 1. The screen display of DeskMate protocol parameters.

Only nine pins have to be wired on each connector. Don't use ribbon cable, which is awkward for crossing connections. Use solid wire instead of stranded for better support. The wire path should be flexible so the connectors won't dislodge if the computers are accidentally bumped. Figures 1 and 2 show the protocols you should use to configure DeskMate and SETCOM. Pressing the clear and 8 keys together gives you a menu showing the COMM settings.■

Dan Keen and Robert Murray can be reached at RD 1 Box 432, State Highway 83, Cape May Court House, NJ 08210.

RS232 parameters:

Baud	= 300
Word Length	= 8
Stop Bits	= 1
Parity	= NONE
Break Value	= X'03'

Output control line status:

DTR	= OFF
RTS	= OFF

Input control line conditions observed:

RI	= IGNORE
DSR	= IGNORE
CD	= IGNORE
CTS	= IGNORE

Figure 2. The screen display of SETCOM parameters.

Related Articles

Dixon, Bradford N., "You *Can* Get There from Here," September 1985, p. 36. A tutorial on transferring files between computers.

Rowell, Dave, "Have Data, Will Travel," October 1985, p. 52. A tutorial on data file compatibility.

"And the winner is . . ."

80 Micro's
Best of the Model 4

CW Communications/Peterborough
© 1985 Peterborough, NH 03458

Never before has one disk contained so many top-notch programs for Tandy Model 4 users! Introducing. . .80 Micro's "Best of the Model 4." One dozen of the most powerful and practical ready-to-run programs, selected by 80 Micro's experts as the all-time best.

Money-savers and power boosters

You'll work smarter with business programs like "Time Keeper" and Hardin Brothers' "Basic Data Base Management System." Included are programs such as:

- **A Happy Medium**—lets you run Model I/III cassette software in your Model 4.
- **Convert**—Turns Model III Basic programs into Model 4 Basic.
- **Autorun**—Puts Model 100-style bar cursor directories on your Model 4.

The ultimate in Model 4 productivity

You'll also get more out of your computer with such "Best of the Model 4" selections as:
- **Extra Strength DOS**—sets up JCL to automatically configure DOS.
- **Scripaid**—reassigns Scripsit's command keys, provides on-screen text preview, and more.
- **Command Performance**—Edits your JCL files and automatically executes TRSDOS commands.
- **Restored Art**—puts graphics commands Set, Reset, and Point back into Basic.
- **Life**—a machine-language version of a classic computer game.
- **Windows**—full-blown window driver, includes the ability to execute any DOS command from within a program.
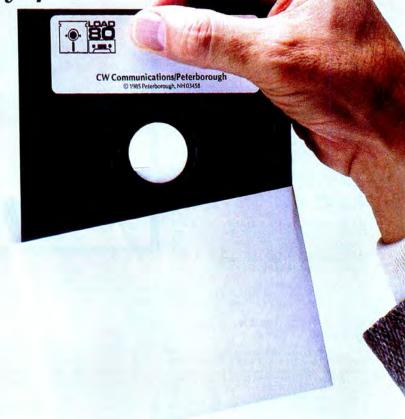
Get immediate results from every program!

Best of all, you can enjoy any program in a matter of seconds. Simply load the programs into your Model 4—no time-consuming keyboarding or debugging! Complete, easy-to-follow loading instructions are included on each disk.

Get a winning software package for your Model 4 today. Complete the coupon or attached order card and return with payment. Or, for faster service, call toll-free **1-800-258-5473.** (In NH, dial 1-924-9471.)

Available in April.

by Roberto Refinetti

Time on Your Side

Reset the Model 4 system time and date from Basic.

Accessing the Model 4 clock with the Basic TIME$ and DATE$ functions is easy enough to accomplish, but resetting the clock is inconvenient. You must return to the operating system via the System statement and wait while TRSDOS loads an overlay into memory, resets the time or date, and reloads part of Basic. As an added obstacle, TRSDOS accepts only dates that fall between January 1980 and December 1987.

These limitations can be annoying if you need to print the date and time in reports and other documents. TimeDate (Program Listing 1) simplifies matters by letting you reset the time and date to any value directly from Basic.

Beat the Clock

TimeDate is a short subroutine you may use in any Basic program. It inserts a machine-language routine (lines 6050–6060 in Listing 1) in the storage space of a Basic variable (lines 6000–6030). When you need to reset the time or date in a program, you call the subroutine, which places the new values where TRSDOS regularly stores this information in memory. (Program Listing 2 contains the machine-language routine's source code.)

You can store TimeDate on disk and merge it with other programs as needed (see your Basic manual for instructions). Place a GOSUB 6000 call after every Clear statement. After the subroutine executes, the machine code will be properly installed. To reset the time, use the command:

CALL TIME (X%,Y%,Z%)

in which variable X% contains the hour, Y% contains minutes, and Z% contains seconds. For the date, use:

CALL DATE (X%,Y%,Z%)

in which the variables contain the month, day, and year respectively. The TIME$ and DATE$ functions will return the values you assign variables X%, Y%, and Z%. The date can be set to any value and will be updated at midnight. Program Listing 3 shows a simple example.

System Requirements

Model 4/4P/4D
Disk Basic

Don't expect all this from TRSDOS, however. Any year later than 1987 will still appear incorrectly at TRSDOS Ready. You might want to type in SYSTEM (DATE = NO) to disable the date prompt that appears when you turn on your computer.

Time Limits

The values for X%, Y%, and Z% must be in the usual two-digit format; for instance, 86 for the year, 03 for the hour. You may give these variables any name— as in CALL TIME (HOUR,MINUTE,SECOND)—but they must be variables. Also,

the variables must be integer variables defined either by a % after their names or by a DEFINT statement.

Finally, to save PEEKs and POKEs in the routine, I used the top eight memory addresses as a nonrelocatable buffer for the machine-language program. If you are not using filters and drivers—in most cases, even if you are—this little buffer won't give you any trouble.■

Roberto Refinetti can be reached at the Dept. of Psychology, University of California, Santa Barbara, CA 93106.

Program Listing 1. TimeDate/BAS.

```
6000 TIMEDATE$=STRING$(25,32) : TIMEDATE=VARPTR(TIMEDATE$)+1
6010 TIMEDATE=PEEK(TIMEDATE)+PEEK(TIMEDATE+1)*256 : RESTORE 6050
6020 FOR DATETIME=0 TO 24 : READ MORETIME%
6030 POKE TIMEDATE+DATETIME,MORETIME% : NEXT DATETIME
6040 TIME=TIMEDATE : DATE=TIMEDATE+4
6050 DATA 62,19,24,2,62,18,229,213,197,33,248,255,239
6060 DATA 213,225,22,3,193,10,119,35,21,32,249,201
6070 RETURN
```

End

Program Listing 2. Source code for TimeDate's Data statements.
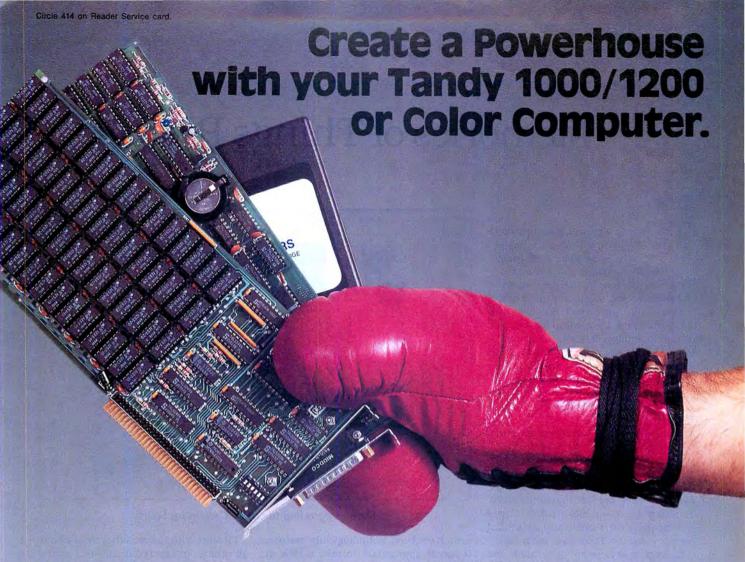
```
00100           ORG     3000H        ;DUMMY ORIGIN
00110 TIME      LD      A,19         ;SELECT @TIME SVC
00120           JR      COMMON       ;GO TO MAIN PROGRAM
00130 DATE      LD      A,18         ;SELECT @DATE SVC
00140 COMMON    PUSH    HL           ;BEGINNING OF MAIN PROGRAM
00150           PUSH    DE           ;   SAVE PARAMETERS RECEIVED
00160           PUSH    BC           ;   FROM BASIC
00170           LD      HL,65528     ;SUPPLY BUFFER TO SVC AT THE
00180           RST     28H          ;   TOP OF MEMORY
00190           PUSH    DE           ;GET ADDRESS OF TIME$ OR DATE$
00200           POP     HL           ;AND PUT IT HERE
00210           LD      D,3          ;COUNTER FOR LOOP
00220 LOOP      POP     BC           ;GET ADDRESS OF X%, Y%, Z%
00230           LD      A,(BC)       ;AND FIND THEIR VALUE
00240           LD      (HL),A       ;AND PUT IT INTO TIME$ OR DATE$
00250           INC     HL           ;NEXT SPOT IN TIME$ OR DATE$
00260           DEC     D            ;NEXT ITEM IN THE LOOP
00270           JR      NZ,LOOP      ;IF NOT DONE, THEN REPEAT
00280           RET                  ;ELSE RETURN TO BASIC
00290           END
```

End

Program Listing 3. Sample program using TimeDate.

```
10 CLS : CLEAR
20 GOSUB 6000
30 LINE INPUT "Enter date (MM/DD/YY): ";D$
40 MM% = VAL(LEFT$(D$,2))
50 DD% = VAL(MID$(D$,4,2))
60 YY% = VAL(RIGHT$(D$,2))
70 PRINT "Current date: ";DATE$
80 CALL DATE(MM%,DD%,YY%)
90 PRINT "Your date: ";DATE$
100 PRINT : GOTO 30
```

End

Create a Powerhouse with your Tandy 1000/1200 or Color Computer.

When you're ready to expand, look to the champion. PBJ boards have been setting industry records for years. Designed to enhance and increase your computer's capabilities, without putting the crunch on your wallet.

Add some extra punch to your Tandy with any of these PBJ expansion boards:

XRAM: Satisfies Memory Hunger Applications

Our expanded memory board solves that only too familiar problem, "out of memory." XRAM breaks the 640K barrier imposed by DOS and adds up to 2 megabytes of memory to your system. Designed around the Intel/Lotus/Microsoft expanded memory specification, XRAM feels right at home with programs like Framework™ II, and 1-2-3® Release 2, Symphony® Release 1.1, Microsoft® Windows, and many others. In addition to the expanded memory DOS Driver, we also include a full diagnostic package and a versatile Ramdisk program.
(List $299.95 w/256K)

MFB-1000: Solves Basic Needs

Our multifunction board contains the three most needed functions on a single expansion card. The MFB-1000 contains up to 512K of Ram, a serial port for your modem, printer or mouse, and a battery

backed real time clock/calendar for your convenience. Provided with the board is a clock driver and a Ramdisk program. *(List $299.95 w/128K)*

Worpak-RS: A Real Screen for the CoCo

The newest version of the popular Wordpak, our 80 column video cartridge, goes hand-in-hand with the latest release of OS-9® for the Color Computer. The board is fully supported by software packages from both Radio Shack and third party vendors — a must for any serious application on the Color Computer.
(List $99.95)

Bonus Software Coupons*

With the purchase of XRAM or MFB-1000, we will include discount coupons worth hundreds of dollars. These coupons may be applied towards the purchase of some of the more popular software packages.

PBJ Boards are available at dealers nationwide and at your local Radio Shack store.

pbj PBJ, Inc.
503 East 40th Street, Paterson, NJ 07504, 201-523-8663

Heavyweight performance at a knockout price.

Radio Shack and Tandy are registered trademarks of Tandy Corporation.
Microsoft and Windows are registered trademarks of Microsoft Corporation.
1-2-3 and Symphony are registered trademarks of Lotus Development Corp.
OS-9 is a registered trademark of Microware, Inc.

*This offer may be discontinued without notice at the sole discretion of PBJ, Inc.

Memories of Things Past

This month's project is an update to the I/O-accessed memory board I discussed in the September 1985 Project 80 (p. 76). Many readers have asked how to increase the board's memory capacity, so I redesigned it to support up to 64K.

This is the last Project 80—after two and a half years, the column has reached its retirement age. To wrap things up, I'll take a brief look at the technological achievements that have made modern computers and peripherals possible.

I/O-Accessed Memory Board Revisited

The original I/O-accessed memory board contained a single 24-pin JEDEC socket and permitted access of up to 4K of ROM or 2K of RAM. The new board allows up to 64K of memory, using eight 28-pin sockets (see Fig. 1 and the Photo). You may use fewer sockets if you need less memory; each socket supports 8K of memory (see the Table for a parts list).
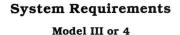
Eight chip selects are generated, one for each 28-pin memory socket. CS0/ goes to the lowest-addressed socket, while CS7/ goes to the highest-addressed socket. If you are using fewer than eight sockets, use CS0/ first, then succeeding chip selects until all of the sockets have a chip select. You can leave unused chip selects unconnected.

Each socket has a jumper allowing selection of RAM (6264 device) or ROM (2764 device). It's OK to have any mix of RAM and ROM.

The operation of the board is the same as described in the September 1985 column, except for the increase in memory. Before accessing the board, Model III and 4 users must remember to execute an OUT 16,236 command; this allows external I/O bus access.

Silicon Technology

My projects have used available technology as a tool to accomplish a wide variety of electronic tasks. While most of the electronic devices used in this col-
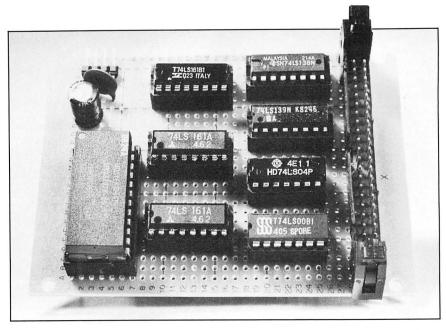
Photo. Upgrading the JEDEC memory board.

umn have been technologically mature, it seems appropriate to take a look at technological developments over the last few years.

I'll start with the area that most affects all others: integrated circuit (IC) manufacturing. IC manufacturing technology has advanced tremendously over the

Quantity	Description	Part number	Price (each)
1	74LS138 3-to-8 decoder IC (LS-TTL)	74LS138	.39
1	74LS139 dual 2-to-4 decoder IC (LS-TTL)	74LS139	.39
2	74LS04 hex inverter IC (LS-TTL) ·	74LS04	.16
4	74LS161A 4-bit latch/counter ··	74LS161	.39
1	74LS02 quad 2-input NOR gate IC (LS-TTL) ¶	74LS02	.17
1	74LS00 quad 2-input NAND gate IC (LS-TTL)	74LS00	.16
	6264LP-15 8K by 8 (64K-bit) CMOS static RAM†	HM6264LP-15	3.95
	2764 8K by 8 (64K-bit) EPROM (450 NS) ‡	2764	1.95
	47 µF/35V capacitor, electrolytic, radial		.18
9	0.1 µF/50V capacitor, ceramic disk §		.12

· If interfacing to Model III or 4, U1 should be a 7416 instead of a 74LS04.
·· Make sure the 74LS161A IC has the ''A'' suffix.
¶ Required only for Model III or 4 users.
† You can use any combination of 6264s and 2764s up to eight devices.
‡ If using fewer than eight RAM/ROM sockets, subtract one 0.1 µF capacitor for each socket not used.
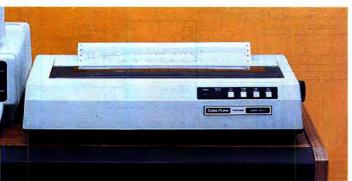
All parts distributed by JDR Microdevices, 1224 S. Bascom Ave., San Jose, CA 95128, 800-995-5430 outside CA, 800-667-6279 inside CA.
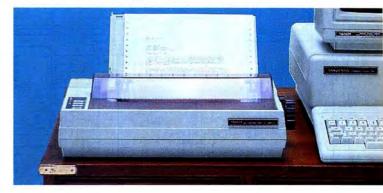
Table. Parts list for the updated I/O-accessed memory board.

System Requirements

Model III or 4

Next to your computer,

nothing beats a Tandy® printer.

Tandy printers make fine print quality, graphics and high performance affordable.

For your best value and selection in top-quality printers, shop your local Radio Shack Computer Center. We've got what you need, whatever your printing requirements.

Dot-matrix power for business

The DMP 2200 (26-1279, $1695.00) gives you efficient, fast printing at 380 characters per second. It supports elongated, double high, bold, underline, super/subscripts, italics and double strike, plus bit-image graphics. It also has a built-in tractor for perfect paper alignment.

Versatile business printer

The DMP 430 (26-1277, $899) is a 132-column dot-matrix printer with an 18-wire print head that delivers superior correspondence characters in a single pass. Choose from micro, italic and double-high fonts, as well as bit-image graphics. In the draft mode, the DMP 430 delivers a fast 180 characters per second.

Low-cost, triple-mode personal printer

The DMP 130 (26-1280, $349.95) lets you choose from word processing, data processing and dot-addressable graphics. Prints in four character styles: standard or italic cursive, in draft or correspondence modes.

Save time while printing

With a PTC 64 Printer Controller (26-1269, $249.95), you can print one job while working on another. The "print buffer" accepts and stores information from your computer.

See the complete selection of printers and accessories at Radio Shack today.

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

New 1986 Computer Catalog! Send me a copy.

Mail To: Radio Shack
Dept. 86-A-133
300 One Tandy Center
Fort Worth, Texas 76102

Name_____
Company_____
_____
Address_____
City_____
State _____ ZIP _____
Phone_____

past decade, resulting in reduced chip size, increased device speeds, and lower production costs. Other improvements include lower power consumption and, of course, higher component densities.
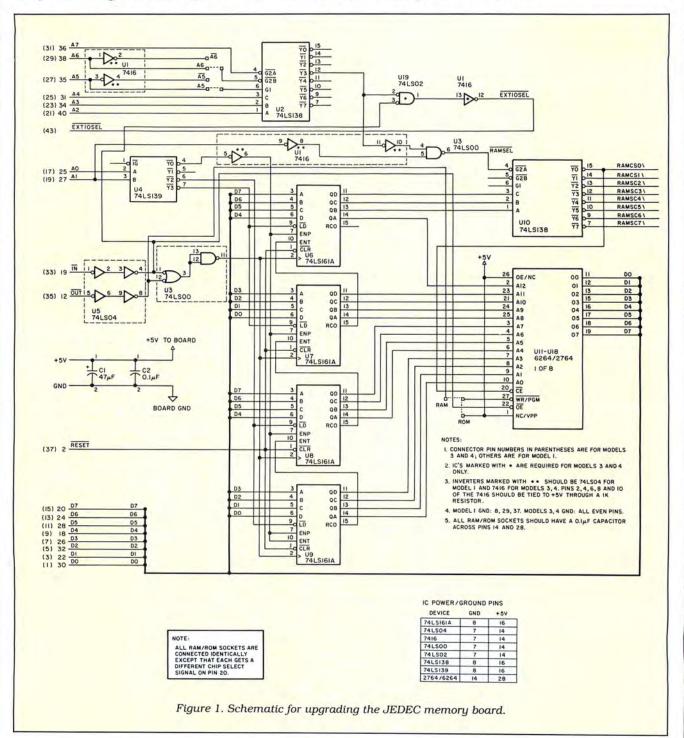
Ten years ago, you were essentially limited to standard MOS technology and a few bipolar (high-speed, high-power) technologies (TTL, RTL, DTL, and so on). The 7400-series bipolar small- and medium-scale integration (SSI and MSI) gate and interface circuit family was firmly establishing itself as a *de facto* standard. Complex devices like microprocessors and most memories used MOS technology, since bipolar technology, though much faster, generated too much heat. MOS technology was slow compared to bipolar but was tolerable for most applications.

CMOS, which provided very low operating power consumption, saw first light back around that time. Early CMOS devices were, however, unacceptably slow for many applications.

As time went on, the MOS and bipolar technologies were improved regularly. Intel Corp., for example, developed the HMOS I, HMOS II, and HMOS III, each of which allowed a smaller die, faster speed, and lower power consumption than its predecessor. Low-power Schottkey (74LS-series), Advanced Low-power Schottkey (74ALS-series) and Advanced Schottkey (74AS- and 74F-series) bipolar technologies also emerged, giving designers much more flexibility and capability, and enabling development of faster systems.

Figure 1. Schematic for upgrading the JEDEC memory board.
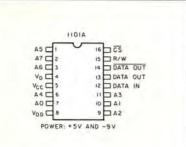
Figure 2. Intel 1101A 256-bit static RAM.
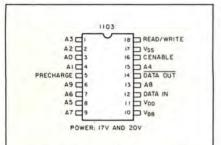
Figure 3. Intel 1103 1,024-bit dynamic RAM.
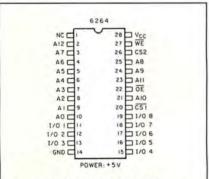
Figure 4. Hitachi 6264 64K static RAM.

Over the past three years or so, advanced CMOS technologies have given rise to the increasingly popular 74HC-series and 74HCT-series devices. These high-speed CMOS technologies have also been taking over new IC designs. CMOS's lower power consumption has allowed more densely packed components to be placed onto a chip without overheating, while at the same time improving device operating speeds. Virtually all of the new VLSI (very large scale integration) chips being developed use at least some CMOS technology in their manufacture.

Memories

There are basically two types of RAMs (random-access memories), dynamic and static. Static memories are easier to design with, since they retain what is written to them without additional circuitry or effort. Dynamic memories, on the other hand, require refreshing; portions of the devices must be read regularly to charge the small on-chip capacitors used for storing the bit values. This involves more circuitry and design effort, and often degrades system performance.

Why use dynamic memories, then, if they're more difficult to use than statics? As with many things, each memory type represents a tradeoff. The simpler design of dynamic memories permits more storage bits to be placed onto a chip than static memories of similar complexity. So, dynamics permit more memory for a given amount of circuit board area, and cost much less per bit than static memory.

Memory devices have led the way in higher chip component densities due to their regular architecture. Both statics and dynamics have evolved rapidly over the past few years, with the density doubling about every other year. I can remember the 1101A 256-bit static RAM (see Fig. 2) and the 1103 1024-bit dynamic RAM (see Fig. 3). Now, 64K statics (Fig. 4) and 256K dynamics (Fig. 5) are ubiquitous, while 256K statics and 1,024K (1 megabyte) dynamics are available.

The memory pinouts in Figs. 2–5 reveal another change that is now taken for granted. Most new devices require only a single +5V power supply, greatly simplifying system design. Earlier devices required multiple and/or unusual supply voltages (like the 1103 in Fig. 3), making designs more difficult; the different supply voltages often had to be sequenced (turned on or off in a specific order) during the power-up and power-down stages, further complicating designs. Modern dynamic memories also have simpler refresh requirements than their predecessors.

Read-only memories (ROMs) have also advanced rapidly. In the early days of microcomputers, the 1702A 2048-bit (256 by 8) EPROM (erasable/programmable ROM) was popular (see Fig. 6). Now 512K EPROMs are not uncommon (Fig. 7), and 1,024K chips are on the way. Again following the general IC manufacturing trends, todays EPROMs are faster and easier to use than the earlier ones and require only +5V, instead of the three supplies required by the earlier devices.

EPROM programming speed has also improved greatly. Intel's 1702A data sheet boasted "fast programming—two minutes for all 2,048 bits." Today's 128K EPROMs (27128), while having 64 times more memory on-chip, typically require only about two minutes for programming.

Another type of ROM also recently appeared. The EEPROM allows a new di-
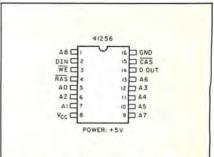
Figure 5. NEC 41256 256K dynamic RAM.
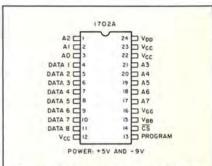
Figure 6. Intel 1702A 2048-bit EPROM.
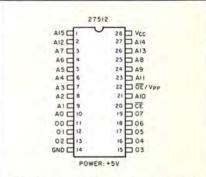
Figure 7. Intel 27512 512K EPROM.
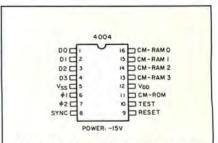
Figure 8. Intel 4004 4-bit central processing unit.

mension in ROM use, permitting information to be stored in nonvolatile ROM (not lost during power loss), while enabling in-circuit erasure and reprogram-

ming. While EEPROMs have not seen the same acceptance level as EPROMs, they have still proven to be valuable devices.

Microprocessors

Like memories, microprocessors have come a long way since their infancy in the early 1970s. One of the first microprocessors, the Intel 4004 (see Fig. 8), was a very simple device by today's standards. It was a 16-pin device with 45 instructions, capable of addressing a whopping 4,096 bytes of ROM and 640 bytes of RAM, and running at a head-spinning 750 KHz (750 thousand cycles per second). Its minimum instruction execution time was 10.8 microseconds.

A logical comparison might be to Intel's latest processor, the 80386 (likely to appear in future IBM microcomputers). The 80386, with its on-chip memory management and virtual memory support, has a full 32-bit architecture and comes in a 132-pin package. It can access up to 4 gigabytes (4 billion bytes) of memory and operates at up to 16 MHz (16 million cycles per second), with a sustained instruction execution throughput of three to four million instructions per second.

Other 32-bit microprocessors, like the National Semiconductor 32032 and 32132 and the Motorola 68020, further verify the trend microprocessors are currently taking. Not only are microprocessors becoming more advanced, but more and more system components are being placed on the same chip as the processor.

Programmable Logic Devices

Another family of devices has made its debut during the last few years: programmable logic devices (PLDs). These devices contain the equivalent of many logic gates (AND, OR, and INVERT) and flip-flops, and generally have small fuses for programming specific logic functions. They are programmed on a device similar to an EPROM programmer; in fact, some EPROM programmers double as PLD programmers.

A typical programmable logic device can replace seven to 10 SSI/MSI devices, allowing much more circuitry to be placed on a circuit board. The number of SSI/MSI devices replaced by a single PLD will continue to increase as more advanced PLDs are developed.

A Look to the Future

The semiconductor industry will continue its current trends for at least several more years. Memory densities should double about every other year and microprocessor performances should increase, though not quite as fast as they have recently. Instead of higher performance, the industry will emphasize stuffing more components onto the same chip as the microprocessor.

PLDs will play an increasingly important and dominant role in system designs. They will continue to improve in speed and complexity and will probably consume less power on the average.

Adieu

Although this is the final Project 80, I suspect you will see me pop up from time to time in the pages of *80 Micro*. Designing, developing, and debugging my monthly projects has been an enjoyable and rewarding experience. It has been especially gratifying to see the hundreds of letters from readers all over the world. I'm interested to know how you felt about the column—please write if you have any comments.■

Write to Roger C. Alford at P.O. Box 2014, Ann Arbor, MI 48106. Please enclose a self-addressed, stamped envelope for a reply.

⊓OCO⊓A

ELECTRONICS

MODEL 4D   MODEL 6000   MODEL 1000

NEW 512K MOD 3000 W/20 MEG HD. . . .2499.00

256K MOD 2000 W/MONO MONITOR . . . 1019.00	512K MOD 3000 . . . . 1819.00
256K MOD 2000 W/COLOR MONITOR . . . 1299.00	DMP 105 PRINTER. 145.00
256K MOD 1200 HD 1 DR. 1499.00	DMP 130 PRINTER. 249.00
128K MOD 1000 W/MONO 1 DR. 759.00	DMP 430 PRINTER. 585.00
128K MOD 1000 W/COLOR 2 DR. 1099.00	DMP 2100P PRINTER 1019.00
256K MOD 1000 W/10 MEG HD 1439.00	DWP 220 PRINTER 329.00
24K MOD 100 PORT 399.00	DWP 510 PRINTER 999.00
24K MOD 200 PORT 679.00	TRACTOR DWP 220 . . . . 99.00
NEW MOD 600 PORT 1149.00	TRACTOR DWP II/510 . . . . 159.00
64K MOD 4D 2DR. 839.00	TRACTOR FEED DMP2100P 99.00
512K MOD 6000 W/15 MEG HD 2579.00	PARALLEL PRINTER SWITCH 96.00
15 MEG HARD DISK 1119.00	PARALLEL PRINTER CONTROLLER 179.00
35 MEG HARD DISK 2059.00	70 MEG HARD DISK 2799.00

NEW DMP 2200 PRINTER   1149.00

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

Mice en Scene

I f you're not a dyed-in-the-wool mouse hater, you might consider trying one on your Tandy MS-DOS machine. Mice are useful as cursor movement/selecting devices for spreadsheets, graphics programs, and even word processors. A handful of programs that support mice are available, or you can write your own. I recently tried two mice on my Model 1000: Microsoft's Serial Mouse, and Tandy's Digi-Mouse.

Into the Laboratory
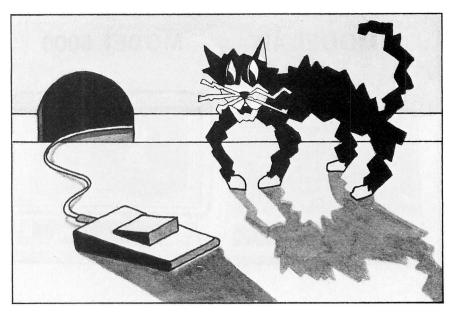
The Microsoft Serial Mouse comes with cable, driver and demonstration software, and a copy of PC Paintbrush (a color clone of Macintosh's MacPaint graphics program) for $195. As the name implies, it plugs into a serial port, a 9-pin connector like that on an IBM PC AT or Tandy 3000. The supplied cable for 25-pin PC-type serial ports is female, so you'll need an additional adapter for the Tandy 1000: a 25-pin male-male gender changer (Inmac sells one for $27). Microsoft also makes a bus version mouse that requires its own adapter card.

Microsoft's mouse has a quality feel, from its heavy, rubber-coated sensor ball to the screws that secure cable attachments. The screws' heads are also rubber-coated, so you can easily tighten them with your fingers. The plastic retaining circle releases the sensor ball for cleaning with a clever sideways snap. The mouse's two buttons curve over the top to the front and can be comfortably pressed from a variety of hand positions, clicking for tactile feedback.

The combination of the coated ball and three friction-reducing Teflon bottom pads makes Microsoft's mouse quiet, smooth, and responsive. It needs only a 5-inch diameter circle for operation, which is handy if your desk surface rarely sees the light of day.

The MS mouse comes with Piano, Life, and Notepad demonstration programs. Life is fun, and works well. The more ambitious Notepad has problems

on the 1000: slow typing and patchy highlighting created when you range the mouse too fast during block selection. These problems disappear on the faster Tandy 3000. PC Paintbrush (manual included) lets you create graphics images with a variety of "tools" including brushes, shapes, fills, and erasers. It's a good demonstration of how useful a mouse can be.

The mouse software also includes programs for using the mouse with WordStar, Lotus 1-2-3, and VisiCalc. Recent versions of Microsoft business software (and Flight Simulator) have built-in support for the mouse.

A set-up program guides you through installation. It puts a copy of the 72K mouse driver program on your system disk and creates the line in your CONFIG.SYS file which sets up the driver during boot up. If you have a hard drive, the set-up program copies all the mouse software to a subdirectory.

The 222-page manual includes installation instructions; details on using the Piano, Life, and Notepad programs; and a large section on programming the mouse driver for other programs.

You can use the "Mouse Menu" language to write interfaces—similar to those provided for WordStar, 1-2-3, and VisiCalc—for existing software. This language provides cursor control and windowing menu functions. Before using an application, you run the Menu program, which interprets your interface program and converts mouse actions to keyboard input for the application. I assume Menu is memory resident and works along the lines of a keyboard macro program. You can't write an interface for programs that use graphics or already have a mouse driver.

You can also call the mouse driver from your own programs, whether they're in Assembly language, Basic, or some other high-level language. You access the

System Requirements

Tandy 1000

Program Listing. Ancient harmonies on the Tandy 1000.

```
100 VOICES$(1)="L16N45N45N45L8N45N50N42N50N42N45L1N50"
110 VOICES$(2)="L16N45N45N45L8N42N42N38N38N42N38L1N42"
120 VOICES$(3)="L16N45N45N45L8N33N26N33N38N18N33L1N26"
125 FOR I=1 TO 3:PLAY VOICES$(I):NEXT I
130 PLAY VOICES$(1),VOICES$(2),VOICES$(3)
```

End

Illustration by Russ Clower

mouse driver through software interrupt 51 (33 hexadecimal), passing four parameters in various registers. One parameter selects one of 17 mouse functions, while the meaning of the other parameters depends on the function selected. According to the manual, mice use a unit of measurement called the mickey (there are about 200 per inch).

Poor Cousin Digi

Tandy's $100 Digi-Mouse is a poor cousin to Microsoft's mouse. They both have two buttons and use the same software interface (interrupt 51), but the Tandy mouse is of cruder manufacture and requires its own hardware interface: either the clock/mouse board (another $100) or the Plus clock/mouse add-on (also $100) for the Memory Plus expansion board. Digi-Mouse takes out one of your serial ports (Com2, hardware interrupt 3). The only software provided is the clock/mouse driver (Mouse.SYS), the Basic piano demonstration program, and two utilities for setting the battery-powered clock.

Digi-Mouse's two small buttons produce a strong click for tactile feedback, but they limit your hand to one position and aren't particularly comfortable. One Phillips head screw secures the plastic ring on the bottom that holds in the metal sensing ball. You must remove it to clean the ball. The ball isn't rubber-coated, and four smaller ball bearings let you roll the mouse around. Even when you use it on a piece of paper as recommended (to keep the sensing mechanism clean and your desk unmarred), it makes quite a racket. A 6-foot cord with a 9-pin female connector plugs into the back of the mouse adapter board.

Tandy's mouse is only half as sensitive (80 ticks per inch rolled) as the Microsoft mouse, and requires a 10-inch-square operation area.

PC Paintbrush, which comes with the MS Mouse, put Digi-Mouse to the test. It took a noticeable effort to point the arrow cursor accurately on a given pixel. Like a Russian tractor, however, it works.

Digi-Mouse itself comes with a short, 11-page manual which briefly covers plugging the mouse in, moving it, and maintaining it, along with some technical specs on the signals the mouse puts out. The main documentation comes with the clock/mouse expansion board. In addition to instructions for installing the Plus board, installing the clock/mouse driver in CONFIG.SYS, and using the clock setting programs, there are 67 pages on programming with the 3,985-byte mouse driver.

The Tandy mouse driver uses the same interrupt and function call system as Microsoft's mouse, and even the same program snippets demonstrating how to use each function. Several calls are missing: two for light pen emulation and another for double speed.

You can use Digi-Mouse and its driver with programs designed for MS Mouse. I tried Life and PC Paintbrush and noticed one minor point of incompatibility with the Microsoft Life demo. When I pointed to an empty cell and clicked to create a cell, it was only partially filled in, depending on how far the cursor arrow pointed into the cell. The program then ran normally, treating partial cells as whole.

Details are provided for using the mouse driver with 8088 Assembly language, Basic, and other high-level languages via machine-language subroutines. The 16 mouse functions let you do things like intercept button presses, set cursor bounds, and read mouse movement. As in the Microsoft manual, several cursor definition examples are provided with graphics maps.

Batch Catch

I can't help it; I like batch files. Here are two that are both useful and instructive.

Barry Erick of Dallas, PA, points out how to get rid of the extra DOS prompt at

the end of batch files (a minor irritation). His suggestion includes a little file called CAT.BAT, which produces a catalog of the files in a directory by using the wide form of the directory command (DIR/W). It's a useful command for getting a long directory in one screen when you don't care about file sizes and creation dates (*shades of CP/M*). Just type CAT with or without a pathname following.

To get rid of the second DOS prompt, write your batch files with EDLIN or Copy CON and put the control-Z end-of-file marker at the end of the last line, rather than on a final line of its own. Here's how you'd type in CAT.BAT with EDLIN:

```
C>EDLIN CAT.BAT
New file
*I
      1:*CLS
      2:*DIR %1 /W^Z
      3:*^C
*L
      1: CLS
      2: DIR %1 /WZ^
*E
```

Even though I ended line 2 with control-Z (and enter), I still had to get out of the editing mode in line 3 with control-C. I listed the file (with L) before saving (with E) to demonstrate an oddity. When control-Z is listed during the creation session of EDLIN, it comes out backwards at the end of line 2. This doesn't create problems, just puzzlement.

Read.BAT is another short, sweet batch file. It takes advantage of More.COM (a filter utility on your DOS disk) to display text files one screen at a time. A filter program takes from another program standard output that is meant for display, processes it in some manner, and sends it to standard output (usually the screen). In Read.BAT, the pipe symbol (|) tells DOS to channel the output of the Type command into the More filter. Without More, Type would send the text file streaming past your eyes—unless you pressed control-S or the 1000's hold key to halt the display. Here's the program:

```
echo off
cls
echo %1
type %1|more
```

The %1 symbol is a variable that stands for any file name typed after Read on the command line. If you type in READ README.DOC with a DOS 2.11.22 system disk in the active drive, you'll see the file README.DOC one screen at a time; press any key to see the next screen.

Fanfare

Although GW-Basic's Sound statement only produces multiple voices (harmony) with an external speaker, you can get multiple voices on the 1000's internal speaker using the Play command. Represent each musical voice as a string separated by commas after the Play statement. Three is the maximum. I wrote a small demonstration program (see the Program Listing) which first plays the three voices separately, then all together.

Correction

I'm glad to find out I was quite wrong about there being no two-button joystick for the 1000. Both Val Manes (San Angelo, TX) and Mitch Silkotch (Bridgewater, NJ) report Tandy does indeed sell such a beast (stock number 26-3012A). ∎

Dave Rowell is an 80 Micro *technical writer specializing in MS-DOS computing. Address correspondence to him to* 80 Micro, *80 Pine St., Peterborough, NH 03458.*

All Keyed Up

SuperMacro Keys from DataDesk International combines an IBM PC AT-style keyboard with Borland International's enhancer program, SuperKey.

The keyboard, which can be installed on the PC/XT/AT and compatibles, incorporates features of the advanced AT keyboard. It has a separate numeric keypad; status lights for Cap Lock, Num Lock, and Scroll Lock; extra-wide control and return; and a return key in the standard typewriter position. The Borland program lets you customize software to the new keyboard.

SuperMacro Keys costs $129.95 and is backed by a two-year unconditional warranty. For information, contact DataDesk International, 7650 Haskell Ave., Suite A, Van Nuys, CA 91406, 818-780-1673.

Circle 574 on Reader Service card.

Quick Bucks

Learn how to get rich quick with the PLS: Personal Learning System, a program combining investment study with speed reading. The 14-lesson course covers stocks, bonds, and other investments in an instructional system designed to increase reading speed and comprehension.

The program runs on Tandy Models III and 4 as well as MS-DOS machines, and can be combined with disks on other subjects that cost $19.50.

The PLS sells for $69.50 and is available from LSR Learning Associates Inc., 707 Broad Hollow Road, Farmingdale, NY 11735, 516-293-6700.

Circle 575 on Reader Service card.

CD Encyclopedia

Activenture Corp. has announced a compact disk read-only memory (CD-ROM) hardware and software package for under $1,000.

The Personal Information

SuperMacro Keys is a keyboard/software combination.

Package combines a CD-ROM player, controller card, connecting cables, and software to access a 540-megabyte disk containing all 21 volumes of *Grolier's Academic American Encyclopedia*. The drive can be connected to any IBM PC/XT/AT or to compatibles via the controller card. Knowledge Retrieval System software lets you search all occurrences of a topic, word, or phrase in the nine-million-word encyclopedia.

The package costs $995. For more information, contact Activenture Corp., P.O. Box 51125, Pacific Grove, CA 93950, 408-375-2638.

Circle 570 on Reader Service card.

Coldware

Mindscape's Balance of Power simulates nuclear age geopolitics on the IBM PC/XT/AT and compatibles through Microsoft's Windows interface. Players assume roles as the leaders of the United States and the Soviet Union in a fictional eight-year contest to enhance their country's prestige while avoiding nuclear war.

Players can call upon a data base that includes facts about GNP, literacy rate, foreign aid, and political stability in 62 nations. Strategies are adapted to a constantly changing scenario revealed in news items.

Balance of Power is for one or two players and costs $49.95. For details, contact Mindscape Inc., 3444 Dundee Road, Northbrook, IL 60062, 312-480-7667.

Circle 577 on Reader Service card.

Laser Fare

The *1986 Laser Jet Supplies Pricing Guide* from Supplysource is offered as a time-saving resource for people who order laser jet printing equipment and supplies.

The *Guide* features information on toner, font cartridges, specialized cables, paper, and machine-specific software. A wide range of machines is represented, including the IBM Page Printer, C.Itoh Laser Jet, Hewlett-

Packard LaserJet (Plus), Xerox 4045, Genicom Laser Printer, Digital Equipment's LN-03, and DataProduct's LZR-Z600.

Corporate users can get complimentary copies by contacting Supplysource, 175 Middlesex Turnpike, Bedford, MA 01730, 800-343-4688 (617-275-6563 in Massachusetts).

Circle 579 on Reader Service card.

Tandy Apple

Diamond Computer Systems' Trackstar Apple II+ emulation board is now available to schools. It allows MS-DOS computers to run most software written for the Apple II.

The board offers an 80-column mode, RGB and composite video output, language card features, and Apple text and graphics modes. It supports parallel and serial ports while interfacing to two floppy disk drives.

Trackstar lets Tandy 1000 owners run 90 percent of the Apple software produced by the Minnesota Educational Computing Corp. It costs $375 plus shipping and is available to schools through Radio Shack National Bid Dept., 1400 One Tandy Center, Fort Worth, TX 76102, 817-390-3258.
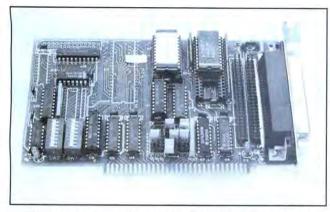
Circle 578 on Reader Service card.

Key of C

C: A Programming Workshop employs a textbook and software to acquaint you with everything from structured software to linked lists and Assembly-language interfacing in C language.

The workshop includes an integrated editor and standard compiler, along with a test module that gives instant feedback on student-written programs. The compiler conforms to the K&R standard; source code is limited to 6,000, object code to 20,000 bytes. Floats and longs aren't included.

Micro Design's card lets you add up to seven peripherals to the Tandy 1000.

C: A Programming Workshop costs $39.95 plus $3 for handling. It is available from Wordcraft, 3827 Penniman Ave., Oakland, CA 94619, 415-534-2212.
Circle 581 on Reader Service card.

Trump Card

Turner Hall Publishing offers a 256K RAM card with clock and software for the IBM PC/XT/AT or compatibles having at least 256K of memory. Tandy 1000s must have 384K already installed.

The Turner Hall Card contains 75 percent fewer chips than cards employing the earlier-generation 64K RAM chip and fewer clock/calendar parts. Less than 5 inches long, it fits in the short slot of Tandy MS-DOS computers.

Other features include clock software, a print spooler, and disk emulation. An illustrated manual lets novice users install the card themselves. There's a Help Hotline, one-year warranty, and 30-day money-back guarantee. The price is $99.95.

For information, contact Turner Hall Publishing, 10201 Torre Ave., Cupertino, CA 95014, 800-556-1234 ext. 526 (800-441-2345 ext. 526 in California).
Circle 580 on Reader Service card.

Instant Typewriter

Alpha Merics' typer is designed to turn an IBM PC/XT/AT or compatible into a full-functioning memory typewriter. It is compatible with most printers.

Typer requires no special installation. After inserting the self-booting disk, users are ready to do typing tasks too small or cumbersome for a regular word processor, such as dashing off a memo, typing a single envelope, completing individual forms, and so on.

The package is available at a $44.95 introductory price from Alpha Merics Inc., 1623 South 21 St., Hollywood, FL 33020, 305-949-8318 (-8188). *Circle 572 on Reader Service card.*

Hard Card

Micro Design International offers a 21-megabyte hard disk card for the Tandy 1000 based on the NCR Small Computer System Interface (SCSI) integrated circuit.

The card plugs into a single add-on slot in the 1000 and provides a path for users wishing to upgrade to hard disk storage. The host section of the card can connect up to six more internal or external peripherals.

The hard disk card is priced at $675. For details, contact Micro Design International Inc., 6566 University Blvd., Winter Park, FL 32792, 305-677-8333.
Circle 576 on Reader Service card.

RAM Board

The Zuckerboard Expansion Memory Board adds 512K RAM to the 128K Tandy 1000. It includes a direct memory access (DMA) chip, two banks of 256K, and a socket for installing an optional clock/calendar powered by a redundant two-

And now, a couple of words about high-quality TRS-80 software at a very low price:

THE RS-232
MODEL III MODEL 4

TRS-80 MODEL III
TRS-80 Model 4
RS-232 SERIAL INTERFACE
MODIFICATION KIT

Manual Number 5100-0002

Compulogic

State of the art technology in board design, our direct replacement of Radio Shack's internal RS-232 board, mounts inside the Model III or 4 on the existing brackets. All cables, screws and complete mounting instructions are included. Non-technical people will find that installation is quick, straight forward and simple requiring less than 15 minutes to complete.

Total compatability with Radio Shack and all existing software is maintained. Software programmable baud rates from 50 to 19,200 baud are supported along with programmable word length, stop bits, and parity. May be utilized in either half of full duplex operation.

Outstanding Value
At Only **$69.95**
Guaranteed One Full Year
Dealer Inquiries invited

Please forward payment by a cashier's check or money order.
Visa or Mastercharge also accepted.
Add $3.00 shipping & handling
(Foreign orders quoted on request)

Compulogic
4831 SOUTH HAMPTON RD LB41
DALLAS TEXAS 75232
214/330-1332

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

MS-DOS NEW PRODUCTS

The TELEcomp 2000 lets you combine color video and computer output.

battery system. The clock is accurate to plus or minus two minutes per year, and the batteries have an estimated combined life of 21 years.

The Zuckerboard sells for $149. It is available from Advanced Transducer Devices Inc., 1287 Lawrence Station Road, Sunnyvale, CA 94089, 408-734-4631.
Circle 571 on Reader Service card.

Videosynthesis

The TELEcomp 2000 from AVAS Corp. lets you add full-color video from a variety of sources to microcomputer output. You can then record or transmit the combined signal. No special software is required.

Window control lets you alter the shape and size of a computer or video image and place it anywhere on the screen. Opaque or transparent overlay selection and intensity control are available, as well as digital-to-analog conversion of RGB or composite video output.

The TELEcomp 2000M works with monochrome computer displays and costs $995. The $1,595 TELEcomp 2000C is for color computers. For details, contact AVAS Corp., 196 Holt St., Hackensack, NJ 07602, 800-631-0868 (201-487-6291 in New Jersey).
Circle 573 on Reader Service card.

MS-DOS New Products Index

Reader Service Number	Company	Page
570	Activenture Corp.	94
571	Advanced Transducer Devices Inc.	95
572	Alpha Merics Inc.	95
573	AVAS Corp.	98
574	DataDesk International	94
575	LSR Learning Associates Inc.	94
576	Micro Design International Inc.	95
577	Mindscape Inc.	94
578	Radio Shack National Bid Dept.	94
579	Supplysource	94
580	Turner Hall Publishing	95
581	Wordcraft	94

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.

The Modular Way

In last month's column, I outlined some abuses of Basic's GOTO command and advised programmers to limit use of it to two circumstances: for directing program flow to a module and for controlling flow within a module. For the purposes of that discussion, I defined a module as a "chunk of logical functions and subroutines."

As promised, I will focus this month on the subject of modules and their usefulness in Basic programs. Let's begin by considering where modules fit in the process of writing a program.

Opposite Approaches

Most Basic programs are written without much thought given to structure and overall design. Typically, programmers begin writing without an outline in front of them. As they write, they discover the need for certain functions and respond by coding subroutines and renumbering the program as necessary. Later, they may try to patch the subroutine so other parts of the program can use it.

This method leads to an excess of conditional logic and a good deal of thrashing around while writing the program. What's worse, since the programmers give little thought to the general applicability of the routines they write, they will have problems later in making variables match the ones expected in the original subroutine.
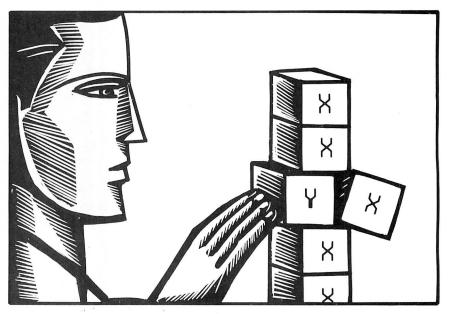
Contrast this method with what I do.

When I sit down to write a program, I give serious thought to what I want to accomplish. When I reach a section that requires a function or subroutine, I ask myself these questions:
- Is this something I'll use only once in the program? Will I need something similar in another section of the program?
- Is this function or subroutine something I'll use only once in the program? Will I need something similar in another section of the program?
- Have I written something like this in

the past? If so, how tricky was it to write and debug?
- Is it important that the routine run fast? Or, is the amount of memory it requires more important than speed?

If I'm sure the routine will be used only once, then I'll place it exactly where the need occurs. If, on the other hand, I think I might use a similar routine elsewhere in the program, then I study the last three points very carefully.

The first issue I consider is generality. Suppose I want to code a routine for retrieving keyboard input, where a "Y," "y," "N," or "n" would be acceptable. If the input is none of these, I want the routine to supply an error message.

Considering the matter further, I realize I could make my program smaller and easier to write if the routine returned only the uppercase form of the answer. This would require less checking and reduce the number of lines I'd have to write.

Clearly, this is something I might use repeatedly in a typical program. It's also something I have probably written before. The question is, should I code it in-line?

The answer, in many cases, is no. This routine is a perfect example of a general-purpose subroutine, which ought to be easy to remember and use. To ensure that they are, I always place general-purpose subroutines at the beginning of my programs. Routines I've written before

always go at the very beginning; routines written for the particular program go at the end of the subroutine block. This makes debugging easier, since I am more sure that routines I have tested previously will run correctly.

I also make sure that variables in general-purpose subroutines don't conflict with those in a main program. To guarantee that they will not, I always begin the subroutine variables with the letter "X" and avoid using this letter for main program variables.

In my programs, lines 10–5000 are reserved for previously written (general-purpose) subroutines and lines 5001–9999 for other subroutines. This way, when I have to renumber the program, I can begin with line 10000. All subroutines will retain the same line numbers, making them easier to remember and document permanently.

My programs begin with line 1, which defines variable types. Lines 2–8 contain DIM statements and user-defined functions. Line 9 always says GOTO 10000.

Program Listing 1 is an example of a general-purpose subroutine I might write. To call the subroutine in my program, I might write code like that in Program Listing 2.

This routine is useful if I don't need to save the answer for later use. But what if I'll need the answer later in the program? I have two options to make sure

---

System Requirements

All systems
Basic

Attention: Tandy 1000 Fans!

Introducing . . .

80 Micro's Starter Pack 1000

A computer as powerful and versatile as the Tandy 1000 calls for software to match. That's why you and your computer need 80 Micro's Starter Pack 1000. Designed exclusively for Tandy 1000 users, the Starter Pack disk contains 13 essential, educational, and entertaining programs, selected for you by the editors of 80 Micro.

You Want It . . . You Got It!

Whether you want to tame your finances, plan a schedule, perform a calculation, or write a letter, the Starter Pack 1000 can help you do it better. Check the lineup! Here's the entire list of 13 Starter Pack programs you can use and enjoy right away:

- **Letter Perfect** -- spelling checker program for ASCII files that works with Deskmate.
- **In the Marketplace** -- commodities futures trading simulator.
- **Time Keeper** -- manages several projects simultaneously, and calculates time spent on each project.
- **Clear-Cut Trends** -- draws, displays, and prints hi-res line and bar graphs.
- **Don't be Late** -- critical path scheduling brings your projects in on time.
- **Made-to-Order Sorts** -- creates custom sorts by answering 10 prompts.
- **Next Generation Software** -- recreates the agricultural experiments of Gregor Mendel.
- **Savings and Loan** -- compares payment schedules calculated by simple interest and by the Rule of 78's.
- **Putting Things Precisely** -- makes Basic programs more accurate with routines that calculate values with up to 16 digits.
- **Finding Your Roots** -- solves linear or non-linear equations for their roots.
- **To Each His Own** -- lets you set up your own data base manager.
- **Screen Save** -- memory-resident program that blanks out your screen to avoid unnecessary wear and tear.
- **MicroTab** -- tabulates the results of opinion polls.

Tandy 1000 is a registered trademark of Tandy Corporation

Docs on Disk

The Starter Pack is an easy and cost-effective way to build your library of Tandy 1000 software. Documentation? No problem! Full instructions for running the programs are contained on each Starter Pack disk. Read the docs right on the screen or print them out for reference.

Value -- in the 80 Micro Tradition

It's a terrific value. You get all 13 programs for your Tandy 1000, complete documentation, plus the confidence of ordering from 80 Micro, the #1 magazine for Tandy users. All for just $26.50 postpaid.

Place your order today. Simply return the coupon or the attached order card, or call toll-free 1-800-258-5473. (In NH, please dial 1-924-9471.)

Program Listing 1. A general-purpose input subroutine.

```
11 X$=INKEY$:IF X$<=" " THEN 11
12 X%=INSTR("yYnN",X$)
13 IF X%<1 THEN PRINT"Use only Y or N!":GOTO 11
14 IF X%<3 THEN X%=1:ELSE X%=2
15 RETURN
```

End

Program Listing 2. Code to call a general-purpose subroutine.

```
20000 GOSUB 1:ON X% GOTO 20010,20030
20010 'HANDLE X$="Y"
20020 GOTO 20040
20030 'HANDLE X$="N"
20040 'CONTINUE PROGRAM
```

End

Program Listing 3. The subroutine in Program Listing 1, modified to pass a character variable to the main program.

```
11 X$=INKEY$:IF X$<=" " THEN 11
12 X%=INSTR("yYnN",X$)
13 IF X%<1 THEN PRINT"Use only Y or N!":GOTO 11
14 IF X%<3 THEN X$="Y":ELSE X$="N"
15 RETURN
```

End

Program Listing 4. The code in Program Listing 2, modified to use the character variable in Program Listing 3.

```
20000 GOSUB 1:A$=X$:IF A$="N" THEN 20030
20010 'HANDLE X$="Y"
20020 GOTO 20040
20030 'HANDLE X$="N"
20040 'CONTINUE PROGRAM
```

End

Program Listing 5. A pared-down version of the input routine.

```
20000 A$=INKEY$:X%=INSTR(" yYnN",A$):
        IF X%<2 THEN 20000:
          ELSE IF X%<4 THEN A$="Y":
          ELSE A$="N":GOTO 20030
20010 'HANDLE X$="Y"
20020 GOTO 240
20030 'HANDLE X$="N"
20040 'CONTINUE PROGRAM
```

End

the answer is available: I can pass a character variable back to the program, or I can use the value of X% to assign a value to the variable.

Since the latter method restricts the program form I can use to call the subroutine, I'll usually use the form shown in Program Listing 3 to write the routine. The part of the program that calls the subroutine would then be written as shown in Program Listing 4.

Only if a routine must run particularly fast will I code it directly where it will be used by the program. To improve speed, I'll pare the routine down to its essentials (see Program Listing 5).

You may have discerned that the key to using general-purpose subroutines is the ability to spot general usage. This means you must have a plan in mind when you begin to write a program. If I want to write a word processing program, I'll begin with a list of general functions the word processor must have. I'll also have a list of the things it would be nice, but not necessary, to have.

Many of you probably think I'm going to suggest using a flow chart. You're wrong. How can anyone use a flow chart and still have fun programming? I use neither a flow chart nor an outline, but I do form a clear mental picture of what I want to accomplish when I begin a program. You should do the same. If a flow chart or outline will help, then use it. What matters is that you have a plan; whether it exists on paper or in your head is not important.

False Conclusions

From the foregoing, you might assume that a module is nothing more than a subroutine. Not true. A module is more encompassing than a subroutine.

Consider an inventory program. A typical inventory program performs four primary operations: finding, updating, adding, and deleting records. You might expect these operations to form four major sections of the program. In fact, they may make up four minor parts: the major actions the program performs are those that the processes of finding, updating, adding, and deleting records have in common.

For instance, the program must be able to display records regardless of which operation it is performing. Since displaying a record shouldn't depend on which action is being performed, the record display subroutine should be a module.

Furthermore, at least three of the program's four primary operations require a previous step: looking up the record. To avoid duplication, it may be necessary to look up a record before adding it. Let's sketch the series of operations you would be likely to program for the primary task of adding records:

● Ask the operator to enter the key(s) used to find the record.
● Make sure each key is a valid possible entry.
● Search for a record that matches one or more of the keys. If one of the keys does match (and if duplicate keys are forbidden), issue an appropriate error message and quit the "add record" function.
● Allow entry of additional data into the record. Again, each item must be validated.
● Write the record to the data file.
● Update the key files so the record can be retrieved.
● Return to the main program menu.

Each of these steps may require many subroutines; each subroutine might be used by other program functions, such

as "delete record" or "update record." In short, the "add a record" function is not a subroutine, but rather a collection of steps organized to accomplish a particular job in the program. That's how I define the term "module."

Muddled Modules

Organization is what defines a module, not whether it is a subroutine, a user-defined function, or something else entirely. Understanding the need for organization is extremely important if you want to create useful modules.

Suppose you write a modular inventory program and discover that a particular sequence of steps occurs in several modules. To save a few bytes, you might try setting the value of a special variable. This would allow you to do a GOSUB to the relevant section of one of the modules and check the value of the variable after the section executes. If the variable has a set value, you can then return to the original module.

In the end, you'll have a module with many ways of getting into it; some will be at the beginning, others will be in the middle. You'll also have many ways of returning.

Take my advice: never write this kind of module. If you absolutely need to save those bytes, put the section at the end of your program as a separate subroutine and call it from each module that uses those steps in that order.

A good module has one entry point (at the beginning) and one exit point. Additional exits may exist for one reason only: to handle errors. The only code in a module should be code that is relevant to it.

The worst kind of program you can write is one that is "sort of" modular, with a subroutine that should be somewhere else sitting right in the middle of a module. This practice makes a program almost impossible to read, generates unnecessary GOTOs, and makes the program run slower and use more memory.

If you pay attention to what you are doing when you write modules, you'll find they can make your programs faster, easier, and less prone to error. If you can't bring yourself to write programs this way, you'd be better off with a language that forces modularity on you. ∎

Bruce Tonkin is an independent software developer, industry critic, and author of The Creator data base manager. You can reach him at 34069 Hainesville Road, Round Lake, IL 60073, 312-223-8595.

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 — $299 Delivered
WordStar®
MailMerge®
SpellStar™
StarIndex™
Montezuma Micro CP/M®
RETAIL VALUE $1009

Bundle #5 — $79 Delivered
Electric Pencil™
Red Pencil™
Blue Pencil™
Works with TRSDOS, NEWDOS-80, DOSPLUS and MULTIDOS
(Model 1,3,4)
RETAIL VALUE $240

Bundle #2 — $333 Delivered
WordStar®
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
RETAIL VALUE $1354

Bundle #3 — $444 Delivered
WordStar®
MailMerge®
SpellStar™
StarIndex™
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
RETAIL VALUE $1699

Bundle #4 — $555 Delivered
WordStar®
MailMerge™
SpellStar™
StarIndex™
dBASE II®
TurboPascal™
Montezuma Micro CP/M®
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 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**
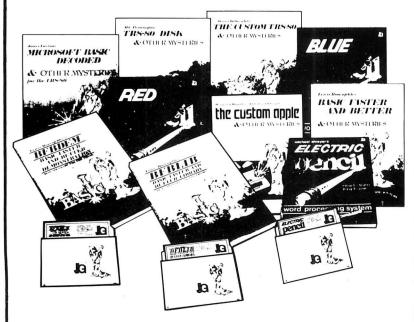
How To Do It On The TRS-80. A complete applications guide to the TRS-80 Model I, II, III, 100, and Color Computer. 352 pages. Retail $29.95 — **Now $24**

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.

©1984 by Montezuma Micro. WordStar®, SpellStar™, StarIndex™, MailMerge®, DataStar™, ReportStar™ and CalcStar™ belong to MicroPro® International Corporation. CP/M® belongs to Digital Research Inc. dBASE II® belongs to Ashton-Tate, Inc. All the Pencils™ belong to Michael Shrayer. Turbo Pascal™ belongs to Borland International.

MONTEZUMA MICRO

214-339-5104
Redbird Airport, Hangar #8
P.O. Box 32027
Dallas, Texas 75232   *"WE KEEP YOU RUNNING"*

VISA   MasterCard

ORDER NOW . . . TOLL FREE
800-527-0347
800-442-1310

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

Inside Information

Problem 1: Model 4 MEMDISK has an annoying feature. When you back up your system files to the MEM-DISK and then install the MEMDISK as disk zero with the System (System =) command, your other drives are suddenly in the wrong order. What was drive zero becomes the highest-numbered drive, and the last one TRSDOS reads when it searches for files.

You can use Floppy/DCT and System to assign any logical drive number to any disk drive, but the process can become cumbersome. You should be able to swap drive numbers cleanly and efficiently.

Problem 2: I recently added a Radio Shack hard disk to my Model 4. I wanted to write a program that would "park" the head over an area with no data before I turned off the machine. The hard disk doesn't have a parking cylinder, but I decided it would be enough to move the heads over the highest-numbered cylinder, which contains data only if the drive is filled.

My dilemma was that I use many drive configurations on my Model 4. I might have a RAM disk or my hard disk as drive zero. I often use Logical Systems' diskDISK program to open "subdisks" and assign those disks to various drive numbers. A useful parking program would have to be able to decide for itself which logical drive numbers are connected to a hard disk.

Although the two problems above might sound unrelated, their solutions are similar. Unlike less sophisticated operating systems, TRSDOS 6 builds a number of special tables in memory that contain information about the present configuration of your computer.

System Requirements

Model 4
TRSDOS 6.x
Basic
Assembly language
Editor/assembler

Program Listing 1. Program that reads the DCT and displays pertinent data.
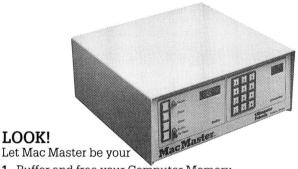
```
  1 '    Demonstration program to display information         '*    49
  2 '    stored by TRSDOS 6.x in the Drive Control Table (DCT) '*    50
  3 '                                                          '*    51
  4 '    This program uses the BASIC SVC routines presented    '*    52
  5 '    here last April                                       '*    53
  6 '                                                          '*    54
 10 DEFINT A-Z: CLS                                            '*  1119
 20 GOSUB 50100                          'Initialize SVC routines '*  792
 30 DEF FN BIT(X,Y) = ( (Y AND (2^X)) <> 0)                   '*  2305
 40 DEF FN PEEK.WORD(X) = CVI( CHR$(PEEK(X)) + CHR$(PEEK(X+1)) ) '*  3670
 50 GTDCT = 81                           '@GTDCT Supervisory Call '*  737
 60 FALSE = 0: TRUE = -1                                      '*  1299
 70 '                                                          '*   103
 95 '  Main loop -- get information about each possible         '*   110
 96 '     logical drive, 0 - 7                                  '*   111
 97 '     `                                                     '*   112
100 FOR DRIVE = 0 TO 7                                         '*  1273
110    REG.C = DRIVE                                           '*  1080
120    REG.BC = FN REG.PAIR(0,REG.C)                           '*  2025
130    REG.A = GTDCT                     'Load A with SVC number '*  1076
140    GOSUB 50000                       'Call the SVC           '*   906
150    DCT.ADDR = REG.IY                 'Get address of DCT entry '*  1349
160    GOSUB 1000                        'Display information    '*   856
165    IF DRIVE <> 7 THEN PRINT                                '*  1810
170 NEXT DRIVE                                                 '*   913
172 WHILE INKEY$<> "": WEND                                    '*  1629
175 WHILE INKEY$ = "": WEND                                    '*  1603
177 CLS                                                        '*   417
180 END                                                        '*   400
190 '                                                          '*   154
990 '   Display information about one drive                    '*   162
995 '                                                          '*   167
1000 PRINT USING "Drive #: ";DRIVE;                            '*  2303
1005 '                                                         '*   198
1010 IF PEEK(DCT.ADDR) = &HC9 THEN PRINT "Disabled": RETURN    '*  3908
1015 '                                                         '*   199
1020 DRIVER.ADDR = FN PEEK.WORD(DCT.ADDR + 1)                  '*  2761
1030 NAME.LEN = PEEK(DRIVER.ADDR + 4)                          '*  2233
1040 DRIVER.NAME$ = ""                                         '*  1253
1050 FOR LOOP = 1 TO NAME.LEN                                  '*  1766
1060    DRIVER.NAME$ = DRIVER.NAME$ + CHR$(PEEK(DRIVER.ADDR + 4 +
        LOOP))                                                 '*  4270
1070 NEXT LOOP                                                 '*   897
```

Listing 1 continued

Illustration by Paul Fisch

Listing 1 continued

```
1080 PRINT"Dvr: ";DRIVER.NAME$;" at ";HEX$(DRIVER.ADDR);TAB(29);     '*   4097
1085 '                                                                '*    206
1090 FLAG1 = PEEK(DCT.ADDR + 3)                                       '*   1770
1100 HARD = FN BIT(3,FLAG1)                                           '*   1548
1110 FLAG2 = PEEK(DCT.ADDR + 4)                                       '*   1765
1120 MAXCYL = PEEK(DCT.ADDR + 6)+1                                    '*   1990
1130 C.FIELD1 = PEEK(DCT.ADDR + 7)                                    '*   1956
1140 C.FIELD2 = PEEK(DCT.ADDR + 8)                                    '*   1959
1150 DIRCYL = PEEK(DCT.ADDR + 9)                                      '*   1897
1160 IF HARD THEN GOSUB 3000 ELSE GOSUB 2000                          '*   2643
1170 RETURN                                                           '*    713
1994 '                                                                '*    215
1995 '   Information for floppy disks                                 '*    216
1996 '                                                                '*    217
2000 PRINT "Floppy / ";                                               '*   1527
2010 IF FN BIT(7,FLAG1) THEN PRINT " WP / ";ELSE PRINT "NWP / ";      '*   3790
2020 IF FN BIT(6,FLAG1) THEN PRINT"DDEN / ";ELSE PRINT"SDEN / ";      '*   3863
2030 IF FN BIT(5,FLAG1) THEN PRINT "8"+CHR$(34); ELSE PRINT "5"+
     CHR$(34);                                                        '*   4233
2040 IF FN BIT(4,FLAG1) THEN PRINT" / 0.5 ";ELSE PRINT" / 1.0 ";      '*   3636
2050     PRINT "settle": PRINT SPC(7);                                '*   2425
2060 STEP.RATE = (-2 * FN BIT(1,FLAG1)) - FN BIT(0,FLAG1)             '*   3225
2070 PRINT USING "Step \\ ms / ";MID$(" 3 61015",STEP.RATE*2+1,2
     );                                                               '*   4020
2080 '                                                                '*    202
2090 IF FN BIT(5,FLAG2) THEN PRINT "2-side / "; ELSE PRINT "1-si
     de / ";                                                          '*   4416
2100 PHYS.DRIVE = FLAG2 AND 15: PRINT USING "pd # / ";PHYS.DRIVE
     ;                                                                '*   4074
2110 PRINT USING "### Cyl / ";MAXCYL;                                 '*   2271
2120 PRINT USING "## Sec / ";(C.FIELD1 AND 31)+1;                     '*   2769
2130 PRINT SPC(7);                                                    '*   1084
2140 PRINT USING "## gpt / ";((C.FIELD2 AND 224) \ 16) + 1;           '*   3335
2150 PRINT USING "## spg / ";(C.FIELD2 AND 31)+1;                     '*   2879
2160 PRINT USING "Dir trk ##";DIRCYL;                                 '*   2483
2170 RETURN                                                           '*    714
2994 '                                                                '*    216
2995 '   Information about hard drives                                '*    217
2996 '                                                                '*    218
3000 PRINT "Hard   / ";                                               '*   1341
3010 IF FN BIT(7,FLAG1) THEN PRINT" WP / ";ELSE PRINT"NWP / ";        '*   3727
3020 PRINT "    /  ";                                                 '*   1024
3030 IF FN BIT(5,FLAG1) THEN PRINT "8"+CHR$(34); ELSE PRINT "5"+
     CHR$(34);                                                        '*   4234
3040 PRINT                                                            '*    628
3050 PRINT SPC(7);                                                    '*   1086
3060 PRINT USING "Drv Cod # / ";FLAG1 AND 3;                          '*   2715
3080 '                                                                '*    203
3090 PRINT "DBLBIT ";: IF FN BIT(5,FLAG2) THEN PRINT " on / "; E
     LSE PRINT "off / ";                                             '*   5065
3100 PRINT USING "Head ## / ";FLAG2 AND 15;                           '*   2557
3110 PRINT USING "### Cyl / ";MAXCYL;                                 '*   2272
3115 PRINT USING "# hd / ";((C.FIELD1 AND 224)/32)+1;                 '*   3000
3120 PRINT USING "## Sec ";(C.FIELD1 AND 31)+1                        '*   2691
3130 PRINT SPC(7);                                                    '*   1085
3140 PRINT USING "## gpt / ";((C.FIELD2 AND 224) \ 16) + 1;           '*   3336
3150 PRINT USING "## spg / ";(C.FIELD2 AND 31)+1;                     '*   2880
3160 PRINT USING "Dir trk ##";DIRCYL;                                 '*   2484
3170 RETURN                                                           '*    715
49997 '                                                               '*    278
49998 'Handle an SVC call -- SVC number is in REG.A%                  '*    279
49999 '                                                               '*    280
50000 REG.AF% = FN REG.PAIR%(REG.A%,0)                                '*   2170
50001 REGISTER%(1)=REG.AF%                                            '*   1559
50002 REGISTER%(2)=REG.BC%                                            '*   1559
50003 REGISTER%(3)=REG.DE%                                            '*   1565
50004 REGISTER%(4)=REG.HL%                                            '*   1578
50005 REGISTER%(5)=REG.IX%                                            '*   1593
50006 REGISTER%(6)=REG.IY%                                            '*   1596
50007 SVC% = VARPTR(SVC%(1))                                          '*   1645
50008 REGISTERS% = VARPTR(REGISTER%(1))                               '*   2483
50009 CALL SVC% (REGISTERS%)                                          '*   1721
50010 REG.AF% = REGISTER%(1)                                          '*   1623
50011 REG.BC% = REGISTER%(2)                                          '*   1623
50012 REG.DE% = REGISTER%(3)                                          '*   1629
50013 REG.HL% = REGISTER%(4)                                          '*   1642
50014 REG.IX% = REGISTER%(5)                                          '*   1657
50015 REG.IY% = REGISTER%(6)                                          '*   1660
50016 REG.A% = REG.AF% \ 256                                          '*   1532
50017 REG.F% = REG.AF% MOD 256                                        '*   1670
50018 Z.FLAG% = ((REG.F% AND 64)=64)                                  '*   1951
50019 C.FLAG% = ((REG.F% AND 1) = 1)                                  '*   1879
50020 RETURN                                                          '*    759
50096 '                                                               '*    260
50097 '    Initialize data handler                                    '*    261
50098 '    Call this section once at the beginning of the program     '*    262
50099 '                                                               '*    263
50100 DIM REGISTER%(6),REGISTERS%,SVC%(27),SVC%                       '*   2910
50102 DIM REG.AF%, REG.BC%, REG.DE%, REG.HL%, REG.IX%, REG.IY%        '*   3616
50104 DIM REG.A%, REG.F%, REG.B%, REG.C%, REG.D%, REG.E%, REG.H%
      , REG.L%                                                        '*   4057
50106 DIM Z.FLAG%, C.FLAG%                                            '*   1497
50108 DEF FN REG.PAIR%(X%,Y%) = CVI(CHR$(Y%)+CHR$(X%))                '*   3056
50110 RESTORE 50130                                                   '*   1108
50112 COUNT% = 1                                                      '*    885
50114 READ Q%                                                         '*    717
50116 WHILE Q% <> 1                                                   '*   1047
50118    SVC%(COUNT%) = Q%                                            '*   1378
50120    READ Q%                                                      '*    778
50122    COUNT% = COUNT% + 1                                          '*   1487
50124 WEND                                                            '*    586
```

Listing 1 continued

Those tables are easily available via supervisory calls (SVCs). Once you understand how the tables are organized, you can write programs to read or change the information in them.

All three demonstration programs this month use TRSDOS 6's drive control table (DCT), which contains a wealth of information about your system's disk drives crammed into 80 bytes. Several TRSDOS commands and utilities use the DCT; programs can also access it directly to read or change information.

Inside the DCT

The DCT holds eight 10-byte records. The first record has information about drive zero, the second about drive 1, and the last about drive 7. When TRSDOS searches your disks for a file, it looks at logical drive zero first and then moves through the DCT, one drive at a time, until it finds the file. The physical order of drives, and the control program for each, is unimportant. The order of entries in the DCT determines which physical device is associated with which logical drive number.

The easiest way to find entries in the DCT table is with the @GTDCT supervisory call, which will point the IY register pair to the record associated with any drive number. From that point on, your program must be able to decipher the information in the DCT for itself. The organization of the DCT is explained in both the Model 4 *Technical Reference Manual* and Roy Soltoff's *Programmer's Guide to LDOS/TRSDOS Version 6* (now out of print).

The information in the DCT for 5-inch floppy disks is easy to understand and use. TRSDOS contains the low-level routines that read and write to floppy drives; you can depend on the DCT entries for floppy drives to be consistent.

However, TRSDOS does not contain the necessary routines for low-level access of hard drives. If you purchase a hard disk drive, you must also buy and install the necessary software to run it. Some of the fields in the DCT entries for hard disks are left to the discretion of the author of the low-level software. Program Listings 1 and 3 assume that you are using Radio Shack's hard disk software; you might have to modify the programs somewhat if your hard disk and/or its software were supplied by a third-party vendor.

Listing 1 is a short Basic program that reads the DCT and displays much of the information it contains. It uses the Basic SVC handler that I presented here last April to find the DCT entries for each logical disk drive. To get as much information as possible onto the screen, I needed to use several abbreviations.

Only Logical

The program begins by determining whether each logical drive is active. If it finds an inactive drive, it displays the word "Disabled" and then looks for the next drive.

If a drive is enabled, the program displays the module name of the software that runs the drive and the address (given in hexadecimal notation) of the entry point into that software. All normal floppy drives are run by a routine called $FD; the address of the routine depends on which version of TRSDOS 6 you are using.

Next, the program determines whether the logical disk is connected with a floppy or hard disk drive. Because this is indicated by a single bit, drives that are neither hard nor floppy (MEM-DISKs, diskDISK disks, etc.) must decide how they want to be known to the system. Most select to reset the bit so that TRSDOS recognizes them as floppy disk drives.

The program then reads the write-protect bit. If that bit is set, the logical disk is software write-protected and "WP" is displayed. Otherwise, the program displays "NWP" to show that the drive is not software write-protected.

Floppy drives can hold either single- or double-density disks. The density bit will be set according to the last successful disk access on a particular drive. The program displays the status of that bit only for floppy drives; the question of density is meaningless for hard disks, which always use the recording density that the physical hard drive and its controller were designed for.

Whenever your computer accesses a floppy drive, the software has to wait for the drive to come up to speed and the head to settle. The low-level floppy disk software in TRSDOS can allow either one or one-half second for this settling time.

Listing 1 displays this settling time as the last item on the first line of each drive specification. Although a settling time is also shown for memory disks, the MEM-DISK software is smart enough to ignore that figure and immediately access the "disk" in memory.

The information on the second display line for each drive depends on whether the DCT record is associated with a hard or floppy drive. For floppy drives, the first piece of information is the drive step rate (assuming that you have the system clock set to fast, or 4 MHz). Next is a description of whether the disk has one or two sides, and the physical drive number (abbreviated "pd") for that drive.

The disk controller in a Model 4 can

Listing 1 continued

```
50126 RETURN                                                        '*    766
50128 '                                                             '*    256
50130 DATA 9086, 28518, -6695, 8661, 0, -11803, 14835, -1575       '*   2976
50132 DATA -15887, -7727, -7715, -7683, -5159, -5319, -1031        '*   2931
50134 DATA -4135, -5159, -1549, -9749, -6659, -6691, -10779        '*   2947
50136 DATA -2619, -1575, -11781, -9759,  201                       '*   2196
50138 DATA 1,    'Dummy value to mark end of list                  '*    696
```

End

Program Listing 2. Program for swapping any two logical drive assignments.

```
00100 ;-----------------------------------------------------------
00110 ;         Model 4 -- Logical Drive Swap Utility
00120 ;
00130 ;         Use with TRSDOS 6.2
00140 ;
00150 ;-----------------------------------------------------------
00160 ;
00170 ;  SVC's used:
00180 ;
00190 @KEYIN  EQU    09H
00200 @DSPLY  EQU    0AH
00210 @EXIT   EQU    16H
00220 @GTDCT  EQU    51H
00230 @CLS    EQU    69H
00240 ;
00250 ETX     EQU    03H
00260 LF      EQU    0AH
00270 CR      EQU    0DH
00280 ;
00290 ;   Macro commands
00300 ;
00310 SVC     MACRO   #NUM
00320         LD      A,#NUM
00330         RST     28H
00340         ENDM
00350 ;
00360 PRINT   MACRO   #ADDR
00370         LD      HL,#ADDR
00380         SVC     @DSPLY
00390         ENDM
00400 ;
00410 RANGE   MACRO   #LOW,#HIGH,#NO
00420         CP      #LOW
00430         JR      C,#NO
00440         CP      #HIGH+1
00450         JR      NC,#NO
00460         ENDM
00470 ;
00480 GET1    MACRO
00490         LD      HL,KEYBUF
00500         LD      BC,2<8+0       ;B=2, C=0
00510         SVC     @KEYIN
00520         LD      A,(HL)         ;Key to A
00530         ENDM
00540 ;
00550 ;-----------------------------------------------------------
00560 ;
00570         ORG     3000H
00580 START   SVC     @CLS;
00590         PRINT   HELLO          ;Log-in message & prompt
00600         LD      IX,HOLD        ;Storage for drive number
00610 GETA    PRINT   PROMPT1        ;Ask for first drive
00620         GET1                   ;Get first drive #
00630         JR      C,LEAVE        ;Leave on <Break>
00640         RANGE   '0','7',GETA   ;Legitimate response?
00650         LD      (DRVA),A       ;Save in message
00660         SUB     '0'            ;ASCII to binary
00670         LD      (IX),A         ;And save
00680 GETB    PRINT   PROMPT2        ;Show second prompt
00690         GET1                   ;Get drive number
00700         JR      C,LEAVE        ;Leave on <Break>
00710         RANGE   '0','7',GETB   ;Test range of answer
00720         LD      (DRVB),A       ;Save in message
00730         SUB     '0'            ;Convert to binary
00740         CP      (IX)           ;Same as first?
00750         JR      NZ,SWAP        ;No -- go
00760         PRINT   DUPL           ;Else display 'no dups'
00770         JR      GETA           ;And start again
00780 ;
00790 LEAVE   LD      HL,0           ;Show 'no error'
00800         SVC     @EXIT          ;And go
00810 ;
00820 ;   2 valid replies, swap entries in DCT
00830 ;
00840 SWAP    LD      C,A            ;Drive 2 stored in C
00850         SVC     @GTDCT         ;Get DCT address
00860         PUSH    IY             ;Save address
00870         LD      C,(IX)         ;Get drive 1 in C
00880         SVC     @GTDCT         ;IY==> drv 1 address
```

Listing 2 continued

Listing 2 continued

```
00890          POP     IX            ;IX==> drv 2 address
00900          LD      B,10          ;Swap 10 bytes
00910 S10      LD      A,(IX)        ;Get byte from one DCT
00920          LD      D,(IY)        ; and byte from other DCT
00930          LD      (IX),D        ;Switch bytes
00940          LD      (IY),A
00950          INC     IX            ;Bump both pointers
00960          INC     IY
00970          DJNZ    S10           ;Repeat 10 times
00980 ;
00990          PRINT   SUCCESS       ;Report success
01000          JR      LEAVE         ;And back to TRSDOS
01010 ;
01020 ;  Messages
01030 ;
01040 HELLO    DB      'TRSDOS 6.2 drive-swap utility',LF,LF
01050          DB      'Enter numbers of two logical drives to swap.',CR
01060 PROMPT1  DB      LF,'First drive  ==> ',ETX
01070 PROMPT2  DB      LF,'Second drive ==> ',ETX
01080 DUPL     DB      LF,LF,'You must specify two different drives',CR
01090 SUCCESS  DB      LF,'Drive '
01100 DRVA     DB      '0 successfully exchanged with Drive '
01110 DRVB     DB      '0',LF,CR
01120 ;
01130 HOLD     DS      1
01140 KEYBUF   DS      2
01150 ;
01160          END     START
```

End

Program Listing 3. Disk park utility.

```
00100 ;-------------------------------------------------------
00110 ;        Hard disk PARK utility for TRSDOS 6.2
00120 ;
00130 ;   Note: This program is meant to work with drives
00140 ;         that are partitioned by head only.
00150 ;
00160 ;   If your hard disk is partitioned by cylinder,
00170 ;   you will need to modify this program before using it.
00180 ;
00190 ;-------------------------------------------------------
00200 ;
00210 ;   SVC's used:
00220 ;
00230 @KEY     EQU     01H
00240 @DSPLY   EQU     0AH
00250 @EXIT    EQU     16H
00260 @RDSEC   EQU     31H
00270 @GTDCT   EQU     51H
00280 ;
00290 ETX      EQU     03H
00300 LF       EQU     0AH
00310 CR       EQU     0DH
00320 ;
00330 ;   Macro commands
00340 ;
00350 SVC      MACRO   #NUM
00360          LD      A,#NUM
00370          RST     28H
00380          ENDM
00390 ;
00400 PRINT    MACRO   #ADDR
00410          LD      HL,#ADDR
00420          SVC     @DSPLY
00430          ENDM
00440 ;
00450 ;-------------------------------------------------------
00460 ;
00470          ORG     3000H
00480 START    PRINT   HELLO         ;Log on and ask permission
00490          CALL    YESNO         ;Get yes/no response
00500          JR      Z,FINDRV      ;If Yes -- find hard drive
00510 OUT      LD      HL,0          ;Else show 'No Error'
00520          SVC     @EXIT         ; and leave
00530 ;
00540 NODRV    PRINT   NOFIND        ;Report no hard drive found
00550 ERROUT   LD      HL,-1         ;Show error on exit
00560          SVC     @EXIT
00570 ;
00580 FINDRV   LD      C,0           ;Start with drive 0
00590 F1       LD      A,8           ;Test for end of drive list
00600          CP      C             ;Are we there?
00610          JR      Z,NODRV       ; Yes -- leave
00620          SVC     @GTDCT        ;Else IY==> drive cntrl table
00630          LD      A,(IY+0)      ;Get active/inactive status
00640          CP      0C3H          ;Is this drive active?
00650          JR      NZ,F2         ;No -- go
00660          BIT     3,(IY+3)      ;Test FLAG1 for hard disk
00670          JR      NZ,MOVHEAD    ;If hard, then go
00680 F2       INC     C             ;Else get next drive #
00690          JR      F1            ;And loop back
```

Listing 3 continued

The Model 4 disk controller can handle up to four physical floppy disk drives.

handle up to four physical floppy disk drives, known to the system as numbers 1, 2, 4, and 8. These numbers are associated with the control lines used to access the drives and do not necessarily bear any relationship to the logical drive numbers (zero to 7) that are associated with the drives.

Following the physical drive number for each floppy drive is a count of the number of cylinders (or tracks) on the drive and the number of sectors per cylinder.

For hard disks, the second line of information begins with a drive code, which is the number of the hard drive unit. The first hard drive is unit zero; if you use non-Radio Shack hard drive software, this number might be incorrect.

Following the drive code is a field which either says "DBLBIT on" or "DBLBIT off." TRSDOS's directory structure allows a maximum of 203 cylinders (or tracks) per disk. Some high-capacity hard disks, however, have more than 203 physical cylinders. If the DBLBIT flag in the DCT is on, it is a signal to the low-level hard disk software that each of the logical cylinders must be mapped onto two physical tracks. Using DBLBIT, hard disks with up to 406 cylinders per logical drive can be used.

Following the DBLBIT flag is the number of the starting drive head for this logical drive, followed by the number of logical cylinders, the number of drive heads used for this drive, and the number of sectors per cylinder. The starting head and the number of heads might be incorrect for non-Radio Shack hard drive software.

The last line of information for each drive is exactly the same for both floppy and hard disks. The first field shows the number of granules allocated for each track (GPT) on the disk; the second field shows the number of disk sectors for each granule (SPG). A granule is simply the smallest possible allocation unit on a disk. If you multiply the sectors per granule by 256, you can calculate the minimum number of bytes that can be allocated to a file. TRSDOS uses the number of granules per track to determine which track and sector should be

read to find a particular file.

The last piece of information is the logical number of the track or cylinder that contains the disk directory. That information is also stored in the first (or boot) sector of each disk. When TRSDOS logs in a new disk, it stores the directory sector number in the last byte of that disk's DCT.

The Listing 1 program is straightforward. It begins by defining two user functions that make the rest of the program easier to understand. The first tests a bit in a given byte and returns a value of true (or – 1) if that byte is set or on. The second is a 2-byte PEEK function that appears to read an entire 16-bit word from memory and return its value in an integer variable.

The information that Listing 1 displays might be interesting, since you can otherwise get that much information about your system's drives only by using a number of utilities plus some deductive reasoning and a set of calculations. The program shows what kind of information is available in the DCT and how it can be found.

Swap Talk

Listing 2 is somewhat more useful. It lets you swap any two logical drive assignments. It works by exchanging the 10-byte records of any two drives in the DCT. After the records have been swapped, the drive numbers that are used to access those drives will also have been traded.

This program solves the problem of misordered disk drives with MEMDISK. Assume that you have a two-drive 128K Model 4. If you install a MEMDISK and make it your system disk, your normal drive zero will be drive 2. To put your drives back in a logical order, you could swap drives 1 and 2, leaving you with the MEMDISK as drive zero, the bottom (or left) floppy as drive 1, and the top floppy as drive 2.

The drive-swap program holds one danger. Drive zero must contain the TRSDOS system files. If it doesn't, the computer is likely to hang up in a series of error messages. If you use Listing 2 to establish a new drive zero, be sure that the new drive contains all the system files that you will need.

The last program this month, Listing 3, is a disk park utility that moves a hard disk's heads to the inside of the disk and then enters an endless loop. After you run Park, you will need to reboot your computer to use it again.

Listing 3 assumes that you have configured your hard disk as either one single logical drive, or that you have partitioned it according to drive heads. It will work with the Radio Shack hard

disk software drivers for the Model 4.

However, if you have partitioned your hard disk by cylinders, the program will not perform a legitimate parking operation. How you modify the program to work with your system will depend on the hard disk configuration that you have created. You will probably want to move the heads to the highest-numbered track of the innermost cylinder in the hard drive. To do so will require either that the program know ahead of time how you have configured your system or that it prompt you for the number of the logical drive that uses the highest-numbered physical cylinders.

A Final Note

When your programs start tweaking the bits in the DCT, they are working at a very low level in TRSDOS. One small bug in your program can potentially destroy all the directory or data on an entire disk.

If you decide to assemble Listings 2 and 3 for yourself, I strongly suggest that you take some sensible precautions. If

you have a hard disk, turn it off or disconnect it from your Model 4 before you run your program for the first time. Use floppies that contain unimportant data. Once you are convinced that a program is working correctly, then you can turn everything back on for a final check.

These programs work properly on my system, a 500K Model 4 with a 15-meg hard disk run by Radio Shack's hard disk software and three (at the moment) floppy disk drives. Please make sure that they work correctly on your system before you entrust your important data and programs to them. ■

Write Hardin Brothers at 280 N. Campus Ave., Upland, CA 91786. Enclose a stamped, self-addressed envelope for a reply. You can also contact Hardin on CompuServe's WE-SIG (PCS-117).

Listing 3 continued

```
00700 ;
00710 MOVHEAD  LD      HL,SECBUF       ;HL==> Buffer for sector
00720          CALL    HISEC           ;Get top sector in E
00730          LD      D,(IY+6)        ;D = # of last track
00740          SVC     @RDSEC          ;Read that sector
00750          JR      Z,HOLD          ;Go if everything okay
00760          PRINT   BADSEC          ;Else print message
00770          JR      ERROUT          ;And leave
00780 ;
00790 HOLD     PRINT   DONE            ;Report success
00800          DI                      ;No interrupts during pause
00810 LOOP     JR      LOOP            ;Stay here forever
00820 ;
00830 ;    Yes/No routine
00840 ;
00850 YESNO    PRINT   YN              ;Display Y/N prompt
00860          SVC     @KEY            ;Wait for keystroke
00870          AND     0DFH            ;Force to uppercase
00880          CP      'Y'             ;Was it 'Y'es ?
00890          RET                     ;Z-flag set on Yes
00900 ;
00910 ;     Find number of last sector per logical track
00920 ;
00930 HISEC    LD      A,(IY+7)        ;Pick up Config byte 1
00940          AND     1FH             ;Strip head info
00950          LD      E,A             ;Save in E
00960          BIT     3,(IY+4)        ;Is DBLBIT on?
00970          RET     Z               ;No -- return
00980          INC     E               ;Make count relative to 1
00990          SLA     E               ;E = E * 2
01000          DEC     E               ;Make count relative to 0
01010          RET
01020 ;
01030 ;    Messages
01040 ;
01050 HELLO    DB      'Hard Disk PARK Utility',LF,LF
01060          DB      'This program will move the hard disk heads to the inner',LF
01070          DB      'track and then "freeze" the computer.  After running this',LF
01080          DB      'program, you will not be able to use the computer without',LF
01090          DB      'rebooting.',LF,LF,'Do you wish to continue ',ETX
01100 ;
01110 YN       DB      '(y/n)? ',ETX
01120 ;
01130 NOFIND   DB      'Could not find a hard drive in the system.',LF
01140          DB      'Program aborted.',CR
01150 ;
01160 BADSEC   DB      'Sector read error.',LF
01170          DB      'Program aborted.',CR
01180 ;
01190 DONE     DB      LF,LF,'Hard drive heads are parked.',LF
01200          DB      'Turn off system or reboot.',CR
01210 ;
01220 SECBUF   DS      256
01230 ;
01240          END     START
```

End

80 MICRO'S LIST of ADVERTISERS

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

How to Use *80 Micro* Program Listings

Basic program listings in *80 Micro* now include a checksum value at the end of each line. This value is the sum of the ASCII values of all characters and spaces in the line, excluding remarks. With our Checksum program, you can use these values to test the accuracy of your typing after you copy listings from the magazine. Also, for easy reading, listings are formatted so that each new program line is set off on the left.

To check your typing, follow these steps:

● Type in program code *exactly* as listed, omitting the indentations when program lines continue to a second or third magazine line, the '* characters and checksum values, and any comments after a program statement.

● Save the program in ASCII format with the command SAVE "file name",A.

● Load and run Checksum (see the Program Listing). The program will prompt you for the name of the file to be verified and give you the option of sending the line numbers and checksum values to the printer or to the screen. Enter 1 for printer, 2 for screen.

● Compare the displayed line numbers and checksum values with the check-sums shown in the listing. Find and correct errors in lines having checksum values that don't match.

—*Beverly Woodbury*
Technical Editor

Program Listing. Checksum.

```
10 'CHECKSUM/BAS by Beverly Woodbury  --  2/7/86          '*    97
20 CLEAR 1000 :CLS:PRINT@140,"VERIFY CHECKSUMS ON PROGRAM" '*  3746
30 PRINT:PRINT:INPUT "Enter name of File to verify";F$     '*  4278
40 PRINT:PRINT:PRINT "List Checksums to:"                  '*  3234
50 PRINT TAB(10) "l. Printer":PRINT TAB(10) "2. Screen"    '*  3638
60 PRINT:PRINT:PRINT TAB(30);:INPUT C                      '*  2484
70 OPEN "I",1,F$                                           '*   857
80 IF EOF(1) THEN CLOSE:END                                '*  1673
90 LINEINPUT#1,L$:L=VAL(LEFT$(L$,6))                        '*  2202
100 A=VARPTR(L$):GOSUB 210   :Q=PEEK(A)                     '*  2299
110 LS=PEEK(A+1):MS=PEEK(A+2):A=MS*256+LS:GOSUB 210         '*  3170
120 FOR K=1 TO Q:P=PEEK(A):CS=CS+P:A=A+1:NEXT K             '*  2940
130 IF CS=0 OR L=0 THEN 80                                  '*  1495
140 IF CS<100000 THEN D$="-"                                '*  1500
150 IF CS<10000 THEN D$="- "                                '*  1485
160 IF CS<1000 THEN D$="-  "                                '*  1470
170 IF CS<100 THEN D$="-   "                                '*  1455
180 IF C=1 THEN LPRINT "Line";L;D$;CS:CS=0                  '*  2693
190 IF C=2 THEN PRINT "Line";L;D$;CS:CS=0                   '*  2619
200 GOTO 80                                                 '*   723
210 IF A>32767 THEN A=(65536-A)*-1                          '*  1831
220 RETURN                                                  '*   660
```

End

READER SERVICE

This card valid until July 31, 1986.

A. How would you describe your interest in MS-DOS, IBM Compatible Systems?
1. ☐ I own an MS-DOS, IBM compatible system.
2. ☐ I intend to purchase.
3. ☐ I'm interested but have no immediate plans to purchase.
4. ☐ Not interested.

B. On a scale of 1 (no interest) to 5 (great interest), please rate your interest in seeing the following types of programs published in 80 Micro:
____ 1. Business applications
____ 2. Science/math applications
____ 3. Home management applications
____ 4. Money management applications
____ 5. Games
____ 6. Programming Utilities

C. *Excluding yourself*, how many people read your copy of 80 Micro?
1. ☐ One   3. ☐ Three   5. ☐ Five or more
2. ☐ Two   4. ☐ Four

D. What type of TRS-80/Tandy Computer do you own? Check all that apply.
1. ☐ Model I
2. ☐ Model II/12
3. ☐ Model III
4. ☐ Model 4/4P
5. ☐ Model 4D
6. ☐ Model 16/16B/6000
7. ☐ Model 100/200
8. ☐ Model 600
9. ☐ Model 1000
10. ☐ Model 1200
11. ☐ Model 2000
12. ☐ Model 3000
13. ☐ Color Computer
14. ☐ Pocket Computer

E. Approximately when did you purchase your TRS-80 computer?
1. ☐ 1977–1978   4. ☐ 1983–1984
2. ☐ 1979–1980   5. ☐ 1985–1986
3. ☐ 1981–1982

F. Do you subscribe to an information utility, such as Compuserve, Dow Jones News Retrieval, etc.?
1. ☐ Yes   2. ☐ No   3. ☐ Not now, but intend to within 12 months.

G. Do you plan to purchase another TRS-80/Tandy Computer during the next 12 months?
1. ☐ Yes   2. ☐ No   3. ☐ Don't know

H. Where do you use your TRS-80/Tandy Computer? Check all that apply.
1. ☐ At home   2. ☐ At work   3. ☐ At school

I. The articles in 80 Micro are:
1. ☐ Too simple   2. ☐ Too complex   3. ☐ Just right

J. How many purchases have you made based on an ad you saw in 80 Micro?
1. ☐ 0   2. ☐ 1–3   3. ☐ 4 or more

K. Which of the following columns do you read? Please rate them on a scale of 1 (seldom read) to 5 (always read).
____ 1. Side Tracks
____ 2. Feedback Loop
____ 3. Pulse Train
____ 4. The Art of Programming
____ 5. The Next Step
____ 6. Dave's MS-DOS Column
____ 7. Spreadsheet Beat
____ 8. Project 80
____ 9. Reader Forum
____ 10. Reviews
____ 11. New Products
____ 12. Hot CoCo

L. If you are not a subscriber, please circle 500.

M. If you would like a one year subscription to 80 Micro, please circle 501 on the Reader Service Card. Each subscription costs $24.97. (Canada & Mexico $27.97, Foreign Surface $44.97, one year only). Please allow 10–12 weeks for delivery.

Reader Service: To receive more information from any of the advertisers in this issue, circle the number of the Reader Service Card that corresponds with the Reader Service number on the ad in which you are interested. You will find numbers. Complete the entire card, stamp and drop into a mailbox. In 4–6 weeks you will hear from the advertiser directly.

1	6	11	16	21	151	156	161	166	171	301	306	311	316	321	451	456	461	466	471
2	7	12	17	22	152	157	162	167	172	302	307	312	317	322	452	457	462	467	472
3	8	13	18	23	153	158	163	168	173	303	308	313	318	323	453	458	463	468	473
4	9	14	19	24	154	159	164	169	174	304	309	314	319	324	454	459	464	469	474
5	10	15	20	25	155	160	165	170	175	305	310	315	320	325	455	460	465	470	475
26	31	36	41	46	176	181	186	191	196	326	331	336	341	346	476	481	486	491	496
27	32	37	42	47	177	182	187	192	197	327	332	337	342	347	477	482	487	492	497
28	33	38	43	48	178	183	188	193	198	328	333	338	343	348	478	483	488	493	498
29	34	39	44	49	179	184	189	194	199	329	334	339	344	349	479	484	489	494	499
30	35	40	45	50	180	185	190	195	200	330	335	340	345	350	480	485	490	495	500
51	56	61	66	71	201	206	211	216	221	351	356	361	366	371	501	506	511	516	521
52	57	62	67	72	202	207	212	217	222	352	357	362	367	372	502	507	512	517	522
53	58	63	68	73	203	208	213	218	223	353	358	363	368	373	503	508	513	518	523
54	59	64	69	74	204	209	214	219	224	354	359	364	369	374	504	509	514	519	524
55	60	65	70	75	205	210	215	220	225	355	360	365	370	375	505	510	515	520	525
76	81	86	91	96	226	231	236	241	246	376	381	386	391	396	526	531	536	541	546
77	82	87	92	97	227	232	237	242	247	377	382	387	392	397	527	532	537	542	547
78	83	88	93	98	228	233	238	243	248	378	383	388	393	398	528	533	538	543	548
79	84	89	94	99	229	234	239	244	249	379	384	389	394	399	529	534	539	544	549
80	85	90	95	100	230	235	240	245	250	380	385	390	395	400	530	535	540	545	550
101	106	111	116	121	251	256	261	266	271	401	406	411	416	421	551	556	561	566	571
102	107	112	117	122	252	257	262	267	272	402	407	412	417	422	552	557	562	567	572
103	108	113	118	123	253	258	263	268	273	403	408	413	418	423	553	558	563	568	573
104	109	114	119	124	254	259	264	269	274	404	409	414	419	424	554	559	564	569	574
105	110	115	120	125	255	260	265	270	275	405	410	415	420	425	555	560	565	570	575
126	131	136	141	146	276	281	286	291	296	426	431	436	441	446	576	581	586	591	596
127	132	137	142	147	277	282	287	292	297	427	432	437	442	447	577	582	587	592	597
128	133	138	143	148	278	283	288	293	298	428	433	438	443	448	578	583	588	593	598
129	134	139	144	149	279	284	289	294	299	429	434	439	444	449	579	584	589	594	599
130	135	140	145	150	280	285	290	295	300	430	435	440	445	450	580	585	590	595	600

Name _____ Title _____

Address _____

City _____ State _____ Zip _____

Telephone ( ) _____

80micro • June 1986 6

---

80 Micro
P.O. Box 306
Dalton, MA 01227

80 Micro
P.O. Box 306
Dalton, MA 01227

can load your program and then turn it off. In fact, I used rechargeable batteries in the drive and found them to have sufficient life in spite of a power capacity of approximately one-fourth the life of alkalines per charge.

A Skimpy DOS

The Tandy drive doesn't come with a full disk operating system (DOS). Only seven commands—Files (directory), Load, Save, Backup, Kill, Name, and Format—are provided. You enable them by running a 3K machine-language program and invoke them with the function keys. You can use these commands only through the program provided. Calling them from within your own Basic or machine-language programs would require special programming effort and a knowledge of the Tandy operating system.

If you aren't familiar with 3½-inch disks, you'll be pleasantly surprised. Unlike fragile 5¼-inch floppies, 3½-inch disks are enclosed in hard plastic and have a metal hub. A metal shutter closes over and protects the disk surface when it is out of the drive.

The manual is easy to read and to the point. It describes the drive, the disks, installation, and system setup. A section on the file management program explains the start-up procedures and commands. A third section explains the error messages and what they mean. The appendices contain specifications and a wiring diagram. It would have been nice if Tandy had included a memory map of routines so programmers could utilize and access the program from other applications.

Using the Portable Disk Drive is a pleasure. The drive is connected solidly to the computer's RS-232 interface by a small 3-foot cord. I preferred using battery power even at home, since the computer's power supply, the drive's power supply, and the drive cable create a jumble of wires. In a hotel room or elsewhere on the road, wall outlets can be in short supply, so I'd recommend carrying an extension cord for the power supplies.

Conclusion

While the Tandy drive has only a barebones operating system, third-party software developers are already planning enhancements. With more powerful software, the drive could help rejuvenate the Model 100 market.

Even without enhancements, though, the disk drive is a winner. Its low price has already attracted users who recognize a bargain, and Tandy has had problems keeping up with the demand. This is the peripheral that Model 100/200 owners have been waiting for. ■

Inner Space
by Hardin Brothers

★ ★ ★ ★

Superdisk runs on the Models I, III (48K), and 4 (64K and 128K) and requires one disk drive. Intellitech Corp., 21 Campbell Drive, Dix Hills, NY 11746. 516-462-6970. $49.95.

If you use NEWDOS/80, you may have envied the memory extension boards and RAM disk programs that run under TRSDOS 6.x and LDOS 5.x. With Intellitech's Superdisk program, you no longer have to choose between working with a restricted 64K computer and changing operating systems. Superdisk lets NEWDOS/80 users allocate all banked memory in the computer or any part of it to a very fast RAM disk.

Until you have used a RAM disk, it's difficult to imagine how much more powerful your computer will seem when it can load and save files almost instantaneously. You won't have to wait for system overlays to load before library commands are executed, or for Basic to load its overlays while it is running a program.

A Bit Confusing

At first, Superdisk seemed to be very easy to use. Once I copied the program onto my normal system disk, a single Superdisk command promptly created a 448K RAM disk in the banked memory of my 512K Model 4. It took only a couple of seconds to format this new disk with the NEWDOS Format routine, a minute or two longer to back up all of the files from my system disk to the RAM disk. Then, following the instruction manual, I ran Superdisk once again to make the RAM disk my system disk.

Next, I decided to move the programs from another disk into the RAM drive. NEWDOS/80 started to copy the files correctly but soon stopped and reported that the RAM disk directory was full. I went back through the Superdisk manual and boot-up procedure more carefully, and realized that I had a 448K disk with only 64 directory slots. Suddenly, the RAM disk did not seem nearly so useful.

I read back through Superdisk's 10-page manual several times, looking for instructions about how to increase the size of the directory. The manual is generally very clear, but it only hints that it's possible to establish PDRIVE settings for the RAM disk; it never explicitly describes how to do so.

I then went back to my NEWDOS/80 manual to find the problem. After some study and further experimentation, I finally solved the puzzle and was able to create a RAM disk with 224 directory slots.

Superdisk lets NEWDOS/80 users allocate all banked memory to a fast RAM disk.

If I used NEWDOS/80 more frequently, I would probably have realized quickly how to change the RAM disk directory size. However, a less experienced user might not even realize that it's possible to specify directory size.

A 448K RAM disk (or 960K RAM disk in a 1-megabyte computer) certainly needs more than 64 directory slots. Superdisk automatically optimizes all the PDRIVE settings for a RAM disk, except DDGA (directory size)—an unfortunate omission. You must set that parameter explicitly to make a useful RAM disk, a fact the manual doesn't mention. If you are already using four disk drives with NEWDOS/80 2.1, you will have to give up one of the drives when you use Superdisk. But that's a flaw in NEWDOS, rather than in Superdisk. With NEWDOS 2.5, the RAM disk can become any of the eight possible logical disk drives.

You can use Superdisk even if your computer has only the standard 48K or 64K of main memory. In such a configuration, Superdisk creates a small RAM disk in main memory. For special purposes, a main-memory RAM disk might be useful, but both the remaining main memory and the RAM disk will probably be too small to be used on a regular basis.

Conclusion

Superdisk seems to work fine with NEWDOS/80 2.1 and is supposed to be compatible with version 2.5 (a feature I was not able to test). It should work with any program that respects the HIMEM setting (you can place the main Superdisk module anywhere in memory) except for those few programs that bypass the DOS and directly access the physical floppy drives. One of Superdisk's nice features is that it can recover the RAM disk data after a reboot.

If you enjoy using NEWDOS/80, you should seriously consider adding Superdisk and a memory board to your computer. Any thoughts that you may have had about switching to a more powerful computer will soon disappear. ■

A Reasonable Facsimile

by John B. Harrell III

★ ★ ☆

CPEmulator runs on the Tandy 1000/1200/2000 and requires 256K, MS-DOS 2.x, and one disk drive. GTEK Inc., P.O. Box 289, Waveland, MS 39576. $199.95.

While many observers see the CP/M market as dead, there are still good reasons to run CP/M. You can find certain programs that don't exist in MS-DOS versions, and the amount of public-domain CP/M software is overwhelming.

CPEmulator and Speed Kit are a software/hardware package that lets you run many CP/M programs on an MS-DOS computer. CPEmulator is a software Z80 emulator that runs on IBM PC compatibles (including the Tandy 1000/1200/3000) and on the Tandy 2000. Speed Kit, for 8088-based computers only, provides a NEC V20 replacement CPU (central processing unit) and an 8080 software emulator based on the V20.

The package works as advertised and is easy to use, but, depending on your computer, you can pay a heavy performance penalty to run CP/M software on a non-CP/M machine.

How It Works

The package comes with only 24 looseleaf pages of documentation. Of these, six pages cover the Speed Kit, including replacing the 8088 chip with the V20.

The CPEmulator program is a snap to install: its files are contained on an MS-DOS disk and you simply copy them to a work disk or hard disk. Unlike other such programs, CPEmulator binds the emulator to the Z-80.COM file. You can use any CP/M program just as you would a regular MS-DOS program. This is the beauty of CPEmulator.

The first step is to transfer your CP/M operating system and all of its associated files to your MS-DOS disk format. The disk provided with CPEmulator contains a utility, COPYCPM, that transfers files to or from your MS-DOS disks, but it does it on the host machine. You simply invoke the utility and tell it what you want to copy.

With COPYCPM, you can specify files by name or use a wild card, as with the DOS Copy command. CP/M files with the .COM extension are copied to an MS-DOS equivalent with the .B80 extension. This prevents you from executing the program file until you have completed the next step. COPYCPM shows you directories of the target and source disks while you are setting up.

The next step is to bind the emulator software to the CP/M operating system, which is now in the .B80 file. This operation requires only one command for each CP/M program to be converted.

CPEmulator, like other CP/M systems, emulates its own version of terminal support, and you have to install your software properly for the new CP/M system. If you have never done this, it could be a considerable project. CPEmulator does give you the best of both worlds: it provides an emulated terminal and it lets you use extended video and keyboard control codes (ANSI.SYS) that can be passed directly to the DOS driver. A judicious mixture of these codes generally accomplishes the job.

How Compatible?

First of all, you have to realize you're not operating a CP/M system. The usual resident CP/M commands (DIR, REN, etc.) are not present; they're replaced by their MS-DOS counterparts. For other functions, you can easily use DOS commands in lieu of the corresponding CP/M commands.

You can't use Submit files with CPEmulator—you have to convert them to equivalent MS-DOS batch files. Similarly, the CP/M file user structure is not supported. The MS-DOS equivalent program treats all files as if they are in user number zero. There are no equivalents of a read-only or system file, and you won't get the dreaded read-only disk problems that occur with regular frequency on a CP/M system after switching disks.

The standard CP/M BDOS (basic disk operating system) function calls are supported by direct conversion to their MS-DOS equivalents or by a modest change to the way MS-DOS handles the corresponding call. The exception involves manipulation of the disk parameters, allocation vectors, and other CP/M disk related functions. These functions either return dummy values or do nothing.

The CP/M BIOS (basic input/output system) functions are handled differently. Where these functions relate to peripheral devices such as the console or printers, they are converted to their direct MS-DOS equivalent as are the BDOS functions. Disk related functions are illegal and will cause termination of the program.

One design limitation is that the Z80 N and V flags aren't supported. GTEK states in the documentation that they made this and other sacrifices of accuracy in order to gain speed. That's unfortunate, since this means some good Z80 programs won't run. For example, the KAMAS outline processor uses the V overflow flag to detect if it is executing on a Z80. The program terminates immediately after informing you that you must use a Z80 CPU.

Performance

CPEmulator won't win any prizes for speed. On my fast Tandy 2000, CP/M programs running under the emulator execute approximately four to five times slower than in native CP/M mode on a Model 4. Tandy 1000 and 1200 users can expect programs to run another two to three times slower than on the Tandy 2000.

Performance would be even slower without the Speed Kit, which replaces an 8088 CPU chip with the V20, a speedier processor that can mimic the 8088. The speed increase was from 10 to 20 percent depending on the 8088 instruction mix.

The real benefit of using Speed Kit is the special 8080 version of the emulator supplied with the kit. This emulator is only for an 8080 CP/M system and makes special use of the V20's instruction set to emulate the 8080 hardware. If your CP/M software does not require a Z80, you'll do well to make this modification.

Conclusion

I have mixed emotions about this package. On the plus side, CP/Emulator is simple to use. The COPYCPM utility is an excellent feature. Conversion of some of my treasured CP/M software was relatively painless. And the program runs on the Tandy 2000 as is.

However, CPEmulator and Speed Kit have some serious drawbacks. Unless your software is for an 8080, the converted program's performance is miserable. It takes a supercharged Tandy 2000 to get performance comparable to that on a 1 MHz Z80.

Because the documentation is limited, you have to discover many of CPEmulator's features on your own. COPYCPM has no supporting documentation. The help information is excellent, but the novice still might feel lost.

Software converted using Speed Kit didn't like memory-resident packages, especially SideKick. When the NEC V20 chip is shifted into the hardware-emulate mode, apparently it no longer thinks like a PC. This means the processor can't properly handle interrupts, which resident programs depend on.

I give CPEmulator a qualified recommendation. It isn't something that everyone needs, but if you must use CP/M software on an MS-DOS computer, it does the job. The Speedkit documentation is equally slight. If you don't feel comfortable pulling your 8088 CPU out of your machine, then don't bother with the V20.■

From Basic To 8086/8088 Assembly Language

★

From Basic to 8086/8088 Assembly Language. Harley Templeton. Softcover, 181 pp., Woodware Publishing Inc., P.O. Box 1747, Plano TX 75074. ISBN 0-915381-51-6. $12.95.

Occasionally a book is published that should have never made it through a publisher's technical review. *From Basic to Assembly Language* is just such a book. Author Harley Templeton takes the position that Assembly language can be taught by translating Basic statements into their Assembly-language equivalents. If the author were successful, then this would indeed be a novel approach. However, he falls short of his goals.

Templeton makes many erroneous statements. For example, in the first chapter he says, "The 8088 Assembly-language instruction set includes instructions that are unique to the computer and do not relate directly to any Basic statement." Obviously, he has forgotten the similarity between the Assembly-language Call instruction and the Basic GOSUB statement.

He makes another absurd statement: "Basic sometimes requires two operations to perform an operation that only has to be done once." He is referring to operations requiring one Assembly-language statement as opposed to two or more Basic statements. Templeton seems to forget that a single Basic statement when interpreted can cause the execution of hundreds or perhaps thousands of machine instructions.

The book is also full of technical errors and false assumptions: "In Basic, when you enter a GOTO or an IF THEN ELSE statement, you use a line number to tell the interpreter where the next statement to be interpreted is." Here, Templeton demonstrates his lack of knowledge of the Basic language. For example, I could use the If. . .Then. . .Else structure as follows:

```
10  A = 10
20  IF A = 5 THEN PRINT " A EQUALS
        FIVE" ELSE PRINT "A EQUALS TEN"
30  END
```

I didn't use a line number to inform the interpreter where to find the next line.

Most notably absent from the text is a chapter introducing the 8088 architecture. An Assembly-language programmer would be hard pressed to exploit any microprocessor without a full knowledge of its architecture.

The Templeton book fails both to teach Assembly language adequately and to prove that Assembly language has any type of advantage over Basic or other high-level languages. I also found the lack of any reference material, such as a table of the 8088's instruction set, to be a most unfortunate omission. Its poor organization, technical inaccuracies, and absurd premises make this book a candidate for the dog of the year award.

—Gary A. Shade

Rambo: First Blood Part II

★★

Rambo: First Blood Part II runs on the Tandy 1000/1200/3000 (256K) and requires one disk drive. Mindscape Inc., 3444 Dundee Road, Northbrook, IL 60062. $29.95.

Ever since the publication of Zork and Deadline, Infocom has owned the market for text adventure games. Some companies trying to challenge Infocom have strived to develop better command parsers or more clever plots. Mindscape, on the other hand, has slapped together games based on three commercial properties: bogeyman Stephen King's *The Mist*, the disastrous James Bond film *A View to a Kill*, and *Rambo*.

As a text program, Rambo loses the primary appeal of Sylvester Stallone's Vietnam vendetta: the sight of Stallone's mighty physique photographed against spectacular explosions. But it's considerably more realistic because Rambo can get killed in any number of ways—"There is a sudden flash of light, and then the terrible agony of hundreds of steel balls ripping into your right side. Bloody bits of your flesh cover the forest floor."

There's some appeal in portraying the homicidal hunk out rescuing POWs with only a bow, gun, knife, and a bag of salt against legions of rotten Reds. But Mindscape's software is as smart as Stallone's dialogue. Saving games is a disk-swapping chore, you can't string commands together, and some responses are simply wrong. For instance, while sightseeing with Co and Kinh, I typed "Give camera

> *Whether you think Rambo was a heroic cartoon or a witless insult, the computer game is a disappointment.*

to Kinh" and Co said "Thank you."

Inferior to Infocom games at anticipating moves and providing complex puzzles, Rambo rarely springs traps that can't be discovered in a cautious survey of the area. Reactions to dumb moves are occasionally witty ("I knew you had a limited vocabulary, Rambo, but you're really getting much worse."), but the writing is generally as flat and lurid as the bits-of-flesh sequence above, which might make you think twice about letting children play.

Whether you think the movie was a heroic cartoon or a witless insult to the struggle in Vietnam, the computer game Rambo is a disappointment.

—Eric Grevstad

Modem80

★★★★

Modem80 runs on the Model 4 (64K) and requires one disk drive. Alternate Choice, 9505 W. Brown Deer Road, Milwaukee, WI 53224, 414-355-4544. $79.95.

Modem80 is a Model 4 telecommunications program that delivers enough power for the average user and yet is simple to use. In addition to the usual gamut of telecommunications commands, Modem80 contains a host program allowing remote access to your computer.

All action takes place from the system menu, which puts you in terminal mode, ready to connect with a host system. From there you can send single lines of a file to the host (much as a telecommunications macro works), transmit or receive ASCII files, and control your printer.

From Modem80's main menu, you can change the UART parameters to match those of the host you're calling, perform DOS commands, and send and receive files using ASCII or XMODEM protocols.

The menu system is clear and easy to understand. While other Model 4 programs may offer more features, Modem80's simplicity is hard to beat. Using the program, I called several BBSes, CompuServe, and our corporate email system, changing UART parameters on the fly for each system as needed. I received mail and sent files using ASCII and XMODEM utilities with no problems.

Modem80's rudimentary host program, Host1, allows callers access to your computer if your modem is set to the answer mode. This can be handy if you need to read a file while you're away from your computer. Host1 isn't menu driven; you're greeted with nothing more than the message "Host1." From there, you can receive or send files using two programs, Type/CMD and Save/CMD, included in the Modem80 package. Also,

since Host1 resides higher in memory than Basic, you can load and run Basic programs from a remote computer just as if you were sitting at the keyboard.

Although the documentation looks unimpressive—staple-bound sheets punched for a three-ring binder—its explanations are complete. Tables and illustrations will help new users find their way through the program.

If you're new to telecommunications and don't want to spend a lot of time or money on a full-blown, complicated package, Modem80 is worth looking into.

—*Bradford N. Dixon*

Mask Maker

★ ★ ★ ★

Mask Maker runs on the Model 4 (64K) and requires TRSDOS 6.2 and one disk drive. Strachan Holdings Ltd., RR #8, Site #10, Comp #13, Prince George, BC V2N 4M6, 604-963-7927. $49.95.

Creating attractive and functional screen displays consumes a great deal of program development time. That's why video display worksheets are so useful: they let you work out the screen's design before coding a program. Mask Maker takes the worksheet idea one step further by providing what is essentially an electronic video display worksheet. With Mask Maker's full screen editor, you can quickly design as many as 18 screen displays per program. A special block-graphics editing mode allows you to spice up displays with graphics borders.

Mask Maker will convert completed displays into one of six types of program code: a standard Assembly-language source file that uses the TRSDOS 6.2 @DSPLY SVC, a compressed Assembly-language source file that uses a special video driver and a 1920-byte screen buffer, a Basic source file that uses the Print command, a self-relocating high-memory module that can be called from either a machine-language or Basic program, a high-memory module that can be called from a machine-language program, or a high-memory module that can be called from a Basic program.

While Mask Maker is fairly simple to use, the manual's poor organization greatly hinders learning about the program's finer points. To be fair, the manual does contain an adequate explanation for all of Mask Maker's functions and features. It's just that finding a particular topic is easier said than done.

Although the manual needs a good going over, Mask Maker performs superbly and is an excellent product. Anyone who develops Model 4 programs could benefit from using it.

—*Mark D. Goodwin*

HOT CoCo
for the Color Computer

Table of Contents

Piece Work

by Wayne McArthur

Photo 1. Picture 3 prior to scrambling.

If you enjoy jigsaw puzzles but have frequently been frustrated by missing pieces, perhaps you should try the computerized variety. Jigsaw, my Basic jigsaw puzzle generator, will get you started (see the Program Listing).

Although the puzzle pieces are uniform blocks, rearranging them is still challenging. You can choose from two difficulty levels, testing your skill with five pictures. To add an element of competition, I've included a timing option that keeps track of the hours, minutes, and seconds required to complete a puzzle.

Piecing It Together

When you run Jigsaw, the opening screen explains the function keys and prompts you to choose a picture number from 1 to 5. You must then select the level of difficulty: a level 1 puzzle contains 56 pieces; a level 2 puzzle has 224 pieces. Finally, you must decide whether you want to time your performance.

After a short delay, Jigsaw displays the picture you have chosen and breaks the image into the appropriate number of segments, distributing the pieces randomly on the screen. Photos 1 and 2 show picture 3 before and after scrambling.

A blinking cursor then appears in the upper-left corner of the scrambled picture. To position the piece marked by the cursor, use the left- and right-arrow keys to move to the correct spot, pressing the space bar to drop the piece into place.

Moving a piece that is not in the upper-left corner requires multiple steps. First, position the cursor over the piece you want to pick up. When you press the space bar, Jigsaw transfers the piece to the upper-left corner. You can relocate the piece on the screen using the arrow keys and the space bar.

If you want to check your progress, press the enter key. A large cursor moves through the puzzle piece by piece, pointing out any pieces that are misplaced. If you are stumped, press H (help) to get Jigsaw to show where the piece in the upper-left corner belongs.

When you successfully complete your picture, Jigsaw displays a "Puzzle complete" message and tells you how long reconstruction took (if you answered Y in response to the timer prompt).

System Requirements

16K RAM
Extended Color Basic or
Disk Extended Color Basic

Photo 2. Picture 3 after scrambling (level 1).

Variations on a Theme

For a further challenge, you can replace the pictures with other graphics routines (see the Table for an outline of program structure). Be sure to start each picture routine with:

IFQ>picture number

and end with a Return statement. You should also exercise caution in choosing your routines' variables: Jigsaw uses V\$, E, F, G, H, I, J, K, L, M, T(Z), and U(Z) to set up the program and move puzzle pieces.

If, like many puzzle fanciers, you view help as a form of cheating, you can eliminate Jigsaw's help option by deleting line 60 and removing ORA\$ = "H" from line 230. You'll then be totally on your own to solve the puzzle. ■

Write to Wayne McArthur, 1105-204 Fairmont Drive, Saskatoon, Saskatchewan S7M 4P5.

Program Listing. Jigsaw.

```
10 ' PUZZLE -BY- WAYNE MCARTHUR
20 CLSØ:PRINT@37,"COLOR COMPUTER
   PUZZLE";:B$=CHR$(128)
30 PRINT@288,"USE ARROW KEYS TO
   MOVE PIECE & ";
40 PRINT@32Ø,"< SPACEBAR > TO DR
   OP PIECE OFF.";
50 PRINT@384,"TO CHECK PUZZLE --
   --- < ENTER >";
60 PRINT@448,"FOR HELP --------
   PRESS 'H' KEY";
70 PRINT@224,"";:INPUT"CHOOSE PI
   CTURE - ENTER <1-5>";Q$:SOUND2ØØ
   ,1:Q=VAL(Q$):IFQ<1ORQ>5THEN7Ø
80 PRINT@224,"";:INPUT"ENTER DIF
   FICULTY  < 1 OR 2 >";W$:SOUND2ØØ
   ,1:W=VAL(W$):IFW<1ORW>2THEN8Ø
90 PRINT@224,"";:INPUT"DO YOU WA
```

```
NT TIMER ON < Y/N >";V$:SOUND2ØØ
,1
1ØØ CLSØ:PRINT@224,"please"B$"wa
it"B$"for"B$"picture";
11Ø IFW=1THENE=56:F=8:G=7:H=32:I
=27:J=3Ø:K=26:L=224:M=162
12Ø IFW=2THENE=224:F=16:G=14:H=1
6:I=13:J=15:K=12:L=24Ø:M=169
13Ø PMODE4,1:PCLS1:DIMD(4,4):FOR
Z=1TOH/4:CIRCLE(H/2,I/2),Z,Ø:NEX
T:GET(Ø,Ø)-(Ø+J,Ø+K),Ø
14Ø DIMT(E):DIMU(E):FORZ=ØTOE-1
15Ø T(Z)=X:U(Z)=Y:X=X+1
16Ø IFX=F THENX=Ø:Y=Y+1
17Ø NEXT:DIMB(6,6):DIMC(4,4):GOS
UB53Ø
18Ø PLAY"P1":GOSUB4ØØ:TIMER=Ø
19Ø A$=INKEY$:IFPPOINT(X+J/2,Y+K
```

```
/2)=5THENPRESET(X+J/2,Y+K/2):PP=
1
2ØØ IFTIMER>6ØØØØTHENTI=TI+(((TI
MER/6Ø)/6Ø)/6Ø):TIMER=Ø
21Ø IFA$=""THEN19ØELSEIFPP=1THEN
PP=Ø:PSET(X+J/2,Y+K/2)
22Ø A=ASC(A$):IFA=32GOSUB32Ø:GOT
Ø19Ø
23Ø IFA=13ORA$="H"GOSUB32Ø:GOTO1
8Ø
24Ø GET(X,Y)-(X+J,Y+K),B
25Ø PUT(X,Y)-(X+J,Y+K),C
26Ø IFA=9THENX=X+H ELSEIFA=8THEN
X=X-H
27Ø IFA=94THENY=Y-I ELSEIFA=1ØTH
ENY=Y+I
28Ø IFX<ØTHENX=L ELSEIFX>L THENX
=Ø
```

```
29Ø IFY<ØTHENY=M ELSEIFY>M THENY
=Ø
3ØØ GET(X,Y)-(X+J,Y+K),C
31Ø PUT(X,Y)-(X+J,Y+K),B:GOTO19Ø
32Ø IFX+Y=ØTHENGET(Ø,Ø)-(Ø+J,Ø+K
),C
33Ø PUT(X,Y)-(X+J,Y+K),B
34Ø PUT(Ø,Ø)-(Ø+J,Ø+K),C
35Ø Z=(Y/I)*F+(X/H)
36Ø XX=T(Ø):YY=U(Ø)
37Ø T(Ø)=T(Z):U(Ø)=U(Z)
38Ø T(Z)=XX:U(Z)=YY
39Ø X=Ø:Y=Ø:RETURN
4ØØ X=Ø:Y=Ø:Z=Ø
41Ø XY=RND(F)-1:YX=RND(G)-1
42Ø ZZ=YX*F+XY:XX=T(Ø):YY=U(Ø)
```

Listing continued

Photographs by Mark Corliss

The Ultimate Color Computer

Enhancements for Productivity from HJL Products

To achieve maximum productivity with your Color Computer, you have to make it as easy as possible to get information into and out of the system.

This is why we developed the HJL family of high-performance enhancements for ALL MODELS of the Color Computer.

The Keyboard - $79.95
The overwhelming favorite of serious Color Computer users worldwide, the HJL-57 keyboard has the smooth, consistent feel and reliability you need for maximum speed with minimum input errors. Includes 4 Function Keys and sample function key program. Installs in just a few minutes with no soldering.

The Numeric Keypad - $89.95
The NumberJack is a self-contained, cable-connected keypad for heavy-duty number-crunchers. Besides the number keys, it has all the cursors, symbols and math keys, including autoshifted (one-touch) ADD and MULTIPLY. Comes complete with 3-foot cable and all necessary connectors for quick and easy installation without soldering.

The Monitor Adapter - $25.95
This universal driver works with all monochrome monitors, and is easily installed without clips, jumpers or soldering (except in some later CoCo 2s with soldered-in video chips). Here's crisp, clear, flicker-free monitor output with all the reliability you've come to expect from HJL Products.

The Monitor - $89.95
The GoldStar high-resolution amber monitor brings you the monochrome display that's preferred by most computer professionals today. Once you've used it you'll never connect your computer to a TV set again. The 12-inch diagonal CRT has an etched non-glare faceplate. (Requires adapter sold below)

The BASIC Utility - $25.95
Quick Basic Plus, a high-performance programming utility, can be used with any color computer that has four function keys. 26 pre-defined BASIC statements, 10 user-defined macros at a time (you can save as many sets of macros as you like), automatic line-numbering, word wrap, global search, and instant screen dump to printer, make this software the BASIC programmer's dream come true. Comes with re-legendable 3-way reference chart. Specify disk or cassette.

The HJL Warranty
Every HJL product comes with a full, one-year warranty and the exclusive HJL 15-day unconditional guarantee (except software).

Pick a Pair & Save 15%
Now, for a limited time, we'll give you 15% off the price of any two or more products shown here. Just mention this ad when you order.

Call Now, Toll Free
1-800-828-6968
In New York 1-800-462-4891
International calls: 716-235-8358

PRODUCTS
Div. of Touchstone Technology Inc.

955 Buffalo Road • P.O. Box 24954
Rochester, New York 14624
Circle 491 on Reader Service card.

Ordering Information: Specify model (Original, F-version, or CoCo 2 Model Number). Payment by C.O.D., check, MasterCard, or Visa. Credit card customers include complete card number and expiration date. Add $2.00 for shipping, 3.50 to Canada; except monitors (call for shipping charges before ordering monitors). New York state residents add 7% sales tax. **Dealer Inquiries Invited**

Listing continued

```
430 T(Z)=T(ZZ):U(Z)=U(ZZ)
440 T(ZZ)=XX:U(ZZ)=YY
450 X=X*H:Y=Y*I:XY=XY*H:YX=YX*I
460 GET(X,Y)-(X+J,Y+K),B
470 GET(XY,YX)-(XY+J,YX+K),C
480 PUT(X,Y)-(X+J,Y+K),C
490 PUT(XY,YX)-(XY+J,YX+K),B
500 X=X/H:Y=Y/I:XY=XY/H:YX=YX/I
510 X=X+1:IFX=F THENX=Ø:Y=Y+1
520 Z=Y*F+X:IFZ<E THEN41ØELSEX=Ø
:Y=Ø:RETURN
530 IFQ>1THEN56ØELSEMODE4,1:PCL
S1
540 FORZ=1TO15ØSTEP1Ø:CIRCLE(128
,96),Z,Ø:NEXT:PAINT(123,55),Ø,Ø:
PAINT(21Ø,44),Ø,Ø
550 SCREEN1,1:RETURN
560 IFQ>2THEN74ØELSEMODE3,1:PCL
SØ
570 CIRCLE(13Ø,5),6Ø,5,1,.11,.4
580 CIRCLE(126,12Ø),9Ø,5
590 CIRCLE(126,125),45,5,2
600 CIRCLE(15Ø,125),45,5,2
610 CIRCLE(11Ø,125),45,5,2
620 PAINT(1,1),1,5
630 DRAW"BM12Ø,4Ø;C2R15U4ØH2G4H5
L2G4D37
640 PAINT(124,24),2,2
650 DRAW"BM9Ø,9Ø;C2G2ØR4ØH2Ø
660 PAINT(92,94),1,2
670 DRAW"BM+7Ø,Ø;C2G2ØR4ØH2Ø
680 PAINT(162,94),1,2
690 DRAW"BM-35,+3Ø;C2G2ØR4ØH2Ø
700 PAINT(127,124),1,2
710 DRAW"BM7Ø,15Ø;C2F2ØR3ØU1ØR1Ø
D1ØR3ØE2ØL3ØD1ØL1ØU1ØL3ØD1ØL1ØU1
ØL3Ø
720 PAINT(9Ø,16Ø),1,2:SCREEN1,1
730 COLOR5,2:RETURN
740 IFQ>3THEN83ØELSEMODE3,1:PCL
S4
750 DRAW"BM118,46;C2L1ØE2ØF2ØL1Ø
F32L18F38L26F44L16ØE44L24E38L18E
32
760 CIRCLE(128,13Ø),11Ø,2,.4,.13
,.39:PAINT(2,2),3,2:PAINT(128,36
),1,3
770 DRAW"BM124,185;C4R1ØU1ØL1ØD1
ØC2":PAINT(128,18Ø),4,4
780 FORZ=5ØTO21ØSTEP4:PSET(Z,19Ø)
:PSET(Z+2,188):PSET(Z,186):PSET(
Z,RND(18Ø)):NEXT
790 CIRCLE(1ØØ,5Ø),11Ø,2,1,.14,.
32:CIRCLE(128,2Ø),8,4
800 CIRCLE(1Ø5,4Ø),75,4,1,.12,.3
2:CIRCLE(128,2Ø),4,2:COLORØ,1
810 CIRCLE(11Ø,3Ø),45,3,1,.1,.32
:CIRCLE(16,14),32,2,1,.14,.3
820 FORZ=5ØTO21ØSTEP4:PSET(Z,19Ø)
:C=1+RND(3):IFPPOINT(Z,D)=5ORPPO
INT(Z,D)=1THENCIRCLE(Z,D),4,C:NE
XTELSENEXT:SCREEN1,Ø:RETURN
830 IFQ>4THEN1Ø4ØELSEMODE4,1:PC
LS1:SCREEN1,1:B=1ØØ:X=.6
840 IFX<2.5THENX=X+.1
850 FORZ=1TO4ØSTEP2:PRESET(255-
Z,B+RND(1+X)):PRESET(1+Z,B+RND(1
+X)):NEXT:B=B+X:IFB<178THEN84ØEL
SEZ=16
860 GET(Z,1Ø1)-(Z+16,18Ø),B
```

```
870 Z=Z+16:PUT(Z+15,1Ø1)-(Z+31,1
8Ø),B:IFZ<192THEN86Ø
880 DRAW"BM25,14Ø;C#U3R2E5R4U25H
2U5U2U6E2R8F2D6F2D5G2D25R6F2R1D1
F2G1L5G1L4H1L4G1L2H1L4G2L6H1
890 DRAW"BM2ØØ,145;E4F4G1H4F4G1H
4G2D1ØG3E3U1ØL8D3L65F1ØR75L75H15
900 DRAW"E25D9E21R4F3U1H3L4D14F1
7U5F12L12D5U5L17D5U5L21D5U5L25
910 FORZ=1TO9:PAINT(4Ø+Z,125+Z),
Ø,Ø:PAINT(16Ø+Z,148+Z),Ø,Ø:PAINT
(16Ø+Z,13Ø+Z),Ø,Ø:NEXT
920 FORZ=11TO243STEP2:LINE(1+Z,1
2)-(1+Z,1ØØ),PRESET:PSET(2ØØ-RND(
65),158+RND(2Ø)):PSET(135+RND(6
5),178-RND(2Ø)):NEXT
930 FORZ=1ØØTO2ØØSTEP2:X=RND(1Ø)
940 PSET(5Ø-RND(15),135+RND(45))
950 PSET(35+RND(15),18Ø-RND(45))
960 CIRCLE(Z,5Ø+X),Z/1Ø,Ø,1,.1,.
4
970 CIRCLE(Z,54+X),Z/1Ø,1,.4,.1,
.7:NEXT
980 FORZ=ØTO54STEP1.5:CIRCLE(8Ø,
1Ø1+Z),Z/2,1,1,.1,.4:NEXT
990 DRAW"BM45,95;C1U2E1D6H1U2L6D
2G1U6F1D2R6
1ØØØ FORZ=1TO48STEP8:PSET(RND(25
5),RND(85)):PSET(145+Z,15Ø)
1Ø1Ø CIRCLE(8Ø,45),Z/8,1:LINE(5Ø
,95)-(1ØØ,92+Z/8),PSET:NEXT
1Ø2Ø DRAW"BM1Ø,18;C#D171R235U17Ø
L234D169R233U168L232H12U191E1ØG1
ØR255H1ØF1ØU191G1ØE1ØL255
1Ø3Ø X=Ø:RETURN
1Ø4Ø IFQ>5THEN121ØELSEMODE4,1:P
CLS1
1Ø5Ø CIRCLE(14,11),1Ø,Ø:DRAW"BM1
Ø,15;CØU8F2E2F2E2D8L8
1Ø6Ø GET(Ø,Ø)-(24,24),B:PAINT(12
,12),Ø,Ø:GET(Ø,Ø)-(24,24),C
1Ø7Ø PCLS:COLOR1,Ø
1Ø8Ø FORZ=ØTO255STEP31.875:LINE(
Z,Ø)-(Z,191),PSET:NEXT
1Ø9Ø FORZ=ØTO191STEP23.875:LINE(
Ø,Z)-(255,Z),PSET:NEXT:X=1:Y=12
11ØØ FORZ=X TO8STEP2:ZZ=Z*31.875
-16
111Ø PAINT(ZZ,Y),1,1
112Ø IFY<8ØTHENPUT(ZZ-8,Y-11)-(Z
Z+9,Y+1Ø),B
113Ø IFY>13ØTHENPUT(ZZ-8,Y-11)-(
ZZ+9,Y+1Ø),C
114Ø NEXT:IFX=1THENX=2ELSEX=1
115Ø Y=Y+24:IFY<191THEN11ØØ
116Ø DRAW"BMØ,Ø;CØD191R255U191L2
55C1
117Ø X=Ø:Y=Ø:SCREEN1,1:RETURN
118Ø X=Ø:Y=Ø:V=Ø
119Ø FORZ=ØTOE-1:IFT(Z)<>X OR U(
Z)<>Y THEN122Ø
12ØØ X=X+1:IFX=F THENX=Ø:Y=Y+1
121Ø NEXT:X=Ø:Y=Ø:IFV>ØTHEN19ØEL
SE131Ø
122Ø R=X*H:S=Y*I:V=V+1
123Ø GET(R,S)-(R+J,S+K),C
```

```
124Ø PUT(R,S)-(R+J,S+K),D
125Ø SOUND25Ø-Z,1
126Ø PUT(R,S)-(R+J,S+K),C:IFA$><
"H"THEN12ØØ
127Ø R=T(Z)*H:S=U(Z)*I
128Ø GET(R,S)-(R+J,S+K),C:PUT(R,
S)-(R+J,S+K),D
129Ø SOUND25Ø-Z,1:PUT(R,S)-(R+J,
S+K),C
13ØØ X=Ø:Y=Ø:IFV>ØTHEN19ØELSE131
Ø
131Ø CLSØ:PRINT@5,"PUZZLE COMPLE
TED";
132Ø TI=TI+(((TIMER/6Ø)/6Ø)/6Ø):
```

```
HO=FIX(TI):SE=(TI-HO)*6Ø:MI=FIX(
SE):SE=(SE-MI)*6Ø:SE=FIX(SE)
133Ø IFVS="Y"THENPRINT@16Ø,"TIME
 TO COMPLETE PUZZLE WAS":PRINT@1
91,HO"HOURS"ELSE135Ø
134Ø PRINT@223,MI"MINUTES":PRINT
@255,SE"SECONDS"
135Ø PRINT@448,"";:INPUT"TRY ANO
THER PUZZLE <Y/N>";Y$:IFY$="Y"TH
ENCLEAR:GOTO2Ø
136Ø SCREEN1,1
137Ø A$=INKEY$:IFA$=""THEN137ØEL
SECLEAR:GOTO2Ø
```

End

Line(s)	Description
20–60	Information page.
70–90	Menu choices.
110	Variables for a level 1 puzzle.
120	Variables for a level 2 puzzle.
130	Cursor for the Help and Check sections.
140–160	Initial values set for T(O)–T(E) and U(O)–U(E).
190–310	Movement of puzzle pieces.
320–390	Exchange of piece positions.
400–520	Initial scrambling of puzzle.
530–550	Picture 1.
560–730	Picture 2.
740–820	Picture 3.
830–1030	Picture 4.
1040–1170	Picture 5.
1180–1260	Check for misplaced pieces.
1270–1300	Help checks for the first misplaced piece and shows where it belongs.
1310–1370	Game ends. Time readout. Prompt for a new game.

Table. Program structure.

Look Both Ways

by Jeff Grainger

System Requirements

64K RAM
Disk or Extended Color Basic
Macro-80C editor/assembler

Basic's List command is virtually useless as a programming tool. The problem is that when you invoke List, your program lines scroll off the screen so rapidly that it's impossible to read the listing. While you can freeze the screen by pressing shift-@ or set param-eters to display particular program lines, you have no control over scrolling speed. And you can't back up if you press shift-@ too late.

To overcome these restrictions, I wrote Scroll, an Assembly-language utility that gives you control over a Basic pro-

Program Listing. Scroll.

```
0001 0E00                  ORG   $7000
0002 01E9       BUF    EQU   $1E9     EQU ADDR. TABLE
0003 01EB       BUFF   EQU   $1EB
0004 01ED       COUNT  EQU   $1ED
0005 01EF       SCOUNT EQU   $1EF
0006 01F1       CHK    EQU   $1F1
0007 01F5       CHK1   EQU   $1F5
0008 01F7       BU     EQU   $1F7
0009 0088       CURSOR EQU   $88
                *THIS ROUTINE SETS SOME OF THE STARTING POINTS*
                *TO THE PROGRAM                                *
0010 7000 0F6F         CLR   $6F        SET OUTPUT TO SCREEN
0011 7002 BDA928 START JSR   $A928      CLEAR THE SCREEN
0012 7005 108E01F7      LDY   #$1F7      SET ADDR. TABLE BEGINNING
0013 7009 10BF01F7      STY   BU         KEEP TRACK OF IT FOR NOW
0014 700D 9E19          LDX   $19        GET START OF BASIC PROGRAM
0015 700F EC84          LDD   ,X         CHECK FOR VALID LINE NUMBER
0016 7011 10270089      LBEQ  EXIT       GET OUT IF NO PROGRAM
0017 7015 BF01E9        STX   BUF        KEEP TRACK OF THE START
0018 7018 1700B9        LBSR  UNCRUN     GO UNCRUN SOME LINES
                *THIS IS THE ROUTINE TO POLL FOR A KEY PRESS*
0019 701B AD9FA000 BEGIN JSR   [$A000]   CHECK FOR A KEY
0020 701F 108EFFFF      LDY   #$FFFF     CLEAR ROLLOVER TABLE
0021 7023 10BF0155      STY   $155       TO CAUSE AUTO KEY REPEAT
0022 7027 27F2          BEQ   BEGIN      IF NO KEY LOOK AGAIN
0023 7029 815E          CMPA  #$5E       IS IT THE UP ARROW
0024 702B 270E          BEQ   UP         IF SO GO DO IT
0025 702D 810A          CMPA  #$0A       IS IT THE DOWN ARROW
0026 702F 2774          BEQ   DOWN       IF SO GO DO IT
0027 7031 8103          CMPA  #3         IS THE BREAK KEY PRESSED
0028 7033 2769          BEQ   EXIT       IF SO GOTO BASIC
0029 7035 810C          CMPA  #$0C       IS THE CLEAR KEY PRESSED
0030 7037 27C9          BEQ   START      IF SO START OVER AGAIN
0031 7039 26E0          BNE   BEGIN      IF WRONG KEY GO LOOK AGAIN
                *THIS IS THE ROUTINE THAT LOOKS FOR THE ADDR. OF*
                *THE LINE NUMBER PRIOR TO THE FIRST ONE ON THE   *
                *SCREEN AND GIVES IT TO THE MAIN ROUTINE         *
                *THIS ALLOWS FOR THE SCROLL UP                   *
0032 703B 10BE01F9 UP    LDY   $1F9      GET THE TOP LINE NO.
0033 703F 10BF01E9      STY   BUF        KEEP TRACK OF IT FOR NOW
0034 7043 AEA4     A@    LDX   ,Y        GET THE NEXT ONE TO FOLLOW
0035 7045 272F          BEQ   NLX        IF IT = 0 THEN GOTO NEW LINE
0036 7047 BF01F1        STX   CHK        OTHERWISE KEEP TRACK OF IT
0037 704A BC01E9  C@    CMPX  BUF        IS IT THE SAME AS THE TOP
0038 704D 2716          BEQ   STOREY     IF SO STORE IT FOR USE
0039 704F 10AE84        LDY   ,X         IF NOT LOOK AT THE NEXT LINE
0040 7052 272A          BEQ   NLY        IF IT = 0 THEN GOTO NEW LINE
0041 7054 10BF01F5      STY   CHK1       OTHERWISE KEEP TRACK OF IT
0042 7058 10BC01E9 D@   CMPY  BUF        IS IT THE SAME AS THE TOP
0043 705C 2702          BEQ   STOREX     IF SO STORE IT FOR USE
0044 705E 20E3          BRA   A@         LOOP BACK AND LOOK TILL YOU
                *                        FIND A MATCH
0045 7060 BF01E9  STOREX STX   BUF       STORE THE ADDR. FOR USE IN
0046 7063 2006          BRA   B@         THE MAIN ROUTINE
0047 7065 10BF01E9 STOREY STY   BUF
0048 7069 2000          BRA   B@
0049 706B 8D19    B@    BSR   BUMP       GO UPDATE THE TABLE
0050 706D 8E0400        LDX   #$400      RESET THE CURSOR LOCATION
0051 7070 9F88          STX   CURSOR
0052 7072 8D60          BSR   UNCRUN     GO UNCRUNCH AND PRINT LINE
0053 7074 2050          BRA   CLEAR      CLEAR THE REST OF THE SCREEN
0054 7076 9E19    NLX   LDX   $19        RESET THE LOCATE ROUTINE BACK
0055 7078 10BE01F1      LDY   CHK        TO THE BEGINNING FOR IT TO
0056 707C 20CC          BRA   C@         START LOOKING FROM THERE IF
0057 707E 109E19  NLY   LDY   $19        NO MATCH HAS BEEN FOUND
0058 7081 BE01F5        LDX   CHK1
0059 7084 20D2          BRA   D@
                *THIS ROUTINE UPDATES THE TABLE SO THAT WE KNOW   *
                *WHAT LINES ARE ON THE SCREEN. THE FIRST AND LAST*
                *ARE THE ONLY IMPORTANT ONES                      *
0060 7086 8E01F9  BUMP  LDX   #$1F9      GET ADDR. OF TABLE
0061 7089 108E01FB      LDY   #$1FB      GET NEXT ADDR. OF TABLE
0062 708D ECA1    A@    LDD   ,Y++       GET THE NEXT ADDR.
0063 708F ED81          STD   ,X++       PUT NEXT IN THE FIRST
0064 7091 10830000      CMPD  #0000      IS THE LAST ONE CLEANED UP
0065 7095 26F6          BNE   A@         ARE WE DONE IF NOT GO AGAIN
0066 7097 8E01F7        LDX   #$1F7      RESET THE TABLE TO THE BEGIN
0067 709A BF01F7        STX   BU
0068 709D 39            RTS              GO DO THE JOB NOW
0069 709E 7F02DD  EXIT  CLR   $2DD       GET OUT OF HERE
0070 70A1 7F02DE        CLR   $2DE
0071 70A4 39            RTS
                *THIS IS THE ROUTINE TO SCROLL DOWN THE LIST*
                *AND CHECK TO BE SURE A LINE WAS PRINTED     *
0072 70A5 AE9F01F7 DOWN  LDX   [BU]       GET THE ADDR. OF LAST LINE
0073 70A9 BF01F1        STX   CHK        KEEP TRACK OF IT FOR NOW
0074 70AC 8DD8          BSR   BUMP       UP DATE THE ADDR. TABLE
0075 70AE 8E0400        LDX   #$400      RESET CURSOR TO TOP LEFT
0076 70B1 9F88          STX   CURSOR
0077 70B3 BE01F9        LDX   $1F9       GET THE NEXT ADDR.
0078 70B6 BF01E9        STX   BUF        KEEP TRACK OF IT FOR NOW
0079 70B9 8D19          BSR   UNCRUN     GO UNCRUNCH THE CURRENT LINE
0080 70BB AE9F01F7      LDX   [BU]       WAS A LINE PRINTED
0081 70BF BC01F1        CMPX  CHK
0082 70C2 2602          BNE   CLEAR      IF SO CLEAR THE REST OF THE SCREEN
0083 70C4 20DF          BRA   DOWN       IF NOT GO DO IT AGAIN
                *THIS ROUTINE IS TO CLEAN UP THE REMAINING PART OF*
                *THE SCREEN SO THAT GARBAGE OR DOUBLE IMAGING IS  *
                *NOT LEFT AT THE BOTTOM                           *
0084 70C6 9E88    CLEAR LDX   CURSOR     GET THE CURSOR POSITION
0085 70C8 8660          LDA   #$60       GET A BLANK
```

Name	Function
BUF, BUFF	Store the addresses of Basic lines in memory.
COUNT, SCOUNT	Store the length of Basic lines after uncrunching.
CHK, CHK1	Used by the Up routine to check for the previous line.
BU	Keeps track of the table of line addresses.
CURSOR	Stores the current location of the cursor.

Table. Address labels.

gram's line-by-line display (see the Program Listing). Scroll lets you move forward and backward through the listing at your own speed; you can return to the beginning of the listing or exit to Basic with a single keystroke.

Scroll 'Em

After assembling the Listing, load the Basic program you want to list, then load Scroll into memory at address 7000 hexadecimal. If you have less than 64K of RAM (random-access memory), you must offset-load Scroll to accommodate your system.

To use Scroll, type in EXEC. The up- and down-arrow keys move you up and down within the program. The listing wraps in both directions, displaying the first line directly after the last and vice versa. Press the clear key to return to the beginning of the program; to edit a line, press the break key to exit to Basic. Once you have made the necessary editing changes, type in EXEC to call up Scroll again.

At the beginning of the program, Scroll equates labels with addresses used for short-term storage (see the Table). The remainder of the program is documented with remarks to assist you in following program logic.

If you are familiar with Assembly-language programming, you might want to add other features, such as a page routine that scrolls forward (or backward) one screen at a time. You could also add a routine that returns you to your previous location when you reexecute Scroll after editing a line from Basic.■

Jeff Grainger welcomes your questions or comments on his program. Write to him at Box 2235, Bracebridge, Ontario P0B 1C0.

64K EXT. BASIC 128⁹⁵

Color Computer II

Monitor Interface for any Color Computer 29.95 plus
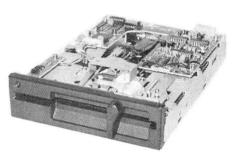7.50 installation. (color, green & amber compatible)

AMBER MONITOR 64⁹⁵

Epson's Comrex 5650 has a 12″ screen with 900 lines.
Resolution for 80 column text and 18 MHZ band width.
Retail price is 139.95. These are new, in factory sealed
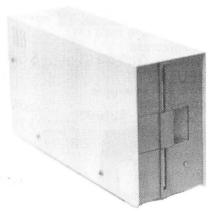cartons, NOT used, repacked, or refurbished. Add 7.00 s/h.

2 Drives 299⁹⁵

Both our drive 0 and 1 in one case, with cable and R.S.
controller. The best just got better!
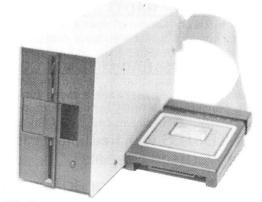
Drive 1 Upgrade 119⁹⁵

Add a second ½ height drive to your Radio Shack
26-3129. Comes with 3 minute installation instructions,
screwdriver required.

Drive 1 125⁹⁵

Your Choice
Silver or White

199⁹⁵ Drive 0

SUPER DRIVE SALE

Special prices on new first quality disk drives. They even have GOLD connectors on the back...Some other places charge 229.00 for
dr. 1 and 299.00 for dr. 0, not us! Drive 1 is for mod I, Second Color Computer drive, or external mod III, IV. Drive 1 just plugs into the
extra connector on your Drive 0 cable. Both drives are compatible with any version of the Color Computer and all versions of drives.
Drive 0 is your first Color Computer drive and comes complete with cable, manual, and R.S. controller. For double-sided, add 45.00
(only for those who have DS-DOS, boards and knowledge) Bare full hgt SSDD drive only 79.95.

THE COMPUTER CENTER
901-761-4565, 5512 Poplar, Memphis, TN 38119
Add $4.90 for shipping and handling—Visa, MC & money orders accepted
Allow an additional 2 weeks for personal checks—Drive faceplates may vary slightly
Prices subject to change without notice.
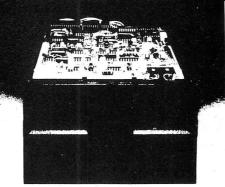
New Hard Drives

---------- COMPLETE SYSTEM ---------- JUST PLUG IN ----------

Call For
BEST PRICE

Warranty - One Full Year

5 to 20 Megabyte, ready to run on the TRS 80 Model I/III/IV/4P, color computer,

64K UPGRADES $39⁹⁵

NEW QUALITY VIDEO MONITORS

Starting at **$79.⁹⁵**

MONOCHROME
COLOR MONITORS

VIDEO DRIVER
ENABLES YOUR COCO TO OPERATE WITH A VIDEO MONITOR INSTEAD OF A TELEVISION!

SCREEN DUMP PROGRAM

The best screen dump program for the Epson & Gemini printers ever !! Have the option of standard images or reverse w/regular or double sized pictures **$19⁹⁵**

star™ PRINTERS
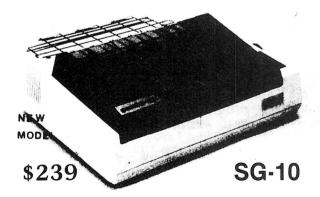
PRINTER CABLES AND INTERFACES AVAILABLE
Call for current pricing

- 100 - 120 - 160 CPS
- Bidirectional Logic Seeking
- Friction and Tractor
- 9X9 Dot Matrix
- True Decenders
- High Res-Bit Image Block Graphics

- Super Script-Subscript
- Underlining
- Backspacing Doublestrike
- 5, 6, 8 1/2, 10, 12 and 17 Pitch
- Programmable Line Spacing
- SIX (6) MONTH WARRANTY

GEMINI 10X (9 Inch Carriage, 120cps) Friction and Tractor . . . . . $CALL
GEMINI 15X (15 Inch Carriage, 120cps) Friction and Tractor . . . . $CALL
DELTA 10 (10 Inch Carriage, 160cps) Friction and Tractor . . . . . . $CALL
POWER TYPE Letter quality

SP-3 INTERFACE for Color Computer
- 300-19,200 BAUD rates
- External to printer — No AC Plugs
- Built in modem/printer switch—no need for Y-cables or plugging/unplugging cables

Only:
~~$59.95~~

$239   **SG-10**

COMPLETE SYSTEM
~~299~~⁹⁵ **NEW LOW PRICE** now with screen dump

Nothing more to buy!

Dealer inquiries invited

TRUE DATA PRODUCTS
9 S. Main
Uxbridge, MA 01569

CALL US TODAY!!
ORDER TOLL FREE

(617) 278-6555
1-800-635-0300

Doctor ASCII

by Richard E. Esposito,
Richard W. Libra, and Raymond W. Rowe

Having technical difficulties? Consult the Doctor for an answer. Due to the volume of mail Doctor ASCII receives, we can't guarantee publication of your query. Please send a self-addressed, stamped envelope with all letters to Doctor ASCII, c/o 80 Micro, 80 Pine St., Peterborough, NH 03458.

Q: Is there a CGP-220 Jet Printer dump I can use with CoCo Max? Also, many BBSes here in San Diego are going to 1,200 or 2,400 baud. Other than price, what should I look for in a high-speed modem, and where can I get one? *(Gary Bradbury, Fallbrook, CA)*

A: The only high-resolution screen dump I know of for your printer is Radio Shack's Color Screen Print Utility (catalog number 26-3121, $9.95).

If price is no object and you want a real Cadillac terminal for your CoCo, get a Hayes-compatible modem; PBJ's 80-column card, Word-Pak (P.O. Box 813, N. Bergen, NJ 07047, 201-523-8663); or Cer-Comp's terminal software, Data Pak II (5566 Ricochet Ave., Las Vegas, NV 89110, 702-452-0632). Check the ads in *80 Micro* or *Computer Shopper*, a monthly newspaper with lots of ads.

Q: I can't seem to load Basic programs typed in from Scripsit. What am I doing wrong? *(Kelly Butler, Mebane, NC)*

A: Scripsit saves files in a continuum; Basic saves them as discrete records. To get around the problem, use option 2 of Scripsit's print menu to create a spool file, which stores the program in ASCII format.

Q: I have a 64K CoCo 2 with Extended Color Basic and a Radio Shack disk drive. I need as much free memory as possible for a voice buffer. Can you supply a memory map of POKE addresses and a description of the map in all-RAM mode? *(Stephen Sabulak, Farmington, CT)*

A: Spectral Associates (3418 S. 90th St., Tacoma, WA 98409, 206-581-6938) markets a series of manuals containing the disassemblies of Color Basic, Extended Color Basic, and Disk Extended Color Basic as well as the empty RAM spaces. The complete set is $32.95; individual manuals are $14.95 (prices don't include shipping and handling). If you need a lot of extra memory, consider J&R Electronics' 512K Banker (P.O. Box 2572, Columbia, MD 21045, 301-987-9067 or 301-788-0861). The bare board (etched and drilled) is $34.95; an assembled and tested version is $149.95.

Q: Can I reduce blank spaces in Basic programs to minimize memory use? *(Ricky Sutpin, Henry, VA)*

A: With the exception of the spaces that appear after line numbers when you invoke the List command, each space in a Basic program consumes 1 byte, so reducing spaces can save memory. However, since each new line uses 7 extra bytes, the best way to compact a program is to eliminate line numbers by creating multi-statement lines.

Q: I like my Star SG-10 printer, but I can't get the download character set or sample program in the SG-10 manual to work. I have a 64K upgraded E board (fall of 1981) and a Botek serial-to-parallel interface. Because the Botek has a jumper for 7-bit and 8-bit modes, I put in a switch so that I can print out text

Program Listing. Tapefix.

```
10 GOTO480
20 CLS
30 PRINT@0,STRING$(32,"*")
40 PRINT@32,"*"+STRING$(30,"")+"*"
50 PRINT@64,"* MACHINE LANGUAGE CASSETTE TO  *"
60 PRINT@96,"*          DISK UTILITY          *"
70 PRINT@128,"*     COPR 1982 BY              *"
80 PRINT@160,"*          RICHARD E. ESPOSITO  *"
90 PRINT@192,"*          & RALPH RAMHOFF       *"
100 PRINT@224,"*        << 32K RAM >>          *"
110 PRINT @256,"*    DISK COLOR BASIC 1.0       *"
120 PRINT@288,"*"+STRING$(30,"")+"*"
130 PRINT@320,STRING$(32,"*")
140 PRINT"IS TAPE READY?"
150 A$=INKEY$:IF A$="" THEN 150
160 IF LEFT$(A$,1)<>"Y" THEN PRINT "READY THE TAPE, THEN RUN
AGAIN":END
170 INPUT"NAME OF TAPE FILE CONTAINING PROGRAM";F$
180 IF LEN(F$)>8 THEN PRINT"too big":GOTO 170
190 CLOADMF$,&H2000
200 SL=PEEK(487)*256+PEEK(488)
210 EN=PEEK(126)*256+PEEK(127)-1
220 EX=PEEK(157)*256+PEEK(158)-&H2000
230 NN=EN+42: XX=EN+1
240 FOR I=EN+1 TO NN
250 READX$: X=VAL("&H"+X$)
260 POKEI,X
270 NEXT I
280 ST=SL+H2000
290 S1=INT(SL/256): S2=SL-S1*256
300 N7=EN+27:POKEN7,S1: N8=EN+28:POKEN8,S2
310 EL=EN-&H2000
320 S3=INT(ST/256): S4=ST-S3*256
330 POKEEN+23,S3: POKEEN+24,S4
340 E3=INT(EN/256): E4=EN-E3*256
350 POKEEN+34,E3: POKEEN+35,E4
360 X1=INT(EX/256): X2=EX-X1*256
370 POKEEN+41,X1: POKEEN+42,X2
380 INPUT"NAME OF FILE ON DISK";F$
390 IF LEN(F$)>8 THEN PRINT"too big";GOTO380
400 SAVEMF$,ST,NN,XX
410 SOUND125,3:PRINT"READY":END
420 DATA4F,B7,FF,40,86,34,B7,FF
430 DATA03,8E,04,00,86,80,A7,80
440 DATA8C,06,00,2D,F9,8E,26,00
450 DATA10,8E,06,00,A6,80,A7,A0
460 DATA8C,47,10,2D,F7,0F,71,7E
470 DATA18,38
480 PCLEAR1:GOTO20
```
End

and graphics. Can you help me overcome the problem? *(Daniel Jameson, Hollywood, CA)*

A: If you are still running Color Basic 1.0 (type in EXEC 41175 to find out), get Color Basic 1.1 or 1.2; your printer will then work in 8-bit mode with text and graphics. Version 1.0 has only a 7-bit nongraphics printer driver.

Q: My friend and I are having trouble with CoCo I's (with the original "chiclet" keyboard) that have been upgraded to 64K. Both machines have begun to freeze up or print garbage on the screen.

The trouble isn't related to the length of time they've been on. If we press the reset button, the problem clears up, but memory has been corrupted. Dennis Kitsz' RAM test routine gives them a clean bill of health. Do you know what the problem might be? *(Chris Rogers, Oakville, Ontario)*

A: Tandy has issued a service bulletin on the problem. You can have Tandy fix it or order the bulletin and parts from Radio Shack National Parts (900 E. Northside Drive, Fort Worth, TX 76102). Ask for Technical Bulletin CC:20 and the Final Fix Kit,

which includes a 26-3003B logic board AX-7089, SN74LS12 triple-input NAND open collector IC, and SN74LS393 dual 4-bit binary counter. "Final Fix" (*HOT CoCo*, February 1986, p. 32) offers an alternative that involves drilling holes in the CoCo's cover to vent heat from the pass transistor.

Q: My gray CoCo behaves as if it has a bad PIA chip: keystrokes won't register in columns A, B, and C, even when I change the keyboard.

I have a basic understanding of electronics, but I'm not familiar with digital circuitry. What should I check to find out if the problem is something other than the PIAs? I don't know the board version, but the model number is 26-3004A. What schematic should I request?

Also, I've been thinking about upgrading my CoCo with The Banker or Thunder RAM. Are there any differences between them?

Are they compatible with peripherals and Radio Shack software? Someone from Radio Shack said the upgrade might overheat my CPU. Is this true? *(Pete Vitale, Riverside, NJ)*

A: The manual for your machine, TM 26-3004A, is available from

Radio Shack National Parts (see address above). Pay close attention to the circuitry that interacts with the keyboard when you check for problems.

I use J&R Electronics' The Banker. It doesn't require connections to the cartridge port or test points. You establish connections by removing the SAM chip, plugging the Banker board into its socket, and replacing the 64K chips with prewired 256K chips.

None of the CoCos, except the newest Korean models, require soldering or installation; however, you must run wires to pin 1 of each 256K RAM chip. If you call J&R for support, you'll talk to a person, not to an answering machine. All software, including the OS-9 RAM disk, is included at no extra charge.

Spectrum Projects' Thunder RAM is currently available for the original CoCo only. Its OS-9 RAM disk costs extra and its built-in ROM can create compatibility problems with standard CoCo software.

Overheating should not be a problem: the 256K chips draw less power, so your CoCo should run cooler.

Q: Like Milagros Rivera Diaz (*HOT CoCo*, August 1985, p. 15), I had a problem with the Gorilla Banana

From Computer Plus to YOU...

PLUS after PLUS after PLUS

Tandy 200 24K $649
Model 600 32K $1269
Model 100 24K $425

Tandy 3000 $1969
Tandy 3000 HD $2699

Tandy 1000 $685
Tandy 1000 HD $1539
Tandy 1200 HD $1599

DMP-130 $269

Color Computer II
w/64K Ext. Basic $165

DMP-105 $160

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS
Tandy 1000 1 Drive 128K	685.00
Tandy 1000 HD 10 Meg. 256K	1539.00
Tandy 1200 10 Meg. 256K	1599.00
Tandy 3000 1 Drive 512K	1969.00
Tandy 3000 HD 20 Meg. 512K	2699.00
Model IVD 64K with Deskmate	889.00

PRINTERS
Radio Shack DMP-130	269.00
Radio Shack DMP-430	660.00
Radio Shack DWP-220 Daisy Wheel	469.00
Radio Shack TRP-100 Portable	229.00
Silver Reed EXP-500 Daisy Wheel	229.00
Star SG-10	245.00
Star SG-15	410.00
Star SD-10	365.00
Panasonic P-1091	259.00
Panasonic P-1092	339.00
Toshiba 1340	439.00
Okidata 192	375.00
Epson LX-80	245.00
Epson FX-85	419.00
Epson HS-80 Portable Ink Jet	339.00

Tandy 1000/1200 ACCESSORIES
Tandy 1000 Disk Drive Kit	159.00
Tandy 1000 10 Meg. Hard Drive	579.00
Hard Drive Controller Board	249.00
256K Ram Board (inc. 128K & DMA)	199.00
512K Ram Board (includes 128K)	169.00
256K Memory Plus Expansion Brd.	249.00
PBJ Multi-Function Board (128K)	259.00
PBJ Multi-Function Board (256K)	279.00
PBJ Multi-Function Board (512K)	299.00
128K Ram Upgrade Kit (NEC)	78.00
256K Ram Upgrade (for PBJ Board)	69.00
300 Baud Modem Board	129.00
1200 Baud Modem Board	249.00
RS-232 Serial Board	89.00
Digi-Mouse/Clock Board	89.00

Tandy 1000/1200 MONITORS
Tandy VM-2 Green Monitor	129.00
Tandy CM-2 RGB Color Monitor	379.00
Tandy CM-4 RGB Color Monitor	249.00
Tandy VM-3 TTL Green Monitor	179.00
Tandy 1200 Text Monitor Adapt.	219.00
Tandy 1200 Color/Graphics Adapt	240.00

MODEMS
Radio Shack DCM-3 Modem	52.00
Radio Shack DCM-5 Modem	99.00
Radio Shack DC Modem 2212	315.00
Hayes Smartmodem II 300 Baud	169.00
Hayes Smartmodem 1200 Baud	429.00
Hayes Smartmodem 1200B	415.00

FOR EVERYONE
Network Four Outlet Surge Prot.	69.95
CCR-81 Cassette Recorder	52.00
CCR-82 Cassette Recorder	43.00
C-20 Digital Cassette Tapes (10pk)	9.95
Verbatim SSDD Plastic Box (10pk)	24.95
Verbatim DDDD Plastic Box (10pk)	29.95
Precision SSDD Diskettes (10pk)	15.95
Precision DDDD Diskettes (10pk)	17.95
NEC 64K Ram Chips (set of 8)	39.00
NEC 256K Ram Chips (set of 8)	69.00
Flip N File Disk 50	15.00
Fanfold CleanEdge Paper (2600)	35.00

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 · AMERICAN EXPRESS · MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

printer when I typed in PRINT #–2. To set things right, I got the correct serial cable (#4109, $18 plus $2 shipping and handling) and replacement chip (free) from DAK Industries (8200 Remmet Ave., Canoga Park, CA 91304, 800-325-0800). The chip and DIP switch settings are important because they coordinate the baud rate (600) between the computer and printer.

A: Thanks for the information.

Q: In the January 1985 issue of *HOT CoCo*, your column (p. 78) had a program for accessing the full 64K RAM. I have a disk drive with a J&M controller (JDOS version 1.06). Whenever I typed in a command, I got a syntax error, but when I ran the program under Extended Color Basic, it worked. What can I do to use the program with JDOS?

A: Remove the JDOS ROM and replace it with a Radio Shack ROM—a lot of software won't run without it. Howard Medical Computers (Box 2, Chicago, IL 60690, 800-443-1444) sells the Radio Shack DOS ROM for $20.

Q: Back issues are no longer available for the September 1983 issue of *HOT CoCo*. I would like a copy of your Tapefix program so that I can convert my cassettes to disk. *(SFC Frank Hoegler, APO, NY)*

A: See the Program Listing on p. 132 for a reprint of Tapefix. ■

Color Monitor

by Scott Norman

Regular readers of this column know I'm intrigued by integrated programs that share information and let me skip back and forth between tasks. Large personal computers have had this ability for some time; I use integrated programs with the 512K Macintosh in my office every day. The real problems begin when you try cramming all the necessary features into 64K or less.

Nevertheless, people are trying, including Tandy. DeskMate is their attempt to bring software integration to the Color Computer. To take advantage of the CoCo's graphics abilities, Desk-Mate leans heavily on Macintosh-like video icons to represent programs and data files. It also lets you enter commands via a joystick or mouse instead of the keyboard. You start an application or call a data file by moving the cursor to the directory location with the mouse and pushing the button twice.

Multiple Personalities

DeskMate consists of six major programs plus auxiliary routines. The major ones are Text Editor, Calendar, Index Cards (a simple data manager), Paint, Ledger (a spreadsheet), and Telecom. Subsidiary routines include utilities for setting printer and display parameters, along with a video representation of a four-function calculator.

Like the Tandy-sanctioned Dynacalc spreadsheet I described last month, DeskMate comes with a somewhat watered-down copy of the OS-9 operating system. Most of the commands you need are there, hidden behind the icons and menus. I would have liked more choices

for communicating with the printer (only 600 and 1,200 baud are available now), and the option of specifying faster stepping rates for disk drives.

It's understandable that Tandy would want to isolate newcomers from details of the operating system, but I'm not sure why they chose a unique file format. Other OS-9 applications can't read Desk-Mate data and vice versa, which might bother someone trying to graduate to more advanced software.

DeskMate components are also limited in their ability to exchange information. About all you can do is save portions of a Ledger spreadsheet as files for merging with Text Editor material, or cut and paste material from one Paint picture to another.

DeskMate routines don't really function as parts of an integrated product, despite first appearances; they're more like standalone programs on the same disk sharing a consistent user interface. A menu bar at the top of the screen lets you scroll the display, direct the opening and closing of files, call for Help menus, and summon desk accessories. Although details of program commands differ on the pull-down menus, the programs all have a similar feel.

Product Information

Tandy Corporation
One Tandy Center
Fort Worth, TX 76102
DeskMate requires 64K RAM and one disk drive; it sells for $99.95.

Writer's Block

Considering the amount of writing I do, it was only natural that I tried Text Editor first. The program is OK for correspondence and other short pieces (I used it to take about 1,500 words worth of notes for this column), but its deficiencies make it impossible to take Text Editor seriously as a full-fledged word processor.

As in other DeskMate routines, Text Editor uses software to generate an upper- and lowercase font. The 22 lines of 32 characters each that will fit on the screen aren't enough for serious writing. The font looks uneven, as well: lowercase letters don't all appear to be the same size.

You can use keyboard commands instead of a mouse to move the cursor. Arrow keys are much more convenient than the mouse for short moves. (The manual describes these commands in terms of the alternate and control keys, and acknowledges that "some Color Computers" may use @ and clear instead—further support for the rumor of a Super CoCo with an expanded keyboard.)

A combination of cursor movements and block menu commands lets you select chunks of text for moving or deletion. To move a block, you must copy it, insert it, and then delete it from its original spot. That's how Cognitec's Telewriter-64 works if you haven't added the Telepatch modification. The whole procedure is tiresome.

Text Editor has some amenities, including a seemingly unbeatable keyboard buffer that prevents speedy typists from losing material, and a fast-working Find Text option. The program fails in several

respects, though. It can't print headers, footers, or page numbers and there's no way to send control codes to a printer for boldface or italics. Also, a bug appears to exist in the print formatting routine: when I printed my notes, the first line of every page after the first one blithely ignored the left margin setting and turned up seven to 10 spaces too far to the left.

Text Editor files can be transmitted by the Telecom application. Unfortunately, the DeskMate manual doesn't say how large a file you can transmit.

Organizers

The Calendar program is one of DeskMate's best components. Its display consists of a calendar page for the current month, a large readout of the current day and date, and space for typing appointments and reminders. You can use the keyboard or a mouse to scroll through the calendar, while the Find Date and Find Text commands let you jump around.

The text area has no predefined slots, so you can organize notes however you like. Although you can see only eight 20-character lines at a time, up to 400 characters can be stored for a given day. The program will handle only 31 maximum-size entries, but you can beat the limit by using multiple date files.

You can print out data for a given day, week, or month, or everything in the calendar file. The listing comes out in a simple columnar format without the graphics.

Index Cards has the makings of a nice general-purpose filing system. It is based on "cards" of data fields that you define and label on the screen. You have to dig information out of the manual to use the program effectively, however.

The introductory part of the manual implies that only a single 18-character line can be devoted to each field. That's not true. You can reserve additional 30-character lines when laying out a card, but this important point is buried in the reference section at the back of the book. Material for each of the DeskMate routines is divided this way, presumably to keep rookies from getting bogged down in details. It's a poor split.

Each card will hold up to 22 lines. Keying in large sections of text is a test of endurance; the screen is *very* slow to catch up to input.

There are compensations. You can specify up to five fields for sorting, and the program will automatically put your file into the corresponding order and keep it there when you add or delete data cards. Other amenities include a versatile Find function with wildcard characters and both equality and inequality comparisons.

This flexibility allows you to select cards for printing in a report or to omit individual fields from a printout. The report format resembles the layout of an individual card, except that multi-line fields are displayed on longer lines without wordwrap. The program doesn't distinguish between alphabetical and numerical data, so you can't calculate totals.

Designs and Figures

The Paint program holds drawing tools for creating video displays. Almost all of the screen is available for drawing, since you access the tools from pull-down menus, not from a permanent menu on-screen.

Unfortunately, Paint doesn't play as smoothly as some of its competitors. Colorware's CoCo Max, in particular, gives you more control over images. Circles and rectangles created in Paint's high-resolution mode always seem to have gaps in their borders. Attempting to "paint" them often results in the pattern leaking out over the screen, and you usually have to touch up the borders by hand.

Paint lacks an image-scrolling feature, so it can't make drawings larger than the CoCo's screen. Its ability to mix text and graphics is also very limited, and only one type font is available.

There are three optional sizes for black-and-white screen printouts, and you can

allegedly get color prints from an ink jet printer. These routines didn't work with my Epson FX-80, perhaps because of oddball switch settings in the printer. I'm skeptical, though; the machine works with a lot of other software, including graphics drivers.

Ledger, a 99-row, 99-column spreadsheet, is another of DeskMate's better components. While lacking functions for scientific calculations and advanced financial work, it features automatic summation and row and column averaging.

Ledger makes it convenient to format a spreadsheet and copy a formula from one cell to another, adjusting address references in the process. You can even set up a formula that prompts you to enter a value from the keyboard for a recalculation. This possibly unique feature can be convenient for the "what if" games spreadsheet fans like to play.

You can insert portions of one spreadsheet into another, and save part of a sheet in text format for merging with a Text Editor document.

I haven't had the opportunity to use Telecom, DeskMate's telecommunications routine; it requires Tandy's Multi-Pak (or a similar interface) and an RS-232 Program Pak, neither of which I have. The program looks fairly straightforward, with one major screen for setting parameters, a second for establishing the communications link, and a third for viewing the contents of the RAM file containing the transmitted or received information. Incoming material is saved as a Text Editor file for further processing.

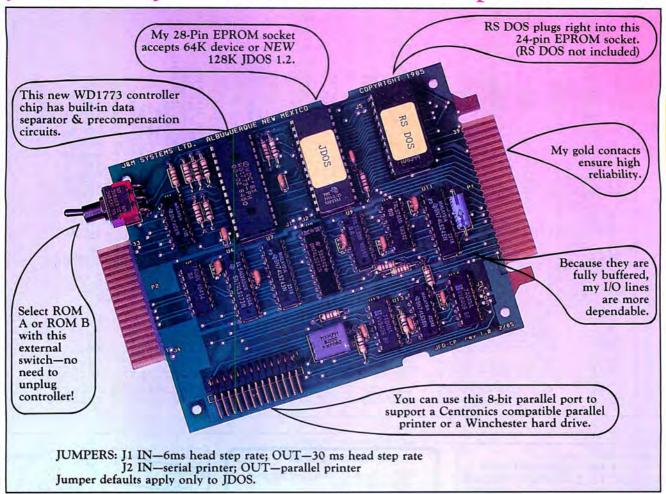## Irreconcilable Differences

The CoCo DeskMate's applications are presumably geared to computer novices. Note, for example, that the electronic mail function of other DeskMates was replaced with Paint. Some functions are too simplified for heavy-duty use, but as an "official" Radio Shack product, DeskMate is likely to attract a lot of attention.

I'm disappointed in Text Editor and Paint, though. While Paint is entertaining, it contributes little to the package; you can't splice its images into a Text Editor document or an index card, among other things.

I've already mentioned my objections to Text Editor. Any word processor that can't access features offered by modern printers is shortchanging its users. If it was too hard to build in that ability to transmit embedded commands, Tandy could have at least given us a program that can print page numbers. ■

Scott Norman is the manager of solid-state science at GTE Laboratories in Waltham, MA. Write to him at 8 Doris Road, Framingham, MA 01701.

J&M's New JFD-CP Disk Controller Speaks for Itself!

My 28-Pin EPROM socket accepts 64K device or *NEW* 128K JDOS 1.2.

RS DOS plugs right into this 24-pin EPROM socket. (RS DOS not included)

This new WD1773 controller chip has built-in data separator & precompensation circuits.

My gold contacts ensure high reliability.

Select ROM A or ROM B with this external switch—no need to unplug controller!

Because they are fully buffered, my I/O lines are more dependable.

You can use this 8-bit parallel port to support a Centronics compatible parallel printer or a Winchester hard drive.

JUMPERS: J1 IN—6ms head step rate; OUT—30 ms head step rate
J2 IN—serial printer; OUT—parallel printer
Jumper defaults apply only to JDOS.

New JFD-CP DISK CONTROLLER $149

Our new JFD-CP has redefined the state-of-the-art for Color Computer users. Gold contacts and fully buffered I/O lines ensure maximum reliability, and the JFD-CP is plug compatible with both the original COCO and the COCO-2.

New JDOS 1.2

JDOS implements all RS DOS commands, plus many more, including:
- auto line numbering
- error trapping
- baud rate selection
- OS/9* boot from floppy or hard drive
- Memory Minder†—the most advanced disk drive analysis system on the market
  (Dysan diagnostic diskette not included)

DRIVE 0 SYSTEM $289

Upgrade your Color Computer by adding our *new* JFD-CP disk controller, supercharged JDOS operating system, and a top quality drive with case and power supply: *All* for only $289! Comes complete with cable and manual.

Drive 0 System with one single side drive $289
Drive 0 System with one double side drive $359
Drive 0,1 System with two single side drives . . . $429
Drive 0,1 System with two double side drives . . $499

MEMORY MINDER†

Memory Minder is a disk drive test program now included in JDOS. Used with a Dysan digital diagnostic disk, Memory Minder allows you to check your drives for speed, alignment, sensitivity, hysteresis, and more! You can actually align the drives while viewing the graphics on the screen. No special equipment needed!

DIAGNOSTIC DISKETTES

508-200: Tests single side disk drives $26
508-400: Tests double side disk drives $33

Memory Minder is available on diskette for those who don't own a JFD-CP Controller with JDOS. Includes Dysan diagnostic diskette.

Single Side Memory Minder Package $79
Double Side Memory Minder Package $99

*OS/9 is a registered trademark of Microwave, Inc.
†Memory Minder is a registered trademark of J&M Systems, Inc.

To order, call (505) 292-4182, or send payment with order to:

J&M SYSTEMS, LTD.
15100-A CENTRAL SE
ALBUQUERQUE, NEW MEXICO 87123
505/292-4182

We accept MasterCard and Visa

Listing continued from p. 126

```
0086 70CA A780      B@      STA     ,X+         STORE IT ON THE SCREEN
0087 70CC 8C0600            CMPX    #$600       ARE WE DONE
0088 70CF 26F9              BNE     B@          IF NOT GO DO IT AGAIN
0089 70D1 16FF47            LBRA    BEGIN       GO START ALL OVER AGAIN
                    *THIS IS THE MAIN ROUTINE FOR BOTH UP AND DOWN*
                    *IT DOES MOST OF THE ERROR CHECKING AND CHECKS*
                    *TO SEE IF LINES WILL FIT AS WELL AS DOING THE*
                    *UNCRUNCHING AND PRINTING TO THE SCREEN        *
0090 70D4 10BE01E9  UNCRUN  LDY     BUF         GET THE LINE TO UNCRUN
0091 70D8 10BF01EB          STY     BUFF        KEEP IT SAFE FOR NOW
0092 70DC AEA4              LDX     ,Y          GET THE ADDR. OF NEXT LINE
0093 70DE 8C0000            CMPX    #00         IS THE ADDR. NONEXISTENT
0094 70E1 2613              BNE     B@          CARRY ON IF GOOD LINE NUMBER
0095 70E3 DC19              LDD     $19         IF NOT RESET TO BEGINNING
0096 70E5 FD01E9            STD     BUF         KEEP IT SAFE FOR NOW
0097 70E8 BE01F9            LDX     $1F9        THIS CHECKS TO MAKE SURE WE
0098 70EB BC01E9            CMPX    BUF         ARE NOT PRINTING THE SAME
0099 70EE 2702              BEQ     B@          LINE TWICE ON THE SCREEN
0100 70F0 20E2              BRA     UNCRUN      GO UNCRUNCH THE NEXT LINE
0101 70F2 3262      B@      LEAS    2,S         GET THE RIGHT ADDR. TO
0102 70F4 20A8              BRA     EXIT        DO OUR RTS
0103 70F6 BF01E9    A@      STX     BUF         MAKE THE NEXT CURRENT
                    *THIS ROUTINE CHECKS TO SEE IF THE NEW LINE WILL*
                    *FIT ON THE SCREEN                              *
                    *THIS PART GETS THE LINE LENGTH                 *
0104 70F9 BE01EB    CHECK   LDX     BUFF        GET THE ADDR. OF THE NEXT LINE
0105 70FC BDB7C5            JSR     $B7C5       UNCRUNCH TO GET THE END OF LINE
0106 70FF CC02DD            LDD     CC02DD      GET THE BEGINNING OF THE LINE
0107 7102 FD01ED            STD     COUNT       KEEP TRACK OF IT FOR NOW
0108 7105 1F20              TFR     Y,D         TFR TO PRESERVE THE Y REG.
0109 7107 C30025            ADDD    #$37        ADD ENOUGH TO COMPENSATE FOR THE
                    *                                     CARRIAGE RETURN
0110 710A B301ED            SUBD    COUNT       DO THE SUBTRACTING TO GET THE
0111 710D FD01ED            STD     COUNT       LENGTH OF THE LINE
                    *THIS PART FINDS OUT HOW MUCH SPACE THERE IS*
0112 7110 CC0600            LDD     #$600       GET THE END OF SCREEN
0113 7113 9388              SUBD    CURSOR      GET THE SPACE LEFT
0114 7115 10B301ED          CMPD    COUNT       WILL THE LINE FIT
0115 7119 2201              BHI     PRINT       IF SO GO PRINT IT
0116 711B 39                RTS                 IF NOT GO CHECK FOR A KEY
                    *THIS ROUTINE PRINTS TO THE                    *
                    *SCREEN. IT DOES THIS IN TWO PARTS             *
                    *THIS PART PRINTS THE LINE NUMBER              *
0117 711C 10BE01EB  PRINT   LDY     BUFF        GET THE LINE ADDR.
0118 7120 EC22              LDD     2,Y         GET THE LINE NUMBER
0119 7122 BDDBCC            JSR     $BDCC       AND PRINT IT
0120 7125 8660              LDA     #$60        PUT A SPACE AFTER IT
0121 7127 A79F0088          STA     [CURSOR]
0122 712B 0C89              INC     $89         AND BUMP THE CURSOR
                    *THIS PART PRINTS THE LINE                     *
0123 712D BE01EB            LDX     BUFF        GET THE CURRENT LINE BACK
0124 7130 8D14              BSR     KEPTRK      GO KEEP TRACK OF THE LINES
0125 7132 BDB7C5            JSR     $B7C5       DO THE ACTUAL UNCRUNCHING
0126 7135 CE02DD            LDU     #$2DD       GET ADDR. OF THE FIRST CHAR.
0127 7138 A6C0      A@      LDA     ,U+         STORE IT FOR PRINTING
0128 713A 2705              BEQ     ENDBUF      IS IT THE LAST ONE
0129 713C BDA282            JSR     $A282       IF NOT GO PRINT IT
0130 713F 20F7              BRA     A@          GO GET ANOTHER ONE
0131 7141 BDB95C    ENDBUF  JSR     $B95C       DO A CARRIAGE RETURN
0132 7144 208E              BRA     UNCRUN
                    *THIS ROUTINE KEEPS TRACK OF THE LINES ON*
                    *THE SCREEN AT ANY ONE TIME              *
0133 7146 10BE01F7  KEPTRK  LDY     BU          GET THE FIRST TABLE ADDR.
0134 714A 3122              LEAY    2,Y         BUMP IT FOR THE REAL ONE
0135 714C AFA4              STX     ,Y          STORE THE FIRST ADDR.
0136 714E 10BF01F7          STY     BU          STORE TABLE ADDR. FOR NEXT
0137 7152 39                RTS                 GO FINISH THE JOB
0138 7153                   END
NO ERRORS FOUND
```

End

Announcing. . . .

MEGADISK PLUS +
More Megabytes For Less Megabucks!

Next Page ——————————>
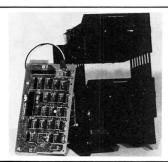
TOLL FREE ORDERING
1-800-343-8841

SOFTWARE SUPPORT, INC.
1 EDGELL ROAD, FRAMINGHAM, MA 01701
1-617-872-9090

Colorful Prints

Juki Office Machine Corp. offers the Juki 5510-Color dot-matrix printer. Unlike the Juki 5510, which uses a black ribbon and must be converted to support the color option, the 5510-Color has the color option already installed.

Maximum print speed for the Juki 5510-Color is 180 characters per second (cps) for draft quality and 30 cps for near-letter quality. It provides 96 ASCII characters, 96 italic characters, and 11 international character sets. The 5510-Color can produce seven colors from its four-color ribbon and features several printing modes, including double-emphasized.

The printer has a built-in tractor for a smooth friction feed and a 3K buffer memory (expandable to 15K). It also comes with a Centronics 8-bit parallel interface or an optional RS-232C serial interface and is available with either IBM or Epson printer software compatibility.

The Juki 5510-Color costs $650. For more information, contact Juki Office Machine Corp., Printer Division, 20437 South Western Ave., Torrance, CA 90501, 800-325-6134.
Circle 557 on Reader Service card.

Looking Ahead

MatheGraphics Software announces Multivariate Regression Analysis, a forecasting tool for businesses and research organizations. The program determines the most likely relationship among as many as 11 variables. Users can apply the relationship predictively to any set of data values they want to examine.

The program is available on tape or disk for the Color Computer and on disk for the Models III and 4. The cost for either version is $30.

For more information, contact Sheldon P. Gordon,

The Juki 5510-Color dot-matrix printer.

MatheGraphics Software, 61 Cedar Road, E. Northport, NY 11731, 516-368-3781.
Circle 560 on Reader Service card.

Better Bets

Someday soon, you might strike it rich with The Lotto Program from Soft-Byte Computer Programs. The software keeps track of winning numbers in state lotteries and helps you select winning combinations based on the laws of probability.

The program includes formats for all major state lotto games and a tutorial to help you get started. According to Byte Computer Programs, the program has produced several winning tickets.

The Lotto Program runs on the Models III and 4 and costs $19.95. For more details, contact Soft-Byte Computer Programs, P.O. Box 5701, Dayton, OH 45405, 513-278-8044.
Circle 563 on Reader Service card.

CP/M Booster

Out-Think is an outline processor for CP/M computers, including systems with the 8080, 8085, and Z80 microprocessors. The software allows you to prepare and edit text in outline form.

In addition to a full-screen outline editor with collapse and expand features, Out-Think offers a full-screen text editor for inserting and editing text. You can copy and move text within and between outlines and mark titles for copying, printing, or deleting.

The software includes more than 20 formatting parameters for such features as writing headers and footers, page numbers, section numbers, and tables of contents. Additional features include a built-in file manager, information retrieval by key word, date and time stamping, and password security on outline files.

Out-Think requires 48K RAM and an 80-column by 24-line screen. The program costs $49.95 including shipping. For more information, contact KAMASOFT Inc., 2525 SW 224th Ave., Aloha, OR 97007, 503-649-3765.
Circle 559 on Reader Service card.

Smart Talk

The Visionary 1200XT, a 1,200-/300-baud intelligent modem from Visionary Electronics, contains its own battery backed-up memory for sending, receiving, and storing messages (including TELEX and TWX) even if the host computer is turned off.

Incorporating a Bell 212A modem and an Intel 8085 microprocessor, the Hayes-compatible unit operates in originate and answer modes and transmits in either half or full duplex. It attaches to any personal computer through an RS-232C connection.

The Visionary 1200XT features call-progress detection, automatic log-on, and automatic data capture/retrieval. Messages can be directed to the unit's printer port for simultaneous printing during reception. With its built-in clock and calendar, you can control when messages are to be sent and when they can be received. Your computer can also use the clock to run programs at preset intervals, or to put a date and time stamp on documents it is printing out.

The Visionary 1200XT is available with 8K RAM for $495. You can purchase additional RAM in 8K increments for $20. A full 48K system retails for $595.

For more details, contact Visionary Electronics, 141 Parker Ave., San Francisco, CA 94118, 415-751-8811.
Circle 565 on Reader Service card.

Fortran Feats

Cleydale Engineering offers utilities to complement the Microsoft Fortran-80 compiler that runs under CP/M-80. Fortran-80 Utilities consists of an optimized scientific subroutine library, FORLIB.REL math additions, an escape sequence and control character generator for controlling peripheral devices, and three Fortran programming tools.

The package provides subroutines for linear and nonlinear regression analysis, statistics, matrix operations, equation roots, graphics, celestial mechanics, and other calculations. Programming tools include a utility for renumbering the numeric labels and line references of a Fortran-80 program; a high-performance, resynchronous ASCII file comparison program; and a file-scrolling utility.

The package costs $49. For more information, contact Cleydale Engineering, Route 1 Box 217-B, Blacksburg, VA 24060.
Circle 553 on Reader Service card.

Five computers can send data simultaneously to one printer with Data Manager.

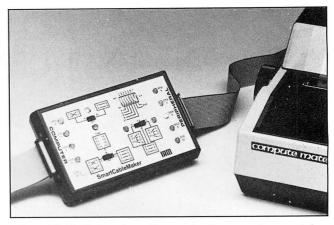*SmartCableMaker does the work of connecting peripheral devices.*

Shared Service

Belkin Components' Data Manager Spooler allows five computers to send data simultaneously to one printer, thus lowering printer costs for individual workstations. The spooler handles each incoming printer independently and processes data on a FIFO (first-in, first-out) basis.

Data Manager interfaces personal computers with the Hewlett-Packard LaserJet and other high-cost serial printers. No operator switching or setup is necessary. The spooler contains the Z80A microprocessor with six serial RS-232C ports (five input and one output). It also has a 64K buffer.

For pricing and other information, contact Steve Bellow, Belkin Components, 4718 W. Rosecrans Ave., Hawthorne, CA 90250, 213-644-3184. *Circle 551 on Reader Service card.*

Making Connections

The SmartCableMaker from IAM is an intelligent RS-232 interface that sets handshaking protocols between computers and peripherals automatically. It connects terminals, printers, modems, bar code readers, digitizers, or any other device that utilizes an RS-232 port.

To attach peripherals, all you do is plug in the SmartCableMaker and flip its three control switches. The device figures out what signals are on each line and sets the handshaking protocol. It also provides a graphics display of the interface configuration. For nonstandard applications, the SmartCableMaker

Share or alternate between peripherals with the A–B Switch.

Computer Accessories' expandable power protector and control center.

has a seven-position DIP switch that provides pull-up voltages to major control or handshake lines.

Contact Stan Burmich, IAM, P.O. Box 2545, Fair Oaks, CA 95628, 916-961-8082.
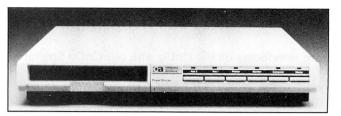
Circle 555 on Reader Service card.

Fast Switches

The A-B Switch from Inmac lets you share a peripheral with a coworker or switch between peripherals of your own. The device elim-inates the need for plugging and unplugging cables; to use a different peripheral, you simply turn a switch.

Two versions of the switch are available: the A–B Serial Switch ($95) with 25-pin female connectors and the A–B Parallel Switch ($139) with 36-pin Centronics female connectors.

Contact Inmac, 2465 Augustine Drive, Santa Clara, CA 95054, 800-547-5444.

Circle 556 on Reader Service card.

A Sense of Power

The Power Director (Model P15) from Computer Accessories Corp. gives you power protection and desktop control of system components. Its integrated circuitry provides high-speed transient voltage surge suppression, filtering of electromagnetic and radio frequency interference, and overcurrent protection.

The Power Director is an expandable unit that fits between your computer's monitor and central processing unit. Available options include a printer buffer, data transfer switches, and a re-mote security device.

The basic unit costs $169.95. For more information, contact Computer Accessories Corp., 6610 Nancy Ridge Drive, San Diego, CA 92121, 619-457-5500.

Circle 554 on Reader Service card.

Modem Media

Novice and experienced modem users can find answers to their questions in a new brochure from US Robotics, *24 Questions & Answers About 2400-BPS Modems*. The six-page, color brochure gives detailed answers to questions about the compati-

---

Circle 464 on Reader Service card.

Graphics Solutions
High-Resolution Software and Hardware

GBASIC 3.0 - Radio Shack Model 4/4D/4P/III hi-res board owners take note of an enhanced graphics Basic: GBASIC 3.0 It not only provides an equivalent for each of the BASICG commands but adds a number of important new ones while using less memory. Without having to exit Basic, the hi-res screen can be saved to disk, loaded from disk, or printed on any of 30 popular printers: Epson, Star Micronics, Radio Shack, Okidata, C. Itoh, NEC, etc. The software works with TRSDOS 1.3, 6.1.2, 6.2; DOSPLUS 3.4, 3.5, 4; LDOS; and NEWDOS80. The disk contains 40 graphics programs/files. Also included is a detailed manual with assembly language entry addresses. $49.95. (Specify Model 4 or III mode or add $10 for both.)

The following eleven programs run on a Model 4/4D/4P/III equipped with a Radio Shack graphics board and GBASIC 3.0 or a Micro-Labs Grafyx Solution board:

DRAW - A powerful full screen graphics drawing and editing program. $39.95.

BIZGRAPH - Create business graphs from hand-entered or VisiCalc data. $75.00.

xT.CAD - Professional drafting aid which outputs to a printer or plotter. $345.00.

SURFACE PLOT - Plot three-dimensional equations of the form $Z=F(x,y)$. $39.95.

3D-PLOT - View three-dimensional data from any perspective or angle. $39.95.

MATHPLOT - Plot equations of the form $Y=F(x)$ with auto scaling. $39.95.

CHESS - A very powerful program with 10 skill levels, 40 play options. $49.95.
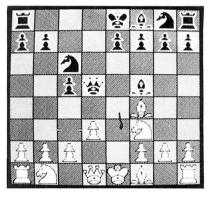
REVERSI - Play Othello with 10 skill levels, 20 execution options. $29.95.

3D Tic-Tac-Toe - Play the computer or a friend on a $4 \times 4 \times 4$ matrix. $19.95.

SLIDESHOW - Create a sequence of hi-resolution picture displays. $19.95.

Biorhythm/USA - Plot your biorhythm or learn the states and capitols. $19.95.

JOY-MOUSE - Allows a Radio Shack Color Computer joystick, mouse, or touch pad to be connected to any Model 4/4D/4P/III. Hardware provides X, Y position values from 0 to 255. $129.95.

GRAFYX SOLUTION - A plug-in, clip-on board enhances any Model 4/4D 4P/III to provide 640 × 240 dot graphics. (512 × 192 on a Model III) The board comes with a 56 page manual and a disk containing both model 3 and 4 mode versions of over 40 programs and files including GBASIC 3.0 which adds over 20 graphics commands to Basic. $199.95.

Please specify your exact system configuration when ordering or requesting information. Payment may be by check, Visa, Mastercard, or COD. Domestic shipping is free on pre-paid orders. Texas residents add 5⅛% sales tax.

MICRO-LABS, INC. 214-235-0915
902 Pinecrest, Richardson, Texas 75080

Our Breadboards and Texts Let You Acquire Data and Control Devices with Your Microcomputer

Learn interfacing on the CoCo with our CoCo Special Offer: Model CC-100 Expansion Connector Breadboard; Book 21893, *TRS-80 Color Computer Interfacing, With Experiments*, Model CC-150 Experiment Component Package. All three for $108.00 includes shipping.

Learn interfacing on the Tandy 1000, 2000, IBM-PC, and compatibles with our BG-Board Special: Buffered interface board, cable, and cablecard, with experiments. All for $325.00 or $305.00 for TRS-80 Model III or 4. Includes shipping.

Virginia residents, add 4% tax. VISA and Master Card accepted. 703-651-3153.

Write for our catalog of books, interface boards, and scientific software.

PUTTING
HANDS
AND
MINDS
TOGETHER

Group Technology, Ltd.
P.O. BOX 87 • CHECK, VIRGINIA 24072

Trademarks: Tandy 1000, Tandy 2000, TRS-80; Tandy Corp. IBM-PC; IBM Corp.

TRS 80 Computers

All Tandy Computer Products Available
Manufacturer's Warranty

Model 1000	**Model 3000**
Model 1200 HD	**Model 4D**
Model 2000	**Model 100/200/600**

Micro Mainframe 4 in 1 Mem. Exp. Boards
All RS Monitors

Unbeatable Prices

Visa — MasterCard
Cashier's Check — Money Order

Business Telephone Systems—Discount Prices—
Install your own. Completely modular.
Call for Prices

Computer Specialist available for assistance.
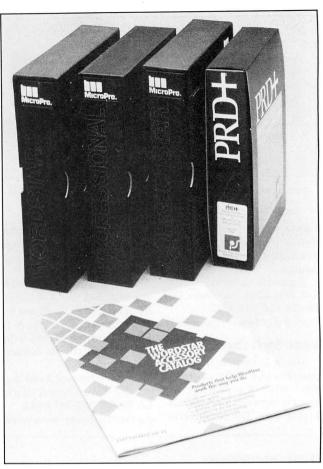
TALLEY COMMUNICATIONS CO.

P.O. Box 193 • 200 W. Walnut
Decatur, Texas 76234 • 817-627-2553
Call for other computer, telephone and accessory prices!

Fast Delivery   **References Available**

The WordStar Accessory Catalog *available from Broad-Reach.*

bility, performance, and technology of 2,400-baud modems. It also answers questions about telecommunications software.

For a free copy, contact US Robotics Inc., 8100 McCormick Blvd., Skokie, IL 60076, 800-342-5877.
Circle 564 on Reader Service card.

WordStar Wares

Shopping for WordStar add-ons and enhancements is made easier with *The WordStar Accessory Catalog* offered by BroadReach, a direct-mail marketing firm based in Seattle. The catalog lists WordStar add-on items from a variety of manufacturers.

Catalog offerings include WordStar utilities for laser printer support, automatic footnotes and file management, software and publications for WordStar customization, an on-line thesaurus, software for merging text and graphics, and keyboard productivity software.

To obtain copies of the catalog, contact BroadReach, 534 Industry Drive, Seattle, WA 98188, 800-641-1116.
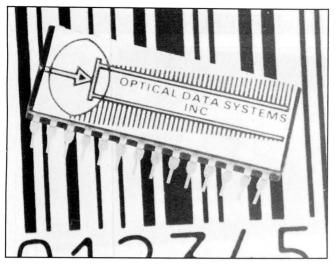*Circle 552 on Reader Service card.*

Bar Scanner

Optical Data Systems Inc. announces BAR+, a low-power CMOS ROM chip that reads and prints the formats of many industry-standard bar codes, including LOG-MARS (the military standard); HIBC (the health industry standard); AIAG (the automotive industry standard); and UPC (the grocery industry standard).

BAR+ can print bar codes to full specification on most Tandy dot-matrix printers in selectable quantities and various heights and sizes. Data input can be either manual or from text files.

The BAR+ ROM contains six bar-code scanning applications programs, plus a built-in file manager and calculator. A printed bar-coded

Optical Data Systems' BAR+ turns your Model 100 into a bar code reader.

keypad, special optical filter, and user manual come with the package, The single-unit price for BAR+ is $195. Quantity discounts are available for corporate users.

For further details, contact Sam Smith, product manager, Optical Data Systems Inc., P.O. Box 1987, Escondido, CA 92025, 619-745-6563.
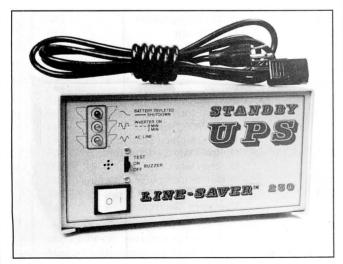*Circle 561 on Reader Service card.*

Juiced Up

Line-Saver (Model LS250) is a standby, uninterruptible power system from Kalglo Electronics Co. that is designed to keep you on-line even if the lights go off.

Line-Saver is available in 120/24-volts, 60/50 Hz with 250-watt capacity. It is furnished with an internal rechargeable battery, two voltage-surge protected ac outlets, audible and visual power failure warning system, test-mode indicator and switch, and replaceable external fuses.

The unit employs pulse width modulation (PWM) technology to regulate the output voltage more efficiently under various load conditions. The power system comes with a 6-foot, detachable, three-prong heavy-duty grounded cordset with a CEE-22 connector.

Line-Saver costs $549. For more information, contact Kalglo Electronics Co. Inc., Dept. CP, 6584 Ruch Road-E Allen Twp., Bethlehem, PA 18017, 215-837-0700.
Circle 558 on Reader Service card.

Line-Saver gives your computer a standby, uninterruptible power source.

DIFFERENT TRACK

Handy dispenser puts AYA Cleaning Wipes within reach.

Dust Buster

Screen Data Corp.'s new line of CRT and office equipment cleaners, AYA Wipes, won't leave you in the dust. The cleaning wipes are available with a dispenser that attaches to your monitor for easy access. The dispenser is theft-proof and can attach to other equipment, such as phones and keyboards.

Three types of wipes are available. AYA One-Step Blue Cleaning Wipes ($13.25) put an anti-static coating on the screen. AYA One-Step Yellow Wipes ($9.69) clean optically coated screens and antiglare filters. AYA Green Office Cleaning Wipes ($9.69) are designed to clean dirt and grime from keyboards, phones, and other office equipment. The dispenser is sold separately for $2.75.

For more information about the dispenser and cleaners, contact Screen Data Corp., 80 South Jefferson Road, Whippany, NJ 07981, 800-248-1212.
Circle 562 on Reader Service card.

New Products Index

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.

THE CLASSIFIED 80micro

Unlisted Numbers

*8*0 *Micro*'s Department of Judgment picked two winners of its Basic black book contest. (Too many good two-line telephone/address programs vied for third place to select three.) Both winners add and delete entries, and both bring up any entry by name, just like any good two-line data base. A combination of relative power and simple use separate the winners from the rest of the field.

Artfully Simple

Neither entry uses fancy tricks, just good planning and cramming—a trick in itself. Herbert Perkins (Thomaston, ME) has won an *80 Micro* T-shirt (they're in) for his Model III random-access Rolodex (Program Listing 1). Perkins' program provides a one-line menu that at all times proclaims its five functions: add, inquire, change, delete, and exit. Type the function number (1-5), press the enter key, then answer another prompt specific to that function.

The powerful search function pulls up any and all records containing the search string you enter. Matching records appear one at a time as you page with the enter key. The search string can be anywhere in a record. You can, for example, bring up all records containing a particular zip code or the name Smith. If you press the enter key at the search string prompt, all records display.

Each record contains four 24-byte fields. The length of the one-line menu shows maximum field length. Records aren't sorted, but they're numbered (a fifth field). You specify the displayed record number when you delete or change an entry.

To add a record, you enter four lines of information, pressing the enter key to go to the next numbered line. Commas are OK. Changing a record replaces an existing record completely; after specifying the record's number, you type in four lines of new information.

Because Perkins' program uses a non-standard record length (96 bytes), you must specify variable-length records when you enter Basic. When you see the "How Many Files?" prompt, type in 3V. Also, the first time you use the program, you must create an empty file as a starter data file. Follow these instructions:

1. Run the program (first time) and choose option 1 (add). You'll get the error message "Internal error in 1."

2. In command mode, type in LSET X$ = MKI$(1):PUT 1,1:RUN.

3. Add some records, and you're in business.

When typing in the program, remember to use the edit mode to finish the long program lines (for example, EDIT 1).

Brian P. Murphy (Paterson, NJ) earned his *80 Micro* bumper sticker with an address book (Program Listing 2) for Color Computer Disk Extended Color Basic. It also runs on the Model III. Murphy's entry adds and deletes entries to a sequential file; each entry may be a record of up to 255 characters. The file is kept in memory and sorted as you add or delete entries. A search function lets you call up all records beginning with the search key you enter. Added entries should start with the search key. On the Model III, you can get fancy with linefeeds (down arrow) and have multiple-line, formatted records (commas included).

A nice feature of Murphy's program is its simple user interface. When it's run, you see the > prompt. Entering a plus sign (+) brings up the data entry (>>) prompt so that you can enter new records. Typing in a minus sign (-) deletes the last record displayed. (If no record has been displayed with the search function, then the first record is deleted.) Typing in an "at" sign (@) ends the program and saves the file to disk if any changes have been made. If no changes are made, the disk file isn't updated (and you receive a harmless error message).

Any other string you enter is considered a search key, and all records starting with that key are displayed. Pressing the enter key displays the whole file.

Like Perkins' program, you must create a data file before you first use it. Type in the line OPEN"O",1,"DIR":CLOSE: RUN after you type in the program. The program is set up for a maximum of 99 entries (variable M in line 1) with an average length of 100 characters (CLEAR 10000). You can adjust M and the amount of string space in the Clear statement to fit your needs or memory size.

Statistically Significant

This month's challenge is highly improbable: squeezing a useful statistics package in (you guessed it) two significant lines of Basic. Sure, you can make yours do means and standard deviations, but what else can it do? Here are the rules:

1. Owners of all TRS-80 and Tandy systems with the exception of the Pocket Computers are eligible. We'll consider degree of difficulty when comparing solutions created on different machines.

2. The deadline will always be the 15th of the issue month. Thus, this month's deadline is June 15. We realize that this doesn't give everyone the same amount of time to come up with entries (we apologize to our overseas readers especially), but postponing the deadline any longer would add another month to our publishing the answers.

3. Speaking of the answers, they'll appear three issues from the issue in which the problem appears. Thus, this month's winners will make their appearance in the September 1986 issue.

4. Employees of CW Communications are not eligible.

5. Send your entry to *80 Micro*, Fine Lines, 80 Pine St., Peterborough, NH 03458. We will not, unfortunately, be able to return entries.

6. Specify your T-shirt size.

Program Listing 1. Herbert Perkins' Model III address filer.

```
1 IFI>5CLS:INPUT"1Add 2Inq 3Chg 4Del 5Xit";I:GET1,1:R=CVI(X$):C=1:GOTO1ELSEIFI=5CM
D"S"ELSEIFI=1FORX=1TO4:PRINTX;:LINEINPUTF$:LSETF$(X)=F$:NEXT:PUT1,R+1:LSETX$=MKI$(
R+C):PUT1,1ELSEIFI>2N=R:INPUT"#";R:IFI=4GET1,N:PUT1,R+1:LSETX$=MKI$(N-1):PUT1,1ELS
EC=N-R
2 IFI=2I=6:LINEINPUT"FIND: ";F$:FORX=2TOR:GET1,X:FORZ=1TO4:IFINSTR(F$(Z),F$)PRINT@
196,X-1:FORY=1TO4:PRINTF$(Y):NEXT:INPUTX:NEXTX:GOTO1ELSENEXT:NEXT:GOTO1ELSEIFI=0OP
EN"R",1,"A",96:FORX=0TO3:FIELD1,X*24ASX$,24ASF$(X+1):NEXT:I=6:GOTO1ELSEI=6+5*(I=3)
:GOTO1
```

End

Program Listing 2. Brian Murphy's Color Computer address book works on the Model III.

```
1 CLEAR10000:M=99:DIML$(M):K=1:F$="DIR":OPEN"I",1,F$:FORL=1TOM+1:IFNOTEOF(1)THENLI
NEINPUT#1,L$(L):N=L:NEXTELSENEXT:CLOSE:FORQ=0TO2:LINEINPUT">";A$:IFA$="-"THENN=N-1
:FORL=K TON:L$(L)=L$(L+1):NEXT:C=1ELSEIFA$="@"THENOPEN"O",C,F$:FORL=C TON*C:PRINT#
C,L$(L)
2 Q=0:IFA$="@"THENNEXT:CLEAR200:ENDELSEIFA$="+"ANDN<M THENLINEINPUT">>";L$:N=N+1:F
ORL=N TO1STEP-1:IFL$>L$(L-1)THENL$(L)=L$:L=0:NEXT:C=1:NEXTELSEL$(L)=L$(L-1):NEXT:N
EXTELSEFORL=1TON:IFA$=LEFT$(L$(L),LEN(A$))THENPRINTL$(L):K=L:NEXT:NEXTELSENEXT:NEX
T
```

End

We have a complete line of the finest software available for only

$29.⁹⁵ each

INTRODUCTORY OFFER
(or order any three packages and pay only $75.00)

Look at all of the features which our packages include!

You will discover that software doesn't have to be expensive to fill all of your needs. Our software runs on the IBM PC, XT, AT, PC Jr. and true compatibles. The accounting packages and WindowMaster require two floppy disk drives or a floppy disk and a hard disk drive.

THE General Ledger:

- Verifies all data input
- Totals entries to check balances
- Supports multiple departments or profit centers
- Proofs input data onto paper for auditing
- Lets you specify recurring journal entries
- On-line inquiry for account status
- Reports include: Income Statement, Balance Sheet, Chart of Accounts, General Ledger Report, Accounts Distribution Report, Trial Balance
- Prints Financial Statement in flexible, user-defined format

THE Accounts Payable:

- 5 transaction types: Release, Hold, Void, Prepaid and Credit Memos
- Access invoices for payment or inquiry
- Priority status or projected payment date available for each voucher
- Maintains numeric control over checks
- Complete payment information on voucher stub checks
- Computes discounts
- Writes automatic payables schedule
- Automatic vouchers for recurring items
- Allows partial payments
- Reports include: Aged Trial Balance, Two Pre-check Reports, Aged Cash Requirements, Vendor History Report, Check Register, Checks, Summary of Accounts Distribution Vendor List, Account List

THE Accounts Receivable:

- Balance forward or open item billing
- Automatic computation of late charges or interest rates
- Handle cash, C.O.D. or credit customers
- Specify any 3 invoice aging periods
- Print statements at any interval
- THE A/R posts cash receipts or invoices to customer accounts, types customer statements and reminder notices, and organizes and prints management information reports
- Automatically records recurring charges
- Reports include: Aged Trial Balance, On-Line Enquiry, Batch Proof, Accounts Distribution Report, Customer Statements, Customer Lists, Invoice & Cash Batch Proof, General Ledger Update
- Optional dunning letters

THE Order Entry:

- Processes invoices singly or in batches
- Up to 99 lines items per order
- Held (unshipped) items are marked on the Order Report
- THE O/E system automatically shows customer credit rating
- Each invoice depletes THE Inventory automatically
- Sends accounting transactions to THE Accounts Receivable
- Assign separate line item amounts to independent sales and receivables accounts
- Checks stock quantities in THE Inventory and notifies you when back orders are required
- Stock depletions will be automatically reported in THE Inventory
- Detailed invoice information
- The Order Report provides urgent information on every order

THE Inventory:

- Perfectly suited to retail environment
- Simple transactions for: addition, deletion, alteration and examination
- Complete current statistics on each inventory item
- Activity statistics for current or to-date period
- Instant on-line inquiries
- Optional hard copy record of all inventory transactions for auditing
- 13 digit, user-definable alphanumeric part number
- 30 character description field
- Reports include: Activity Report, Valuation Report, Auditability Proof, Reorder Report, Item List, and Item File Print

THE Payroll:

- Handles any combination of weekly, biweekly, semimonthly & monthly pay schedules
- Define up to 20 earning or deduction categories for all employees
- Assign any 8 categories for each employee
- Accountants can perform after-the-fact payroll processing
- Prints information for government reports
- Salary expenses can be distributed to different departments, branches or profit centers
- System includes safeguards and verification procedures
- Reports Include: Company History, Pay Transaction Proof, Calculations Proof, Employee History, Employee Master List, Check Register, Payroll Journal, Vacation report, 940 & 941 Reports, W2 Forms, & Prints Checks
- Of course, THE Payroll calculates payroll including Federal & State taxes, prints checks, allows hand written checks, lets user maintain tax tables and keeps an extensive employee payroll history

THE Name & Address System:

- Create name & address files
- Print mailing labels
- Sort files by up to 4 items
- Search or sort by match or range
- Print a report of a NAD file
- Merge files
- 88 character open reference field
- Sort by any character in reference field
- 10 number zip code

THE Spreadsheet:

- On-line tutorial
- Cut & paste multiple spreadsheets
- Format: Global, Column, Row, Entry
- Arrays: Copy, Replicate, Move, Insert, Delete, Blank
- Labels: centered, right or left justified
- Variable cell width
- 2 Windows to view separate areas of worksheet
- 20 special math functions
- Manual or automatic recalculation
- Complete glossary of spreadsheet terms
- Maximum spreadsheet size: 255 Rows x 64 Columns
- Variable print formatting options
- Simple editing functions
- Status on contents of any cell
- Set hold on any row or column to prevent srolling
- On-line help

THE Wordprocessor:

- On-line help
- Powerful editing: Erase, Copy or Cut & Paste characters, words, sentences, lines, paragraphs, pages, or user-defined blocks of text
- Automatic word wrap-around (may be turned off)
- User-defined page breaks, soft hyphen, tab, indent and margin settings
- Proportional & justified printing
- Global Search & Replace, Automatic & Examine
- Optional command character viewing
- Superscript and subscript
- Phrase files for boilerplate
- Print bold, centered, underline
- Edit ASCII files
- Right or left justification
- Hyphenation

THE Mailmerge:

- Integrates THE Wordprocessor & THE Name & Address System
- Mass mailings
- Specialized mailings to select groups in your Name & Address System.
- Can read A/R customer file, A/P vendor file and payroll employee file so you may send customized Wordprocessor letters

WindowMaster:

Our windowing system lets you integrate your application programs. You can run up to 7 applications simultaneously. Run THE Payroll and print checks while you are using THE Wordprocessor. Or enter orders with THE Order Entry while THE Spreadsheet performs a lengthy calculation and THE Mailmerge prints letters for a mass mailing. The simple interwindow transfer feature allows you to pass data from one application to another. The powerful keyboard enhancer lets you program keys on your keyboard to perform special functions which would otherwise require many keystrokes. You can also redefine keys. WindowMaster integrates your IRE applications and enhances your computer.

CALL TODAY TOLL FREE:
1-800-227-2400
Ask For Extension 931
In California:
1-800-772-2666, Ext. 931

(Lines Open 24 Hours, 7 Days a Week)
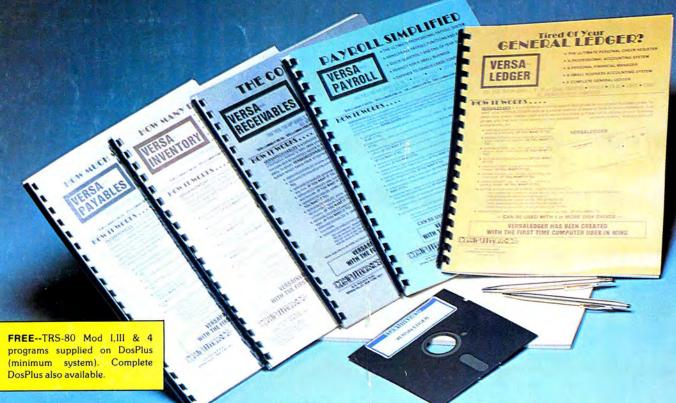FOR MORE INFORMATION CALL:
(415) 268-1435

ACCOUNTING PACKAGES: FULLY INTEGRATED OR STAND-ALONE WITH COMPLETE AUDIT TRAILS

Indian Ridge Enterprises, Incorporated

508 Second St., Oakland, CA 94607 Dept. 80
$29.95 each or $75.00 for any 3 packages. Please Include Shipping & Handling charges listed below. Credit Card Orders by Phone OR Send Check OR Money Order with Coupon. Sorry, No COD's or Purchase Orders.

☐ THE General Ledger ☐ THE Order Entry ☐ THE NAD System ☐ THE Mailmerge
☐ THE Accounts Payable ☐ THE Inventory ☐ THE Spreadsheet ☐ WindowMaster
☐ THE Accounts Receivable ☐ THE Payroll ☐ THE Wordprocessor

☐ Check ☐ Money Order ☐ VISA ☐ MasterCard

Account No. _____ Expiration Date _____

Name _____

Address _____

City _____ State _____ Zip _____

Phone _____ Signature _____

1 or 2 Packages — Add $7.50 Shipping & Handling
3 to 5 Packages — Add $10.00 Shipping & Handling
5 to 10 Packages — Add $15.00 Shipping & Handling
California Residents add 6½% Sales Tax. Outside US add $15.00 and payment must be made by bank draft payable in US by US Dollars.