80 Microcomputing 5/81 #17

May 1981
US $2.50/DM 9

80 microcomputing™

the magazine for TRS-80 users*

High Finance At Home

INSIDE:
Dow Jones/Tandy offer a powerful new investment tool: Instant financial news

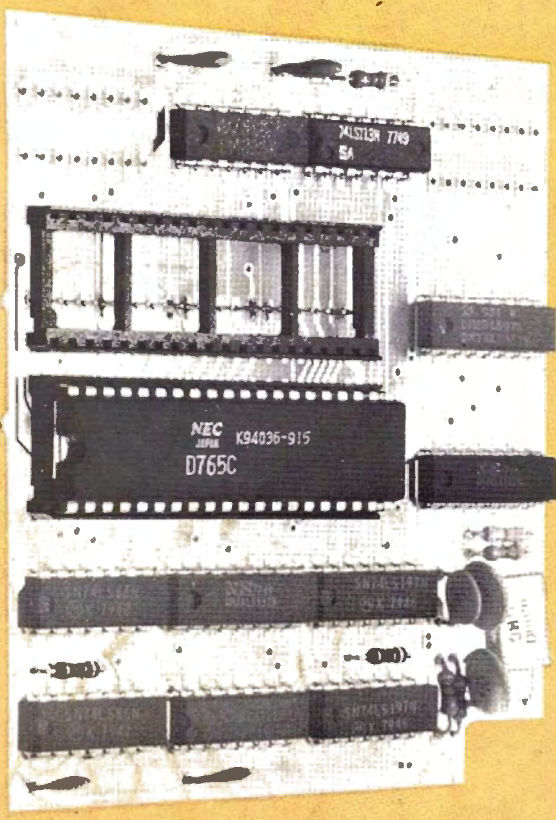
Let your 80 organize your finances!

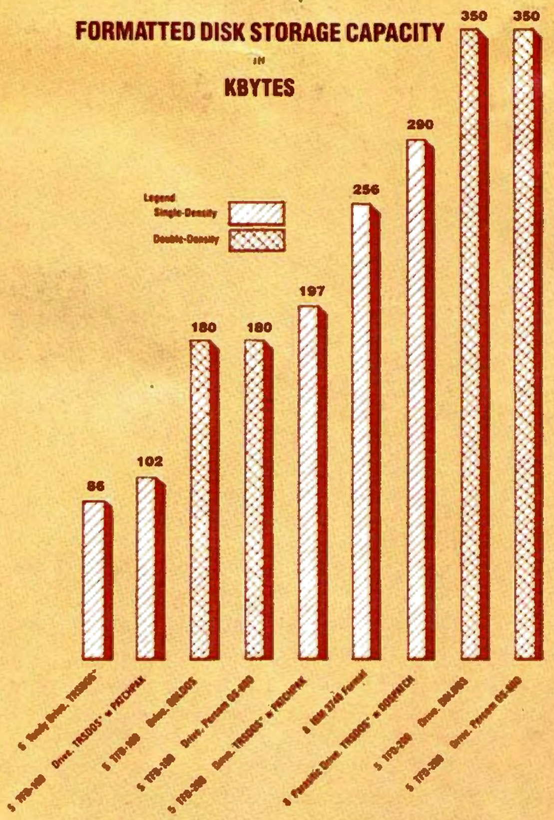
How much is a new home really going to cost?

*trademark of Tandy Corp.

TRS-80* Model I Computer Owners . . .

Store Up to 350 Kbytes on a 5″ Disk





The DOUBLER™. It packs almost twice the data on a disk track as your single-density system. Depending on the type of drive, you can store up to **four** times more data on one side of a minidiskette than you can store using a standard Model I mini-disk drive.

- The DOUBLER™ reads, writes and formats **either** single- or double-density minidiskettes.
- Proprietary design allows you to continue to run TRSDOS*, NEWDOS‡, Percom OS-80™ or other single-density software **without making any changes** to software or hardware. Switch to double-density operation at any convenient time.
- Includes DBLDOS™, a **TRSDOS* compatible** double-density disk operating system.
- CONVERT utility, on DBLDOS™ minidiskette, converts files and programs from single- to double-density or double- to single-density.
- The DOUBLER™ circuit card **includes high performance data separator, write precompensation** circuits for reliable disk read operations — even with 80-track drives.
- **Plug-in Installation** — The DOUBLER simply plugs into the disk controller socket of your Expansion Interface, requiring no strapping or trace cutting. Expansion Interface disk controller may be completely restored to original configuration by simply removing the DOUBLER™ and re-installing the original disk controller chip.
- Works with standard 35-, 40-, 77- and 80-track mini-disk drives rated for double-density operation.
- Introductory price, including DBLDOS™ and format conversion utility on minidiskette, **only** $219.95.

Mini-Disk Systems

More storage capacity, higher reliability — from Percom, the industry leader. One-, two- and three-drive configurations in either 40- or 77-track format. Fully burned-in. From only $399.

Double-Density Software

(On diskette — with instruction manual.)

OS-80D™ Double-Density Disk Operating System — This double-density upgrade version of Percom's acclaimed OS-80™ resides *entirely* in RAM — requiring only 7.5-Kbytes! A BASIC programmer's "dream operating system," even utilities are in BASIC.

DOUBLEZAP-II/80 This program modifies Apparat's NEWDOS/80‡ to run either double — or single-density programs — even to run a mix of the two formats on one system!

DOUBLEZAP-II/V This program modifies Virtual Technology's VTOS 4.0†† to provide the same capability as DOUBLEZAP-II/80 provides for NEWDOS/80.

Call toll-free, 1-800-527-1592, for the address of your nearest authorized Percom dealer, or to order directly from Percom.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

™ trademark of Percom Data Company, Inc.
* trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.
‡ trademark of Apparat Company, Inc. †† trademark of Virtual Technology, Inc.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.
PRICES DO NOT INCLUDE SHIPPING AND HANDLING CHARGES.

Percom Mini-Disk Drive Systems for TRS-80* Computers...

Now! Add-On *and* Add-In Mini-Disk Storage for your Model III.

The industry leader in microcomputer peripherals, Percom not only gives you better design, better quality and first-rate service, but you pay less to *boot.*

New for the TRS-80* Model III

Patterned after our fast-selling TFD Model I drives. And subjected to the same reliability controls. These new TFD mini-disk systems for the Model III provide more features than Tandy drives, yet cost far less.

- **Flippy Capability:** Both internal (add-in) and external (add-on) drives permit recording on either side of a diskette.
- **Greater Storage Capacity:** Available with either 40- or 80-track drive mechanisms, Percom TFD mini-disk systems store more. A 40-track drive stores up to 180 Kbytes — formatted — on one side of a 5-inch diskette. An 80-track drive stores a whopping 364 Kbytes.
- **1.5 Mbyte On-line:** The Percom drive controller (included with the initial drive) handles up to four drives. With four 80-track mini-disk drives you can access over 1.5 million bytes of on-line file data.

  Moreover, the initial drive may be **either** an internal add-in drive or an external add-on drive. And whichever configuration you get, the initial drive kit comes complete with our advanced 4-drive controller, interconnecting cables, power supplies, installation hardware, a DOS and of course the drive mechanism itself.
- **First Drive Includes DOS:** OS-80™, Percom's fast extendable BASIC-language disk operating system, is included on diskette when you purchase an initial drive kit. Originally called MicroDOS, OS-80 was favorably reviewed in the June 1980 issue of Creative Computing magazine.
- **Works with Model III TRSDOS:** Besides being fully hardware compatible, Percom's Model III 40-track drive systems may be operated with Tandy's Model III TRSDOS — without any modifications whatsoever. And, TRSDOS may be easily upgraded with simple software patches for operating 80-track drives.

Percom TFD add-on drives start at only $399. Model III Drive kits start at only $749.95.

Quality Percom products are available at authorized dealers. Call *toll free 1-800-527-1592* for the address of your nearest dealer or to order direct from Percom.

Still #1 for Model I

As if greater storage capacities, exceptional quality control measures and lower prices aren't reasons enough to make Percom your first choice for Model I add-on drives, **all** Percom Model I drives are also rated for double-density operation.

Add our innovative DOUBLER™ adapter to your Model I Expansion Interface, and with Percom drive systems you can enjoy the same double-density storage capability as Model III owners.

The DOUBLER includes a TRSDOS*-like double-density disk operating system called DBLDOS™

We also offer a double-density Model I version of OS-80 as well as DOUBLEZAP programs for modifying NEWDOS/80 and VTOS 4.0† for DOUBLER compatibility.

Of course you don't have to upgrade your computer for double-density operation to use Percom mini-disk drive systems. In single-density operation, our TRS-80* Model I compatible 40-track drives store 102 Kbytes of formatted data on one side of a diskette, and our 80-track drives store 205 Kbytes. By comparison, Tandy's standard drive for the Model I stores just 86 Kbytes.

And like our Model III drives, Model I add-on drives are optionally available with "flippy" storage capability.

System Requirements:

Model III: 16-Kbyte system (min) and Model III BASIC. The second internal drive may be installed after the first internal drive kit is installed, and external drives #2, #3 and #4 may be added if either an internal or external first-drive kit has been installed. External drives #3 and #4 require an optional interconnecting cable.

Model I: 16-Kbyte system (min), Level II BASIC, Expansion Interface, disk operating system and an interconnecting cable. For double-density storage, a Percom DOUBLER must be installed in the Expansion Interface and DBLDOS (comes with the DOUBLER) or other double-density DOS must be used. For single-density operation, a Percom SEPARATOR™ adapter, installed in the Expansion Interface, will virtually eliminate "CRC ERROR — TRACK LOCKED OUT" read errors. Prices and specifications subject to change without notice

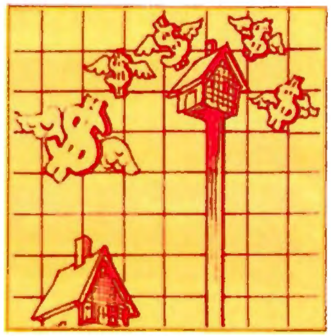
PERCOM

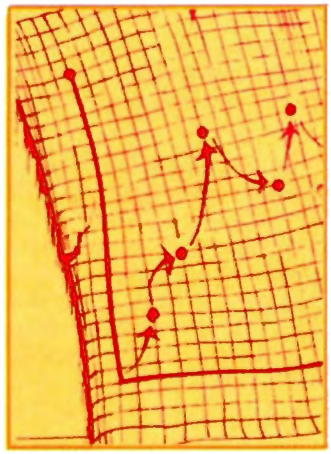
PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND TEXAS 75042
(214) 272-3421

406

*Trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.
™DOUBLER, DBLDOS, OS-80 and SEPARATOR are trademarks of Percom Data Company, Inc.
†Trademark of Virtual Technology Corporation.

80 Contents





DEPARTMENTS

80 Microcomputing (ISSN -0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-3873. Subscription rates in U.S. are $18 for one year and $45 for three years. In Canada, $20—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. Foreign subscriptions (surface mail), $28—one year only, U.S. funds. Foreign subscriptions (air mail), $60—one year only, U.S. funds. In Europe contact Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form -3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.

PUBLISHER/EDITOR
Wayne Green

MANAGING EDITOR
Michael Comendul

TECHNICAL CONSULTANT
Jake Commander

PRODUCTION EDITOR
Bob Dukette

NEWS EDITOR
Bert Latamore

REVIEW EDITOR
Pamela Petrakos

ASST. TECHNICAL EDITOR
Chris Brown

ASSISTANT EDITORS
Lise Markus
Debra Marshall

TECHNICAL CONTRIBUTING EDITOR
Dennis Kitsz

EDITORIAL ASSISTANT
Janet Fiderio

EDITORIAL ADMINISTRATION
Pat Graham
Nancy Noyd

DESIGN ASSOCIATE
Diana Shonk

PRODUCTION MANAGER
Nancy Salmon

ASST. PRODUCTION MANAGER
Michael Murphy

AD GRAPHICS MANAGER
Robert Drew

AD COORDINATOR
Sue Symonds

ADVERTISING PRODUCTION
Steve Baldwin, Bruce Hedin, Maryann Metivier

PRODUCTION DEPT.
Joan Ahern, Fiona Davies, Linda Drew, Frances Benton, Kenneth Jackson, Ross Kenyon, Theresa Ostebo, Jane Preston, Thomas Villeneuve

PHOTOGRAPHY
William Heydolph, Terrie Anderson, Bill Suttenfield

TYPESETTING
Barbara Latti, Sara Bedell, Michele Desrocher, Luann Keddy, Mary Kinzel, Linda Locke, Karen Stewart

Cover by Tom Grubis

80 REMARKS

by Wayne Green

"Before we can accept the computer we may have to adjust to a fact of life: It is not our responsibility to force-feed education to children."

When I went to high school, some 40 years ago, it was enjoyable. I went to one of the best schools in Brooklyn, New York, Erasmus High School. It was beautiful. The "old school" stood in the center of a block long (and the block was very long) quadrangle of the new building, which rose three stories and looked like an enormous castle. During my tour at Erasmus we celebrated its 150th anniversary.

One of the features I enjoyed the most about the school was the wide choice of after-school clubs to which I could belong, —over 120. There weren't enough afternoons in a week to participate in all of them.

The most prestigious of the clubs was the Choral Club, which practiced daily during the second period. Erasmus gave every student a voice test, and the top one hundred voices were able to join this club. The school had about 10,000 students, so it was quite an honor to be selected. I'd been singing several years in a church choir as a soprano, so I had no difficulty in getting into this group with my newly developed baritone voice. We sang daily and gave concerts quite regularly.

One of the other clubs that I enjoyed was the ham club (W2ANU), one of the oldest in the country. There I was able to learn more about amateur radio and operate the club ham station. Then, there was the Savoyards, a club dedicated to singing Gilbert and Sullivan operettas. I still know several of them almost word for word.

On Tuesday afternoons I went to the camera club. Do you know how long ago that was? Well, I remember going over the very first issue of *Life* magazine. Imagine, a magazine based on photographs... what excitement for a camera group!

Not long ago I saw a TV program about Erasmus on one of the news specials. It is now classed as the most dangerous school in New York! It takes a special school police force to keep crime down to "acceptable" levels. It is an armed camp.

Other recent television stories confirm the plight of teachers and of those students wishing to learn. Classes are a shambles. It is getting almost impossible to teach because an ever increasing number of students have lost interest and are distracted by those youngsters who are there against their will. It is demoralizing for teachers, who are deserting the field by the thousands.

Undisciplined Students

Apparently no one has been able as yet to figure out any answer to this situation. I know what it's like to have even a couple of undisciplined students in a class. I ran into that in grammar school in Brooklyn. I don't think any of my classmates will ever forget Myron Nussbaum or Norman Golding (those are the true names, so sue me for libel), who spent each day trying to make life miserable for our teacher. They got up when they felt like it, yelled at her, then tossed erasers at her, as she chased them around the classroom. The school principal couldn't do anything either. Apparently their families didn't care. So we lived with it, were amused by it... and hated it.

Perhaps the computer comes to education just in time. Before we can accept the computer we may have to adjust to a fact of life: It is not our responsibility to force-feed education to children. Isn't it enough that we make it available?

We're a prisoner of our past in much of this. Schools developed for good reasons, but they evolved into giant baby-sitting systems for most families. Summer vacations may have been a blessing for the kids, but many mothers sighed when they came.

I believe that eventually we will provide a better quality of education by means of computerized teaching than is currently available in most schools. Truly interesting courses can be taught by some of the best teachers in the world. The computer, as I've written before, can control the speed of the teaching process, gearing it to the interest and intelligence of the student.

Once we have developed better learning systems, not dependent upon live teachers for other than minor support and advice, it may be possible to move these systems into the home. This may be a healthy change for society and result, in the long run, in more highly motivated children.

But what has brought such a radical change to Erasmus High?

Many factors have transformed schools over the last forty years. We can certainly point to the almost standard family with two working parents, a necessity just to meet payments for homes and food, but which has removed the influence of the mother from our children after school.

Worse, the ubiquitous television and its continuously violent offerings not only claims our children's attention, but almost totally occupies the parents in the evening too. This removes the chances for any serious communication between parents and children.

Add to this the doctrine of Dr. Spock and the "let the children do what they want" attitude, and we may have the roots of the lack of respect and the vandalism which is now engulfing us.

Police Protection

When I see movies of teachers trying to cope with children who are uninterested in learning, I wonder that these teachers aren't lobbying harder for computers to help them with their thankless jobs. With computers for each student there would be less opportunity for students to create havoc, throw erasers (there will be no more erasers), or molest the teachers. At Erasmus teachers have police protection walking around the halls.

One of the problems of compulsory education is not only that it forces children to be in school but that it also acts upon the teachers, who address captive audiences, and thus have no responsibility to make the work interesting.

In my four damned years of college I had only one teacher who made an effort to make his classes interesting, and he succeeded. Every student enjoyed his accounting class...and *learned*, because the course was fun and exciting. Imagine a subject as dry and dusty as accounting being fun! The rest of the classes stunk. I

Continues to page 10

INSIDE 80

by Ed Juge, director of computer merchandising, Tandy Radio Shack

". . . your ideas and opinions. . . are very important to us. We appreciate hearing your thoughts."

Sincere thanks to all of our friends who have written the last couple of months. We appreciate your input. I've answered as many of your letters as time allowed, passed some on to Bill Walters, and some to the appropriate product managers for reply. Those suggestions you made *did* get into the appropriate hands.

Occasionally I'm asked why I talk about products before they're actually available, or say something which readers know, when they read it, is not true. This week, by phone, came the best suggestion of all: Why don't I explain both of these concerns every couple of months for those who haven't followed the column regularly.

It's relatively simple: In an early column we said that we would be as up front with you as possible, even at the risk of being proven wrong, or having a discussed product delayed, etc. I said we'd try it, but that readers would have to allow a fudge factor for Murphy's law, which would surely take its toll. All in all, it's worked well. However, there are a few folks that have really gotten bent out of shape at unintentional inaccuracies.

The second factor is that the column for each issue is written some three months prior to the cover date. Unless the facts of the subject look pretty stable, I normally do not mention them, but even those appearing certain can fall through the cracks. So, now you know how the inaccuracies happen.

Next, I owe all of you an apology. With the appearance of the March issue of *80*, a call came in asking about the "color computer accessory" I promised to talk about in March. Well, I blew it! I'll try to be more careful in the future.

TRS-80 Microcomputer News

Believe it or not, we still get letters almost daily from disgruntled customers who aren't getting the *TRS-80 Microcomputer News*. Or from those who are disgusted about having to pay for it after the first twelve free issues (some of you didn't even get them). I've probably written several hundred personal letters on the subject. I've mentioned it here twice, and it's been on CompuServe, with the hope of getting the word around by hearsay to those not getting the publication.

Your name is placed on the mailing list on our big computer system, dated from the card which came with your computer. If you thought sending in this card as registering your warranty, forget it! Radio Shack warranties products based on the date on your original sales ticket and nothing else. The only reference made to that card is for the purpose of mailing newsletters and special computer information to you. If you thought you were extending your equipment warranty by delaying mailing it, you're in for a big surprise. If you didn't want to invest in an 18 cent stamp, you really didn't want the *News*, and I can't feel sorry for you.

If you didn't get a card, we can register you from a written request and a photostat of your original sales ticket, showing purchase of a new computer (CPU only) from any authorized Radio Shack outlet. But if you try to get on the mailing list a second time, don't bother. . .our computer remembers you, and knows whether or not you've had your 12 labels run already. It also knows whether two separate purchases were involved, entitling you to 24 months, etc.

Unauthorized Dealers and The Black Hole

Remember, there are some unauthorized folks out there selling new TRS-80's. If you bought from them, their sales ticket means nothing to us. If you didn't get a free newsletter card from them, talk to them. . .not us. Maybe they'll buy you a $12.00 subscription.

Let's assume you get properly registered into our computer system as a new owner. We print labels for 12 copies of *Microcomputer News*, following your registration. They're affixed to newsletters, and delivered to the post office. The mailing service returns any labels "eaten up" by the automatic labeling equipment, and we duplicate them. This is to ensure that everybody on the list has a newsletter delivered to the post office. That's all we can do. If yours gets lost in a USPS black hole, the only solution may be to subscribe. Paid subscribers are mailed first class, rather than the bulk mail used for the free copies. First class seems to somehow effectively circumvent the black hole.

Sorry, we can't send the newsletter free forever, nor can we keep sending it until you've managed to receive twelve copies. As a matter of fact, our two largest competitors in the small computer market include one who publishes no newsletter at all, and one who charges you from issue number one! Consequently, I don't feel our policy is all that bad. You'll also find that our newsletter is no longer just an advertising sheet as it originally was. We make new product announcements, along with in-depth product descriptions, but there's no outright advertising at all.

Sorry Again

Sorry again, but we do not have a significant supply of back issues! In fact, many of the back issues are no longer available, so save yourself and us some time and don't bother asking for them.

Level II Teaching Book Available

You've been asking for this one for only about two and a half years now. I hope it's better late than never! Written in-house, it's entitled *Getting Started With TRS-80 BASIC*, and is priced at $6.95.

It applies equally to both Level II and Model III BASIC and is written in the same style as our original Level I training manual.

There are three parts to the book. Part one is for the absolute beginner, the first-time user, who has only just found the on/off switch. Part two takes full-swing into the language and part three goes into some of the more advanced concepts and commands including PEEK and POKE, graphics, error-handling, formatted printing, etc. There are exercises, sample programs, tables and charts, and of course a complete index (342 pages in all). If you're already a proficient programmer, you'll want this book as a reference for your Model I or III; even Model II owners will find it most useful. It is also a valuable reference guide for other family members

META TECHNOLOGIES

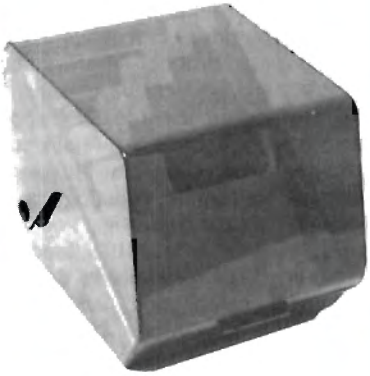
26111 Brush Avenue, Euclid Ohio 44132

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

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

FILE BOX

DISKETTE STORAGE SYSTEM



$24.95 for 5¼" disks

$29.95 . . . for 8" disks

MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" disks $10.95
HUB RING KIT for 8" disks $12.95
REFILLS (50 Hub Rings) . . . . $ 5.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives $24.95

PLASTIC LIBRARY CASES

(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.50
8-inch diskette case $3.95

Single Sided, Soft-Sectored 5¼-inch, (for TRS-80™) Mini-floppy

DISKETTES

$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™

DISKETTES

The Beautiful Floppy with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 (each box) $21.50

PLAIN JANE™ *Gold*

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

PLAIN JANE™ *Gold* $25.95

VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually, 100% error-free certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Reinforcing HUB RINGS help prevent data loss and media damage, reducing errors.

Buy the best . . . buy DATALIFE™

VERBATIM DATALIFE™ DISKETTES

5¼-inch (box of 10)
MD525-01 $26.95
10 boxes of 10 (each box) $25.95

8-inch FLOPPIES
Double-Density, FD34-8000 . $43.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 ★SPECIAL★ $129.95
NEWDOS+ to NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER . . . . $ 7.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

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

SYSTEM DIAGNOSTICS $19.95

An improved version of the SYSTEM DIAGNOSTICS above. Designed for single or double density, 35-, 40-, 77-, or 80-track disk drives. Includes new and modified tests. Features THE FLOPPY DOCTOR, Version 3.0.

SYSTEM DIAGNOSTICS-V3 . . $24.95

MORE PRODUCTS →

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC. ©1981 by Metatechnologies Corporation, Inc.

20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT May 1, 1981 THRU May 31, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8105

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

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

who would like to learn how to program. (Model III owners will receive a copy at no charge, since this is the equivalent of the *Level II Reference Manual* supplied with Model I Level II machines.)

TRS-80 Software Sourcebook, Volume 3

By the time you read this column, the third printing of our sourcebook should be ready. It looks like it will top 2,000 listings with this issue! Amazingly, we have received almost no complaints from folks who have ordered programs from the book. As most of you know, we suggest that you contact the advertiser and draw your own conclusions as to the viability of their product, since Radio Shack can't possibly verify the useability, accuracy, or even the existence of the programs listed.

For those of you unfamiliar with the book, it lists applications programs for various TRS-80's, available from a variety of vendors. There's a brief description of the program, minimum required equipment, formats in which it's available (cassette, diskette, source listing, etc.), and cost. It also includes the vendor's name and address for ordering.

Pocket Computer Sale

Although I know you may receive this magazine a few days before the end of April, I'm going to tell you anyway, because it's exciting news! During the month of May, our fantastic TRS-80 Pocket Computer will be on sale for $199! That's a savings of $50 off the regular price. You can't buy one for the sale price until May 1, so don't bother trying. But come May 1, if you've been putting off getting your Pocket Computer, there won't ever be a better time! There'll be a lot of pre-recorded software available for a number of applications, and of course you can get the optional cassette interface (not on sale) to load our packages, or your own.

I can't tell you how much nicer it is to have a BASIC programmable computer in your pocket or briefcase, than a calculator. Don't miss this chance, because after the end of May the price may never be this low again. I'm telling you about the sale here, since *80 Microcomputing* doesn't carry Radio Shack advertising. Wayne feels it might somehow compromise editorial credibility. Although we'd like to spend a little money with him, we respect his feelings.

Daisy Wheel II Deliveries

Here's a bit of forecasting I sincerely hope proves to be wrong. However, presently every indication says that our projections of demands for the Daisy Wheel II printer was grossly under the mark. We've done everything possible to speed up deliveries, but don't see the situation improving before late summer or fall. We're even trying to fly as many in as we can, and speed up production to the greatest possible extent. We all knew a reasonably-priced daisywheel printer would be a winner. However, we were unprepared for the more than overwhelming response it has received. Thanks for your support, and bear with us. We're doing our best to catch up! We'll be watching the demand on other items too, like the new Line Printer VII.

I'd like to close out by referring those of you who didn't read my March, 1981 column which dealt with what goes into a Radio Shack product. At this writing it's too early to have heard any feedback, but I believe the things I outlined are quite significant. Those TRS-80 owners or potential owners I've talked with by phone, and discussed the subject with were quite surprised to hear what we're trying to accomplish, and a number of them have indicated that it influenced their final "which to buy" decisions.

The point is, if you own a TRS-80, we'd like you to feel pride in ownership, and know that your ideas and opinions about new products, product improvements, and how we could gear up to serve you better are very important to us. We appreciate hearing your thoughts. Again, due to the limited resources for individual replies please direct routine questions to our Computer Services Group, 900 Two Tandy Center, Fort Worth, TX 76102. Send product or service suggestions to Computer Merchandising, 700 One Tandy Center, same city & zip. Thanks, and see you again next month.■

REMARKS

Continued from page 7

was saddened, but not surprised, to read in the school paper a couple years later that this teacher had committed suicide. He was a misfit in that school, possibly in any school. But at least, through him, many of us got just a taste of what education could be.

I think that if education is to survive, we must realize that it is a kind of show biz and accept the fact that education must be interesting. It really *can* be done... even in *accounting*. And, you know, while nearly everything else that I studied in college has been long forgotten, accounting has stayed with me—I still think of it as fun.

What about kids who are *not* interested in education? Is it really *our* responsibility to force feed them school work? Perhaps we should legitimize child labor, eliminate the minimum wage requirements for children under 18, and see what happens. Oh my bleeding heart! Would kids find that the alternative to education is less pleasant than school?

It is my intention with Instant Software to pursue the development of automated learning systems. I have in mind, at present, something akin to a video cassette which is coupled to a microcomputer. The micro would be programmed to show the video program and stop every now and then to make sure the student is keeping up with the material. If the student is unable to answer questions correctly, the machine would rewind and go over the misunderstood material again. A different series of questions would follow. In this way the progress of the course would be matched to the student.

If this should prove effective, I forsee a time when we might be using video disks. A particular program might cost several million dollars to produce, but it could be used to teach hundreds of millions of people. The lesson could be translated into any language and provide relatively low-cost education even for the most remote and poorest of third world nations.

Getting back to Erasmus for a moment. The clubs came to an end when the teacher's union came into power. The union insisted that teachers be paid overtime when supervising after-school clubs. The school had no budget for this additional expense, so there went all those wonderful clubs.■

Wordstar

One of our readers, Charles R. Perelman, has written a very handy book called *Shelter From the Tax Storm.*

The interesting part of the book, other than the contents, is that it was written on a TRS-80 Model II using Wordstar and CP/M. It was printed out on a NEC Spinwriter and photo offset directly from that printout. Chuck would have used TRSDOS, but couldn't find a driver for the NEC, hence the CP/M.

Chuck, by the way, is also a ham. Funny how many computerists are. WA6OGW.

If anyone is interested in the book... and almost everyone should be... it is available from Chuck for $10. His address is 9777 Wilshire Bvd #700, Beverly Hills, CA 90212. Mention me.■

META TECHNOLOGIES

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

The perfect supplement for your NEWDOS, from IJG, Inc.

"TRS-80™ DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

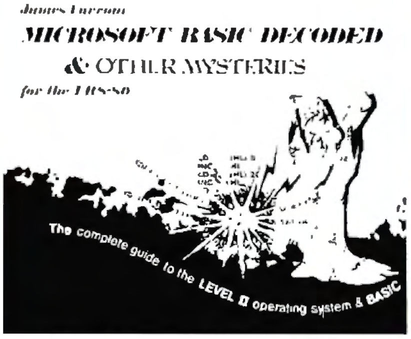
132 pages written in PLAIN ENGLISH packed with HOW TO information with details, examples and in-depth explanations. Recover lost files and directories, remove file protection, make BASIC programs unlistable. How to use SUPERZAP, recover from DOS errors and MORE!

TRS-80™ DISK $19.95

"OTHER MYSTERIES" VOLUME II

foreword by

H.C. PENNINGTON



Call now and place your order for this new book, "MICROSOFT™ BASIC DECODED & OTHER MYSTERIES for the TRS-80™", from IJG, Inc. A primer for cassette and disk BASIC on the TRS-80™, the information provided applies to similar MICROSOFT™ BASIC interpreters. Features include definition of terms, an overview of BASIC and DOS, explanation of exits, error codes, verb actions, "cold" and "warm" restart procedures and examination of system utilities, arithmetic support and I/O driver routines, and the communications region in RAM. Individual routines are explained in detail, with an index provided for easy access. Appendixes include tables for BASIC and DOS vectors, stacks and interrupt locations, PLUS thousands of comment lines for the complete MICROSOFT™ BASIC.

MICROSOFT™ BASIC DECODED . . $29.95

Let your TRS-80™ Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80® Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 . . . . $69.95
FOR DISK SYSTEMS $74.95

Let Your TRS-80™ Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 . . . . $29.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 ★SPECIAL★ $129.95
NEWDOS+ to
NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK
AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR
PERSONAL COMPUTER . . . . $ 7.95

1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER

BY MARK SAWUSCH

333 pages

$7.95

333 pages, written in simple terms, of "what-to-do" and "how-to-do-it". Suitable not only for microcomputers, but for programmable calculators as well. Includes program listings, formulas, a glossary of computer terms and more! Definitely a MUST BUY!

A PARTIAL LIST OF APPLICATIONS

Electric Memorandum	Income Tax	Simultaneous Equation Solver
Energy Efficiency Computation	Speed Reading	Real Estate Evaluation
Test Your Typing Speed	Check Digit Generator	Astrology
Statistical Analysis	Personality Test	General Purpose Clock/Timer
Options Analysis	Morse Code	Recipe Index/Calculator
Logic Circuit Analysis	Finances and Investments	RPN Calculator
Antenna Design	Biorythm	
Carpenter and Mechanic's Helper	Letter Writing	
Calculation of the Number of Days Between Two Dates		

MORE PRODUCTS ⟷

20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT May 1, 1981 THRU May 31, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8105

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

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

80 INPUT

"I can understand such a problem—hey—it happens to everybody. . . . What I want to know is this: How contagious is this, and how long does it last?"

Fantastic Formatter

The LPRINT Formatter II program by Charles Z. Tzinberg in the February issue of *80 Microcomputing* is fantastic!

I use a Centronics 779 printer and in order to get a listing of a program, I must use the smallest print available so as not to lose any of the longer lines. This makes it extremely difficult to read. Tzinberg's program allows the use of the largest print size and makes debugging a program a breeze.

I have one suggestion to improve the program. Currently, when the listing is finished, printing will cease in the middle of the page. By changing the following lines, the printer will advance to the top of the next page (after inserting the string of dashes, if you used that option) so that your last page will be uniform in size to the others. The changes in the following lines are underlined:

```
7 X+E: GOSUB2: S=PEEK(X): X=E+1: GOSUB2:
  S=S+PK*PEEK(X) :X=E+2: GOSUB2: L=PEEK
  (X): X=E+3: GOSUB2: L=L+PK*PEEK(X): IFL>
  HI or S=0 THEN GOTO15: ELSE IF L<LO etc.
15 LPRINTSTRING$(59—LN,138): LPRINTSTRING$
  (ML,"-"): LPRINTSTRING$(4,138): E=S: PG=PG
  +1: IFL>HI THEN END ELSE GOTO 6
```

Patricia M. Finkenbine
Albuquerque, NM 87111

The Thirteenth Place

13.3.1981: After considering all the circumstances from a slightly unbiased view, I can only conclude that your March issue should have never gotten near my Model I.

Everything was fine before it showed up: The drives, printer, interface and rest of the gang were merrily churning along at a consistent ten hours a day pace. No parity panics, no boot blowouts, no directory divebombs.

Then the March issue came. Since I'm seriously considering buying a Model II (thus will be trusting those folks in Texas with my hard earned bucks and future), the article about John Roach grabbed me by the granules. After all, if I'm giving him over four grand, I'd at least like to know if he's a good guy or if he'll scurry off to Tahiti with my money.

I turned to the contents page. Sure enough, there was a teaser about Roach, the new computer chip king, and another piece on Kornfeld, the outgoing DOS scepter holder. But there wasn't a page number. No problem I figured—scan the right hand page, the one with the DIR (I,A,S) type readout.

No luck. Okay, next step. Scan the right page again, this time looking for a filespec with Nancy Robertson's byline. That panned out for a story, but not the one I was looking for.

Using the standard humanoid system (Version 1.0 as best I can tell) I "superzapped" the magazine page (byte) by page. Snuggled in on page 82 I found it.

Having worked for monthly magazines for seven years, I can understand such a problem—hey—it happens to everybody. But at about the same time I was on safari for the article my TRS-80 started rolling over. Disk drive problems: Directory Read Errors, Bad Parity Errors, Motor Speed Too Slow, Motor Speed Too Fast, Dummy At Controls, etc. If it's in the Owner's Manual under disk errors, I got it on my screen.

What I want to know is this: How contagious is this, and how long does it last?

Brad Zimmerman
Fountain Valley, CA

Any computer person will have no trouble spotting the date of this letter. I also take it as read that you've broken a couple of mirrors lately. Sprinkle a pinch of salt over the back of the keyboard and all should be OK.—Eds.

An Unsympathetic Ear

H. C. Pennington's response in the March 1981 *80 Microcomputing* to an unfavorable review of his book, *TRS-80 Disk & Other Mysteries,* heaps scorn on the reviewer. Pennington seems convinced that the ability to hurl insults is a sign of superior intelligence.

When the invective is stripped away, Pennington's defense of his book is lame. The fact remains that *TRS-80 Disk* is of little value without the purchase of a particular kind of software, for which it is an extended advertisement. Neither this fact nor any indication that the book will be of interest only to a very limited audience is apparent in the promotion of the book. The book, like the article, illustrates Pennington's talent for abuse, with Radio Shack as the object instead of John Grass.

Four pages of my favorite computer magazine were wasted on this display of bad temper.

Malcolm Nygren
Champaign, IL

Pocket Computer Tips

Here are a few tips which I have discovered for use with the Radio Shack Pocket Computer:

While in the PRO mode, instead of typing LIST (or L.), just hit the ↓ key or the ↑ key. Holding the ↓ key down will LIST the program starting from the lowest line number and working up, while hitting the ↑ key will start from the highest line number and work down.

If a different "E" is needed for some reason, use the Exp (exponential) "E".

Finally, while using the DEF, RUN, or RESERVE mode to do a series of calculations, and you hit enter but want to check the figures you used, hit either the < key or the > key to review the numbers. Otherwise, you can edit them and just hit enter to get the new result.

Bradley Smith
Yarbo, Saskatchewan SOA 4VO

In Praise of Percom Data

When a customer gets a good response from a manufacturer, the news should be heard by someone. This was the case with me when I purchased a Percom TFD-100 Disk Drive.

I was plagued by chronic inability to make successful backup copies of my TRSDOS using TFD-100 and TRS-80 Model I, Level II. I was able to do anything with OS-80 but not with TRSDOS. I wrote to Mr. Jim Stuttsman at Percom Data Co. in July

META TECHNOLOGIES

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

MTC AIDS-III™

MODEL I . . .$69.95 MODEL II . . .$99.95

Introducing the latest addition to MTC's family of data management systems, AIDS-III. NO PROGRAMMING, easy to use. COMPLETE PACKAGE including demonstration application, documentation and MAPS-III (see below).

- Up to 20 USER-DEFINED FIELDS of either numeric- or character-type.
- CHARACTER-type fields may be any length (total: up to 254 characters).
- NUMERIC-type fields feature automatic formatting, rounding, decimal alignment and validation.
- Full feature EDITING when adding or changing records:
  - ENTER FIELD (can't type-in more characters than specified).
  - BACKSPACE (delete last character typed).
  - DELETE FIELD contents.
  - RESTORE FIELD contents.
  - RIGHT-JUSTIFY FIELD contents.
  - SKIP FIELD (to next or previous field).
  - SKIP RECORD (to next or previous record).
- SORTING of records is MACHINE CODE assisted.
  - 200 RECORDS (40 characters) in about 5 SECONDS.
  - ANY COMBINATION of fields (including numerics) with each field in ascending or descending order.
- SELECTION of records for Loading, Updating, Deleting, Printing and Saving is MACHINE CODE assisted.
  - Specify up to 4 CRITERIA, each using one of 6 RELATIONAL COMPARISONS.
  - LOAD or SAVE selected records using MULTIPLE FILES.
  - Example: Select records representing those people who live in the state of Colorado, but not in the city of Denver, whose last names begin with "F" and whose incomes exceed $9000.00.
  - Example: Select records representing those sales made to XYZ COMPANY that exceed $25.00, between the dates 03/15 and 04/10.

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge, has the following features:

- Full AIDS-III SELECTION capabilities.
- Prints user-specified fields DOWN THE PAGE.
- Prints user-specified fields in titled, columnar REPORT FORMAT, automatically generating column headings, paging and (optionally) indentation.
- Can create a single report from MULTIPLE FILES.
- Prints user-defined formats for CUSTOM LABELS, custom forms, etc.

BELOW ARE TESTIMONIALS from owners of AIDS systems. These are absolutely authentic statements and are typical of the comments we receive.

"This program will do more for my business than all the other programs I have, combined."
David Wareham, Vice President (EDP), National Hospital and Health Care Services Inc.

"We have 32 different Data Base Management packages for the TRS-80. AIDS-III is easily the best. It also makes it easier for us to step up to our Model II since the package is available for both computers."
Jack Bilinski, President, 80 Microcomputer Services

"Your AIDS program is far and away the finest information management system that I've ever seen. I am currently using it to maintain a clear picture of the demographic data on all the kids in our residential treatment program and it is working for me superbly."
Frank Boehm, Director, Front Door Residential Treatment Program

MTC AIDS CALCULATION SUBSYSTEM-III™

MODEL I . . .$24.95 MODEL II . . .$39.95

MTC's most popular AIDS subsystem. Use for report generation involving basic manipulation of numeric data. Features are:

- **User-specified page title**
- **Columnar Headings**
- **Optional Indentation**
- **Use for accounting, inventory, financial and other numeric-based information systems.**
- **Columnar subtotals generated when there is a change in a user-specified column.**
- **User-specified Columnar Totals**
- **Columnar values computed using constants and/or column values**
- **Balance forward calculations (Ex: Gross sales equals previous gross sales + sale amount + sales tax).**

Compare AIDS-III™/CALCS-III™ with any other data management package under $100!

CALCS-III™ REQUIRES THE PURCHASE OF AIDS-III™

AIDS OWNERS! WE HAVE WHAT YOU'VE BEEN WAITING IV. . .

MTC CALCS-IV™, that is.

- **More Computations**
- **Save Report Formats on Disk**
- **Faster, and more!**

CALL FOR DETAILS and CALCS-III UPGRADE PRICING

MTC AIDS MERGE-III™

This subsystem will combine up to 14 AIDS-created data files into a single, large file. An optional purge capability removes duplicate entries while performing the merge operation (can even be used to eliminate duplicates in a single file). Machine-code assisted for high-speed performance, MERGE-III™ properly handles files sorted by any combination of fields, including numerics, with each field in ascending or descending order.

MTC AIDS MERGE-III™ $19.95
For Model II $29.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 ★SPECIAL★ $129.95
NEWDOS + to
NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER . . . . $ 7.95

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.
©1981 by Metatechnologies Corporation, Inc.

20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT May 1, 1981 THRU May 31, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8105

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

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

1980. We exchanged several letters and worked hard.

On September 25, Jim found that the TRS-80 disk controller circuits activate the disk drive for 2.5 seconds. But in my unit, it may have been less than two seconds, which caused a premature "TIME OUT". So he made a special patch to enable a successful TRSDOS backup utility. It really worked! Not only was he observant and creative, but he was very patient with my request.

Shortly after the disk problem was solved, I purchased a Doubler from Percom. It was easily installed as was its data separater. The performance of the Doubler is superb. I am now enjoying both DBLDOS and OS/80D.

Thank you Jim Stuttsman and Percom Data Co.

Paul S. Satoh, PhD.
Portage, MI

RS-232 Interface

Recently there was an article in another publication discussing Electronic Systems' RS-232 interface. I am using their kit version and the bottom line is that it works.

However, other readers might be saved some grief by sharing my experience. In short, if you want to use it either exclusively for output or to input with BASIC at no more than 300 baud, you will probably be satisfied.

Unfortunately, it is not obvious how to use this device with assembler programs because the board expects the input port address on the high order address lines—not on the low order lines where the Z-80 puts it. In spite of this design error, the board works with BASIC because of a fluke, for BASIC happens to leave the value coded as the input port in the A register which the Z-80 latches on the high order address lines as part of executing this input instruction, but it isn't really a port address! In other words, the actual port address specified in the input instruction is essentially ignored.

While the board happens to work with BASIC, there is potential for disaster in assembler if both the port address and the input register were to address different hardware devices. The solution is quite simple: Pull IC U6 and bend pin 2 (not pin 1 as the schematic indicates) up at a 90° angle to the other pins so that it will not go back into the socket. Solder a jumper from pin 2 up over the top of the IC to pin 13; reinsert U6. This breaks the connection from U8 pin 8 to U6 pin 2 and connects instead U4 pin 8 to U6 pin 2 which changes the input port address from the high order to the low order address lines, from 55 (37H) to 248 (0F8H). The address from which status is loaded and into which output data are loaded remains 37F8H. Thus in BASIC the only change is to use INP(248) instead of INP(55).

If you are concerned about possibly destroying something, note that this does nothing to the printed circuit board. Also, the IC costs 29¢ at Hobbyworld. And should you ever need to, that same 29¢ IC would let you return to the original functioning of the board.

All of the above of course assumes no change in design—so check your board (not just the schematic). To spare you a long sad tale of woe, suffice it to say that, once you get it to work, it's a nice device with DIP switch selectable baud rate, parity, word length, and stop bits, but Heathkit it ain't.

Edgar C. Howell
Simi, CA

Tax/Saver Review

We thank Michael Tannenbaum and Gordon E. Lamb for reviewing Tax/Saver in the February issue of *80 Microcomputing*. We are gratified by the favorable reviews Tax/Saver has received.

We would like to comment on some valid points raised by the article.

Several features not included in the demo which was reviewed are built into the marketed version. An example is an option which allows the user to stop the session at numerous controlled points, turn the computer off, and resume processing from that state at a later time.

Tax/Saver covers many schedules and forms in depth and detail. It accepts totals from all other schedules and forms.

We would like to point out that Tax/Saver is more than a program; it's a service. Subscribers also receive our Tax Facts newsletter and are entitled to a discount on the yearly update.

Netta K. Stern
Micromatic Programming Co.
Georgetown, CT 06829

New Patches

Several months ago you published a letter from us describing a set of patches to Model II TRSDOS version 1.2 which moved the break key function out of the operator's way so it would not be pressed accidentally while reaching for the back-space key. Now that TRSDOS 2.0 (and 2.0a) are almost universally in place, I thought your readers would appreciate the equivalent patches for the newer releases:

```
PATCH SYSRES/SYS A=0407,F=03,C=FE
PATCH BASIC A=57E7,F=03,C=7E
PATCH BASIC A=594C,F=03,C=7E
PATCH BASIC A=5F70,F=03,C=7E
PATCH BASIC A=6606,F=03,C=7E
PATCH BASIC A=6515,F=03,C=7E.
```

Be sure to reset the system after applying the patches to make them fully effective.

One word of caution: SCRIPSIT uses <ctrl>6 as a function key, therefore you *must not* apply these patches to a SCRIPSIT disk.

Robert F. Snapp, President
Snapp, Inc.
Cincinnati, OH

CROSSREF Mod III

CROSSREF, (*80 Microcomputing*, January 1981), will work fine on a TRS-80 Model III if you make two changes to Part A, as follows:

In line 65502, change 17128 to 17384
In line 65504, change 17127 to 17383
Part B needs no modification.

To append Part A to your program on a Model III, one small but vital change must be made to the procedure described in Table 1. The last operation in Step 4 should be changed from POKE16549,66 to POKE16549,67.

These differences between Model I and Model III result because Radio Shack advanced the starting address of BASIC programs by 256 bytes. Unfortunately, the Model III manual incorrectly shows this address as being the same as for Model I.

Donald N. Ewart
Scotia, NY

Review Umbrage

Eds. Note: The following is from a letter sent to Dennis Thurlow, author of the review in question.

As a partner in Rational Software and the author of the program "Programmer" which you reviewed in the January, 1981 issue of *80 Microcomputing*, I would like to comment on your review.

The next-to-last paragraph contains a statement ("Once memory has been protected for a program... no more overhead should be necessary for its operation.") which is at best a personal prejudice of your own, and at worst, simply untrue. You would not expect a text editor or an interpreter to function without using external memory; why then demand that

META TECHNOLOGIES

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

MTC AIDS CALCULATION SUBSYSTEM-III™

MODEL I . . .$24.95 MODEL II . . .$39.95

User-specified page title

Automatic Page Numbering

Columnar Headings

User-specified initial balance forward

```
==============================================================
CUSTOMER ACTIVITY REPORT                              PAGE  1

CUSTOMER DATE QTY     SALE AMT    SALES TAX GROSS SALES $/UNIT
======== ==== ======= =========== ========= =========== ======
                                                   0.00
ACME     3/10     100      675.00     37.13     712.13   7.12
                  200     1325.00     72.88    2110.00   6.99
         3/20     400     2475.00    136.13    4721.13   6.53
         4/10     600     3625.00    199.38    8545.50   6.37
         4/20     400     2600.00    143.00   11288.50   6.86
              ------- ----------- ---------
                 1700    10700.00    588.50

META     3/10     200     1345.00     73.98   12707.48   7.09
         3/15     100      674.00     37.07   13418.55   7.11
                  200     1295.00     71.23   14784.77   6.83
         4/05     400     2435.00    133.93   17353.70   6.42
         4/10     150      935.00     51.43   18340.12   6.58
         4/20     600     3585.00    197.18   22122.30   6.30
              ------- ----------- ---------
                 1650    10269.00    564.80

QURCO    3/25     200     1325.00     72.88   23520.17   6.99
         4/10     100      685.00     37.68   24242.85   7.23
                  300     1940.00    106.70   26289.55   6.82
              ------- ----------- ---------
                  600     3950.00    217.25

XYZCO    3/10     150      995.00     54.73   27339.27   7.00
                  200     1345.00     73.98   28758.25   7.09
         3/20      50      355.00     19.53   29132.77   7.49
         4/10     300     1975.00    108.63   31216.40   6.95
         4/15     400     2520.00    138.60   33874.00   6.65
         4/20     700     4175.00    229.63   38279.62   6.29
              ------- ----------- ---------
                 1800    11365.00    625.08

              ======= =========== =========
                 5750    36284.00   1995.62
              ======= =========== =========
```

Optional Indentation

Columnar values computed using constants and/or column values

Columnar subtotals generated when there is a change in a user-specified column.

Balance forward calculations (Ex: Gross sales equals previous gross sales + sale amount + sales tax)

User-specified Columnar Totals

Compare AIDS-III™/CALCS-III™ with any other data management package under $100!

Others make claims, CALCS-III™ delivers with user-specified:

- Fields in any order, with optional indentation
- Computations using field values and constants
- Columnar subtotals and totals
- Full AIDS-III selection of records to be printed

Use for accounting, inventory, financial and other numeric-based information packages.

CALCS-III™ REQUIRES THE PURCHASE OF AIDS-III™

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.
1981 by Metatechnologies Corporation, Inc.

✓20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT May 1, 1981 THRU May 31, 1981.
Prices, Specifications, and Offerings subject to change without notice.
8105

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

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax

other utilities be self-contained? Further, by making that statement without providing any information as to when or why Programmer uses external memory, the entire purpose of the paragraph is lost to the reader. I find this especially irritating as I took some pains to make clear in the instructions when, why, and how external memory is used.

Finally, I'd like to say that the two bugs you reported have been removed from the program, and though it still uses external memory (and always will), our users are (as you said) finding it extremely useful.

Tom Little
Rational Software
Pasadena, CA 91106

Microsoft Computer

I purchased my TRS-80 16K system in June 1980. I had a pressing need to manipulate strings but no computer background. It quickly became apparent that due to the 15,000 characters per file I sought to manipulate in memory, plus my inability to produce efficient code, I must upgrade to a 48K double drive disk system.

After an ineffably frustrating sixty days (and nights) and a weight loss of 25 lbs. (which I won't miss)—my program accomplished the file manipulations intended.

Then reality set in. The program was slow. So slow, in fact, that completion of all the assigned manipulations averaged just under five seconds per string. After boggling my way through several texts and borrowing more techniques from periodicals, I managed to reduce the average time to 2.7 seconds; still unpleasantly slow considering that one program cycle involved six or seven files of 600 to 700 strings per file.

Then the Microsoft compiler arrived. After the initial study period with the manual I put my program titled CMBSCORE/BAS (residing on a TRSDOS disk stripped of utilities and BASIC) in drive 0. The two disks that comprise the compiler program were installed in the other two drives. The rest was simple, even for me. I typed CMBSCORE= CMBSCORE (ENTER) and waited for the whirring and an occasional 'clunk' to stop (the clunk from an older style drive my friendly Tandy computer specialist passed off on me).

Next, my display read "(0000) FATAL ERRORS." Scared the socks off me. Rereading the manual convinced me that the (0000) was the important part of the message and the FATAL ERRORS was there, at least this time, just to keep my blood pressure up.

The next step was also easy. Type CMBSCORE, CMBSCORE-N-E (ENTER). Several minutes later the compiled program was ready. I typed CMBSCORE (ENTER) and the program loaded and was displayed. I specified the files to be manipulated and watched the strings fly by. The processing time was now .16 seconds per string instead of the previous 2.7 seconds.

I now process files containing up to 5500 characters each plus 700 or more five-digit integers at an entirely safisfactory speed. Incidently, several games move so fast that I am destroyed, eaten, mauled or gassed by trolls, goblins and satellites before I can press a finger in my own defense. Graphics flash rather than sputter. Another big, big bonus: TRS-80 no longer withdraws into that familiar catatonic state to adjust its brains. This is no great news to all of you experienced assembly programmers, but for beginners this is an exciting feature.

I am "Bardenizing" myself, so in due time I may master some assembly language. In the meantime, that Microsoft compiler will be my close companion. It may, with my blessing, take all the room in memory it wishes. The least I can do for all the service it renders is to live with what I consider to be more than adequate remaining memory and disk space.

David W. Jessup
Seattle, WA

More Mail File System

M. Barlow's letter in the February *80 Microcomputing,* stunned me. I thought I might pass along my experience with Galactic Software's "Mail File System."

I am involved in a business that involves taking long orders from over 300 customers, updating deliveries and mailing weekly solicitations to the same. No data base program is going to run as fast as I can manually do the filing and updating—not to mention the enormous amount of memory I would need. But with clever coding (if you don't mind me saying so, myself), I have turned the "Mail File System" into a timesaving way to print labels, and an outstanding data base program.

The zip code listing gives me an idea of which accounts are geographically near others, helping me plan sales trips. By coding accounts, I can systematically break down when an account last bought a particular category of items from me. It takes about two hours manually to do this; 10 minutes, total, on "Mail File System".

But, you're saying, any mailing list system can do that. I don't care. All I know is that this system works for me. My office help, with no computer experience, can run it without me being there.

When I had a question, I called Galactic Software, and they gave me a useful answer. When I thanked them for the answer, they said, "That's what we're here for." Boy, that's great to hear!

Yeah, the program isn't perfect. There is a bug in the printing test labels, but it is easily corrected. And I wish I could do form letters and have the program insert names. Maybe, someday, it will (hint). As I said, nothing is perfect, but I am willing to accept a product that "works first time, every time", and this program does.

Neal Elkind
1004 The Apparel Center
Chicago, IL 60654

Paper Tiger

The article entitled "Paper Tiger Screenprinter" by Ruth Lewart in your February 1981 issue was excellent. Her explanation about creating a software alphabet enabled me to construct data statements for the lowercase letters, which were not incorporated in her data.

Below are the data statements for lowercase and some special characters, along with the appropriate program statement changes:

```
10850 DATA 72,126,73,66,32, 32,84,84,84,124, 127,40,68,
68,56,56,68,68,68,40   'lb. symbol, a, b, c
10860 DATA 56,68,68,40,127, 120,84,84,84,24, 0,8,126,9,2,
8,84,84,84,56   ' d, e, f, g
10870 DATA 127,8,4,120,0, 0,68,125,64,0, 0,32,64,64,61,
127,16,40,68,0   ' h, i, j, k
10880 DATA 0,65,127,64,0, 124,4,120,4,120, 124,8,4,4,120,
56,68,68,68,56   'l, m, n, o
10890 DATA 124,8,20,20,8, 8,20,20,8,124, 124,8,4,4,8,
72,84,84,84,36   ' p, q, r, s
10900 DATA 0,4,63,68,32, 60,64,64,32,124, 28,32,64,32,28,
60,64,48,64,60   't, u, v, w
10910 DATA 68,40,16,40,68, 12,80,80,80,124, 68,100,84,76,
68,8,54,65,0,0   'x, y, z, (
10920 DATA 0,0,119,0,0, 0,0,65,54,8, 41,42,124,42,41,
85,42,85,42,85   ' vertical line with space in center, },
undefined character, undefined character
10540 DIM QG$(7,3),QA$(96,3),QM(42),QN(64)   'GRAPH-
IC, ALPHANUMERIC STRINGS AND TABLE ENTRIES
10580 FOR QI=0 TO 95: FOR QJ=0 TO 4 ' CREATE
ALPHA STRINGS
```

Note: Remember to clear extra string space in the driver program. CLEAR 3000 should be sufficient.

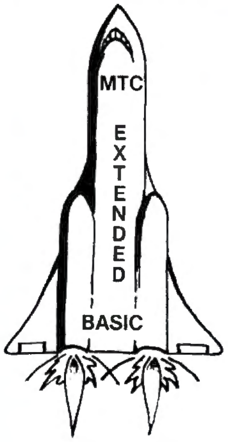
There appears to be an idiosyncrasy in my TRS-80. When running the Screenprinter with a game program, the first several alphanumeric characters PEEKed

Continue to page 20

META TECHNOLOGIES

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

TAKE YOUR MODEL II TO NEW HEIGHTS



MTC is proud to announce MTC EXTENDED BASIC for the Model II, by R. Ryen. Features include "fixes" to existing BASIC, multi-line functions, extending an existing sequential file, PEEK, POKE, greatly enhanced screen control and expanded editing capabilities. The contents of variables are NOT CHANGED when editing, deleting, inserting or merging lines, allowing continued program execution! All this and much more.

MTC EXTENDED BASIC .... $ 99.95

80 − 20 = 129^95

THE ORIGINAL NEWDOS/80

WITH MTC QUE CARD

★ SPECIAL PRICE ★

SAVE $20 **$129.95**

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!). Includes 180-page instruction manual and MTC QUE card.

NEWDOS/80 # SPECIAL # $129.95
MTC QUE Card only .... $ 7.50
CALL REGARDING OUR NEWDOS+ UPGRADE PRICING

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 .. $26.95
8" DATALIFE™ FD34-8000 .... $43.95

NEWDOS by APPARAT

NEWDOS/80 ★SPECIAL★ $129.95
NEWDOS+ to
NEWDOS/80 UPGRADE ..... CALL
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK
AND OTHER MYSTERIES .. $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR
PERSONAL COMPUTER .... $ 7.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

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

SYSTEM DIAGNOSTICS $19.95

An improved version of the SYSTEM DIAGNOSTICS above. Designed for single or double density, 35-, 40-, 77-, or 80-track disk drives. Includes new and modified tests. Features THE FLOPPY DOCTOR, Version 3.0

SYSTEM DIAGNOSTICS-V3 . . $24.95

Complete for Model I with all utilities Plus exclusive MTC QUE card!

NEWDOS +

$69^95 by Apparat

Includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

40-Track Version $79.95
MTC QUE Card only .... $ 1.50

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC. 1981 by Metatechnologies Corporation, Inc.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY
Products damaged in transit will be exchanged.

PRICES IN EFFECT May 1, 1981 THRU May 31, 1981.
Prices, Specifications, and Offerings subject to change without notice.
8105

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

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

Yes, You Can Increase Your Programming Productivity

SNAPP II EXTENDED BASIC

SNAPP II EXTENDED BASIC A family of enhancements to the Model II BASIC interpreter. Part of the package originated with the best of APPARAT, INC.'s thoughts in implementing NEWDOS BASIC. The system is written entirely in machine language for SUPER FAST execution. The extensions are fully integrated into Model II BASIC, and require NO user Memory, and NO user disk space. The package is made up of the following six modules.

XBASIC—Six single key stroke commands to list the first, last, previous, next or current program line, or to edit the current line. Includes quick way to recover BASIC program following a NEW or system or accidental re-boot. Ten single character abbreviations for frequently used commands: AUTO, CLS, DELETE, EDIT, KILL, LIST, MERGE, NEW, LLIST, and SYSTEM.

XREF—A powerful cross-reference facility with output to display and/or printer. Trace a variable through the code. Determine easily if a variable is in use.

XDUMP—Permits the programmer to display and/or print the value of any or all program variables. Identifies the variable type for all variables. Each element of any array is listed separately.

XRENUM—An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered, supports relocation of renumbered blocks of code, and supports duplication of blocks of code.

XFIND—A cross reference facility for key words and character strings, also includes global replacement of keywords.

XCOMPRESS—Compress your BASIC programs to an absolute minimum. Removes extraneous information; merge lines; even deletes statements which could not be executed. Typically saves 30-40% space even for programs without our REM statements! Also results in 7-10% improvement in execution speed.

ENTIRE PACKAGE FOR MODEL II $200
ENTIRE PACKAGE FOR MODEL III $125

DOUBLETAKE 3741

This is not a football play but the way to play ball fast in converting IBM 3741 and similar formated diskettes to Radio Shack formated disks or vice versa. Fast is the name of the game. $200

SNAPP SCREEN

Another dependable product from Snapp, Inc. Using rigid plastic, Snapp, Inc. designed the ultimate green screen at the lowest price. The Snapp Screen installs in a Snapp with the use of pressure sensitive tape. Reduce user fatigue with the Snapp Screen. $10

2.00 shipping and handling—waived with the purchase of any software.

SUPERSNAPP X

You can sort fast using your present facilities, or you can do it faster with Racets' superb facilities, or you can use the fastest: SUPERSNAPP X. The heart of SUPERSNAPP X is a SUPER FAST in-memory sort routine that has been benchmarked against everything on the market and beats them all . . . hands down.

SUPERSNAPP X is the most important component of SNAPP X EXTENDED BUILTIN FUNCTIONS which is a much needed set of additions to the Model II BASIC interpreter that will put time saving power at your fingertips. Let's compare (using random data) SUPERSNAPP X and Racet's GSF SORT for speed:

SORT	SUPERSNAPP X	RACET GSF
10,000 integers	39 seconds	59 seconds
5,000 Singles	22 seconds	34 seconds
2,000 Strings	10 seconds	15 seconds

SUPERSNAPP X is guaranteed to be the FASTEST in memory SORT on the market or your money back. With it you also get these EXTENDED BUILTIN FUNCTIONS. PEEK, PEEKW, POKE, POKEW, XDATS, XTIMS, ETIMS, FILES, AND THE SPECIAL SCMD (SNAPP___COMMAND). PLUS, open "E" Set SCROLL PROTECTION. ERASE all ARRAYS in one command. Specify size and Blink rate of CURSOR. Long ERROR messages. Read from Video Screen Read PEEK complete strings from memory. POKE complete strings to memory, convert upper case to lower case and vice-versa, turn complete screen off and on at will, extract largest or smallest values from user supplied list of numbers and probably the most powerful addition ever made to MICROSOFT BASIC print using into a string. Use your imagination with this function, it will perform wonders. $100.00

SPOOLER Model I, Model II and Model III

Our workhorse! Unlike the one supplied with TRSDOS 2.0, ours requires no special knowledge or training on the part of the operator. Additionally, ours performs much better. On the Tandy SPOOLER, every time a disk is accessed, the printer stops dead! This package is available for Model I, in the TRSDOS/NEWDOS 80 versions, or for the Model II or Model III. Greatly enhances system performance when running typical business applications. Many applications have been benchmarked to run nearly TWICE AS FAST with the SPOOLER installed. Installs in minutes and no changes are required to your programs. The Model II version requires NO user memory. If purchasing more than one system, call for price quote. $100

AUTOFILE

Another big time saver from Snapp, Inc. Autofile offers all programmers much needed time saving tools to use when working with direct files. Eliminate tedious commands such as LSET's, MKIS's, MKDS's, MKSS's, CVD's CVI's and CVS's. When used in conjunction with Automap an update program can typically be reduced by as much as 90%. $50

AUTOMAP

Save time creating a formated screen with our Extended Basic Mapping Support system. The programmers task of communicating and displaying information with the user operator is completely automated with simple SEND and RECEIVE statement commands. Many different types of information can be sent and received from the operators with the use of a single command. Many hours of programming time are saved and cost efficiency is truly realized using Automap. $75

VSAM

Virtual Snapp Access Method. V.S.A.M's development was built upon several man years of research and development by Snapp, Inc. studying all previously available access methods to small computers. What evolved was the ultimate keyed data access method available for any small computer. All records in the Data Base are truly variable in length to maximize the utilization of valuable disk space on TRS 80 systems. Irrelevant blanks are never physically recorded on the diskettes leaving only necessary data actually stored. Data Bases are defined with one primary key. Retrival with this key is quick and CONSISTANT without regard to maintenance activity levels. Additional fields may be described as alternate key fields. Access time using alternate key fields will be slower than through the primary field but it also will be CONSISTENT without regard to Data Base maintenance activity levels. VSAM is fully integrated into BASIC with commands easy to learn and use. $500.00

DOTWO

Allows programmer to build a list of any commands. All languages are supported. With DOTWO a programmer is now allowed to add any new commands of any kind and not be limited to subset restrictions. Anything that can be typed into the keyboard makes an acceptable command with DOTWO files. A programmer can virtually build an unlimited series of commands and not be limited to just TRSDOS commands. Included at no charge: BUILD/BAS which gives a programmer an easier method of creating and maintaining DOTWO or DO files. $100.00

HOSTII/TERMII

Allows remote control of a Model II from another Model II, or any ASCII terminal. Our Host system, unlike the one supplied with TRSDOS 2.0, supports accurate screen positioning on the Term station. Without this feature, formated displays appear on the screen looking like randomly placed garbage. Requires NO user memory! This system is designed to provide software support to our customer locations without ever leaving the office. Custom versions are now available for most nationally distributed terminals as a $25.00 option. Call for details. $50

Preliminary Model III version now available with standard host/term features. Special May intro Price $35

PL/B

The most powerful logical components have been selected from the worlds most powerful language to give you the most unbeatable program development tool to improve your programming productivity! Snapp, Inc. has merged the most useful components of BASIC with the power of PL/1. An unbeatable structured programming facility resulted that can increase programmers productivity up to 50%. Unlike competitive products, which are slow rigid preprocessors' representing a primitive attempt to bring Structured Programming concepts to the Micro-Based user, PL/B is today's fourth generation answer to a fourth generation problem! PL/B can give you triple the power of the earlier preprocessors at triple the speed. PL/B is fully integrated into the BASIC interpreter. Two modes of operation are available. Transparent (hidden) or Compile and save mode. Increase your productivity and save time with PL/B. $200.00

BREAKOUT

With Breakout you can learn the internal workings of TRSDOS 2.0 using any standard disassembler. Find your way out of the maze and save programming time by eliminating the unneeded steps in TRSDOS 2.0. This is a must product for all advanced TRS80 programmers. $200

DOSFIX

A collection of patches to TRSDOS and BASIC to enhance their usability and function. Includes our well known BREAK7E patches and facilities to disable and verify detect which will increase average disk speed by 30%. Free with any Model II Software Package. Purchased separately. $10

3M SCOTCH DISKETTES

Double density certified 8" Floppies for the Model II. Better quality is not available at any price. Ten diskettes to a box.

3M

Quantity (boxes)	Price Per/Box
1	$34
5	$33
10	$32
20	$31

$3.00 shipping charge. This charge is waived if software is purchased on same order.

TERMS OF SALE

Shipments normally made within one day of receipt of your order. Customer normally pays shipping cost, except pre-paid orders including software, in which case shipping cost is absorbed by Snapp, Inc. C.O.D. orders will have standard C.O.D. fee added to shipping charge. Net 10 days will be granted to governmental agencies, educational institutions and well rated business firms. Please include purchase order number. Ohio residents add 5½ % sales tax or exemption certificate.

OUR GUARANTEE:

If your diskette arrives damaged, we will replace it without charge. If you ever accidentally damage it, we will replace it for a $10 handling charge. For a period of one year, we will provide you with any enhancements or updates for a $10 handling charge. For a period of one year, if errors are discovered in the programs, they will be corrected without charge. In the event we cannot correct an error, you may return the program material for a refund. Electric Pencil is a trademark of Michael Schrayer Software, Inc. TRS-80 and TRSDOS are trademarks of the Radio Shack division of Tandy Corporation.
NEWDOS and NEWDOS/80 are trademarks of Apparat, Inc.
RACET GSF is a trademark of Racet Computes.

SNAPP, INC.
SNAPP, INC.
SNAPP, INC.
SNAPP, INC.
SNAPP, INC.
SNAPP, INC.

See us at the National TRS 80 Microcomputer Show May 21-22-23 in Booths 74 and 75.

Call our toll free number: 1-800-543-4628
Ohio residents call collect: (513) 891-4496

SNAPP, INC.
8160 Corporate Park Dr.
Cincinnati, Ohio 45242

have 64 subtracted from their ASCII code. Any spaces PEEKed prior to these characters have the correct code. I changed the following program statement in Screenprinter to overcome this weird occurrence.

```
10090 FOR QI=0 TO 63:QN(QI)=PEEK(QS+QI):IF
QN(QI)<32 THEN QN(QI)=QN(QI)+64:NEXT
```

Perhaps one of your readers can solve this mystery.

William D. Webb
Norristown, PA 19401

Ode to Soft Sector

The following is a copy of a letter I sent to Soft Sector Marketing regarding their word processing system, Lazy Writer. I don't know these guys or anything about them, but I do know that I am so pleased with what I got from them that I have to share it:

I recently purchased "Lazy Writer", to be used with my TRS-80 with 32K, two disk drives and a Line Printer II. I am doing Oil and Gas Leasing that requires much paper work that I have to type myself, and I have been using two of the other systems. I bought your product sight unseen, and I was not sure what I would be getting. You seemed very open and proud of what you had to offer, but I naturally put that off as just the standard sale pitch, or, worse yet, the con artist at work.

I have often bought things I was later sorry I bought because they did not perform the way they should have or the way I thought they should have performed. A few purchases proved to be winners. *But I want you to know that I consider this purchase from you the best I have ever made for performing the function it was intended to do.* If you have any callers who want to talk to a user that can't believe how lucky he was to make a sight unseen purchase, give them my number.

A.M. Phelan, Jr.
Beaumont, TX

Applied Applications

Re: "80 Applications", February, 1981: I found two errors in this otherwise fine article.

Page 42: The A4 address line is not necessary to address the range of 3000H to 37DFH. Remove the connection from A4 and use gate Z2C as an inverter between Z1 pin 8 and Z2D pin 12.

The other, on page 44, is not really a fatal error, but the wires on pins 18 & 20 should be reversed. To quote page 2-10 of the *1980 Intel Component Data Catalog,* paragraph "Output or-Tieing": "To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 18) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus."

The $\overline{CE}$ pin 18 is a power-down standby mode which when brought high, puts the output into a high impedance state and the power consumption to a fraction of what it is in the read mode.

Chuck Barnett
Hanover Park, IL 60103

Reader Barnett is correct on both points. The Radio Shack communications addresses are not completely decoded, meaning that an active A4 address line could select an undesired "phantom" communication location. In fact, to be safe, disconnect not only A4 but also A5 (from Z1 pin 6), and connect pins 6, 11 and 12 of Z1 together.

As for the 2716 pin selection, the 1978 Motorola data sheet I used for this design briefly noted the standby mode, but did not make the Intel recommendations. I agree with Mr. Barnett and Intel that pins 18 and 20 should be switched.

Dennis Bathory Kitsz
Roxbury, VT 05669

Larger Forwarding Address

I enjoyed the article by Bertram A. Thiel titled "No Forwarding Address", in your March issue. Since I use GOSUBs extensively in my programs, I found Program Listing 2 most useful. However, since my machine is a 32K disk system, I had to make some modifications to the routine to make it work for larger programs.

My program listing includes a new line 65305 for addresses above 32767. Line 65300 was changed to show Disk BASIC's program starting address. Lines 65340 and 65360 were changed to GOTO 65305.

James R. Albright
Daytona Beach, FL

PROGRAM LISTING FOR 32K DISK SYSTEM

```
65299 END
65300 P1=27206
65305 IFP1>32767THENP1=-1*(65536-P1)
65310 P2=PEEK(P1)+PEEK(P1+1)*256
65320 IFPEEK(P1+2)+PEEK(P1+3)*256=65298THEN
      65370
65330 PRINT"TO SAVE LINE NUMBER";PEEK(P1+2)+
      PEEK(P1+3)*256;"THEN ENTER A Y ELSE JUST
      PRESS ENTER";
65340 INPUTP3$:IFP3$="Y"THENP3$="":P1=P2:
      GOTO 65305
65350 POKEP1+2,0:POKEP1+3,0
65360 P1=P2:GOTO65305
65370 DELETE 65299-65380
65380 END
```

Further Variations

"80 Input" in the February issue printed a letter from Doug Walker about the starting point for BASIC programs and variations between Level II and TRSDOS 2.3.

Both figures quoted (27172 and 26302) are correct. They are, however, mutually exclusive in that the figure depends upon the number of files asked for when BASIC is invoked. The first number is for three files and the second for zero files. For those who use NEWDOS + or NEWDOS80 the figures are again different.

Files	2.3	DOS +	DOS80
1	26592	26230	26604
2	26882	26520	26905
3	27172	26810	27206
	etc.		

My suggestion would be to use the address already stored in RAM at 16548 and 16549 with the convention POKE (address), PEEK(16548) and POKE(address), PEEK (16549). The PEEKs could be gotten in an earlier statement in the program and stored for use at the proper point.

Sim Tyler
Overland Park, KS 66212

Likes Meta Tech

I would like to tell my fellow readers about a product I purchased. Meta Technologies of Euclid, OH sells Aids3/Calcs3, and I would recommend it to anyone who needs a good data management system.

Two reasons prompt me to recommend their product. The system works exactly as they say it will and every time I run it I am amazed at the things it will do. The people at MTC will give prompt, courteous, competent answers to all questions if you run up on any problems. Since I was new to disk drives and all the problems they can cause for a beginner, I did have problems. MTC helped me with these problems and with their help I went on to have a system that has helped me a great deal in my business.

It is refreshing to find a company such as MTC that will do what they say they will do and go out of their way to help you when help is needed.

Joe L. Peacock, President
Peacock Plastics Company
Jacksonville, Texas 75766

80 AID

Sargon Players

I enjoy playing chess with Sargon II on my TRS-80 48K Model I, but many times the game lasts longer than I expect to play. I must then abort the game or write everything down and reset the chess board when I wish to play again. I thought that if I could load in RSM-2D first, play the game, then save the game to disk, when I want to renew my game all I would have to do is load the game back in under RSM-2D, 32D.

The steps which I follow are:

(a) Load in RSM-2D, 32K.
(b) Reboot, hold auto key and type in Sargon II/CMD.
(c) Play game.
(d) Hold break key and reboot.
(e) Hit enter—memory size appears.
(f) Hit enter—Radio Shack Level II BASIC.
(g) Type System. Computer responds *?.
(h) Type /44032. Computer responds Command.
(i) Type $20 5000 20. Computer Responds "Disk Error"

Help! Can someone tell me what I am doing wrong? I tried different combinations to save the program, but to no avail. How do you know how many sectors to save, and where to start? Does anyone know a better way to save a Sargon game to disk?

Anthony Rattenni
173 Westminster Dr.
Yonkers, NY 10710

Aid Reward

Since last November we have had our Model II with Scripsit and the new Daisy Wheel Printer II up and running in our office. Aside from a few bad disks, the machine has operated flawlessly and has been a tremendous help getting the work out.

There is one quirk which I hope someone out there can help us with. Although the printer has both a section sign and a paragraph sign, Scripsit won't print them. Since we constantly refer to sections and paragraphs of various statutes this is more than a little annoying. It is all the more annoying when the Scripsit manual shows how to print a British pound sign and various foreign language accents.

We will gladly trade the British pound sign for a section sign. How do we get inside the program to change the codes that drive the printer? Since we know very little about computers, we hereby offer a reward of $15 to the first person who can give us a fix for this.

Petralia, Neild & Webb
811 First Federal Plaza
Rochester, NY 14614

DVORAK Aid?

I read with interest the article by Waldo T. Boyd about the DVORAK keyboard *(80 Microcomputing*, December 1980). It's all the rage...even *Time* Magazine has mentioned it.

As Boyd mentioned, his program cannot be used with Electric Pencil or Scripsit since they do not use the keyboard routines in the ROM. Has anyone figured out how to patch the DVORAK keyboard software into Pencil or Scripsit? The keyboard is not much use if it can't be used with a word processor.

Any help would be greatly appreciated.

John T. Phillipp, MD
127 E. 45th St.
Brooklyn, NY 11203

Paragraph Problems

I see that SCRIPSIT/Selectric problems still seem to generate letters from your readers. I, too, have such an arrangment and problem. My problem is in the paragraph indents and the slop-over of the first sentence of each paragraph.

I have tried for a fix to my problems. Thanks to your SCRIPSIT articles and letters from readers I was made aware of the so-called "fix" that Tandy would provide. I now have my three sheets of meaningless gibberish. My last response from Tandy was to a letter I sent them explaining my problem and furnishing examples. Their response was that they did not feel my problem originated with their equipment, but rather in the ASCII/BCD interface. At first I thought this was a typical copout, but the more often I hear about different problems, the more I tend to believe them.

My Selectric is a Model 2740 and I use an interface provided by Micromatic 80. I am going to pursue my problem with the interface manufacturer. If any reader happens to have anything that would contribute to solving this problem, I certainly would appreciate the help.

R. S. Walters
203 N. Patton
Arlington Hts. IL 60005

CRT Trouble

I have set out to follow what seems to be an impossible course. My CRT in my "80" is failing. I would like to replace it with a genuine green phosphorous CRT. I have called all local suppliers and they treat me as if I had the plague.

I received a tip from a repairman for Bunker/Ramo computers. He said that Clinton Mfg. makes their CRTs. However, I have not been able to find any sort of address for them.

Could you give me the name of a supplier that makes these things? Or maybe the address of Clinton Mfg.

Anthony E. Kazee
1425 No. Mojave Rd.
Las Vegas NV 89101

RST Aid Needed

I have been disassembling Level I BASIC for relocation in high RAM. I would very much like to contact anyone who could give me some information regarding the restarts (RST) within the language.

Andy Pickett
520 S. Center St.
Plainfield, IN 46168

Portable Store

Recent comments in your letters column have compelled me to grapple with a real typewriter keyboard (no numeric keypad, lowercase) and rally to your defense. The comments I refer to have to do with the amount of advertising in the pages of *80 Microcomputing.*

It seems that some of your readers do not approve of the large amount of ads. Are they nuts or do they suffer from a heart condition which makes it hard for them to turn so many pages?

I live in a small town (30,000). We have two Radio Shack stores. The downtown store personel are quite knowledgeable about computers. The guys out at the shopping center readily admit that their computing skills are limited to playing blackjack. Outside of these two stores, there is no other source within 120 miles for computer items of any significance.

The pages of *80* are my computer store. It is in these pages that I get an education as to what's happening in the software and hardware markets. If it weren't for *80*, I would be eternally stuck with a mailing list program that takes 15 hours to sort 600 names, and I'd still be paying $5.95 each for disks. Your reviews of your advertisers products serve to (hopefully) protect your readers from some of the bad buys on the market.

Keep up the good work, guys. I for one think I'm getting my money's worth.

Bill Robinson
Merced, CA 95430

Model III AutoEdit

I have been having difficulty making some of your programs for the Model I work on my Model III. One case in point was the "AutoEdit" program described in the February 1981 issue. Apparently the addressing system in the Model III is different enough to foul up the program. However, I thought your readers might be interested in my solution so that they could use this very useful program in their Model III's.

The location for the "current line pointer" (the period in EDIT.) is correct at 16620. If we increment that value by N with a POKE at 16620, and then try to EDIT, we will either have success or an error due to an undefined line. If we use the ON ERROR GOTO and RESUME statements, the auto-editor will keep going until it finds the next line.

This program listing works perfectly in my Model III. I load it in from tape before entering any code in my machine, and when I'm ready for the auto editor, I simply turn line 4 into a remark statement by inserting a single '. The program then works as described by Mr. Rollins: That is, initializing with a LIST of the line where I want to start, followed by a RUN <enter> each time I want to edit the next line.

I used one as the value of N to be sure that I didn't miss any lines. The only drawback of this is that if there is a large jump in the program lines, it takes a while to find the next line. If the program was written with all line spacing equal to ten, the

Continue to page 25

80 DEBUg

Mem Size. . . 20K

Many readers who built the 4K or 16K memory board in accordance with my article ("Mem Size. . . 20K!") on p. 116 of the Nov. 1980 issue have contacted me concerning loss of data in the added memory block—especially the 16K add-on.

I experienced the same problem when the circuits for both the 4K and 16K blocks were built and under test. Difficulties with the 4K block disappeared when additional filtering and bypassing were added onto the —5V line on the memory board. The —5V supply is very lightly loaded by the memory chip; as a result, this supply line is very susceptible to noise. Two or three 50 or 100 MF electrolytics and a handful of .01 to .1 MF disk sprinkled up and down the —5V distribution line should bypass the noise picked up on this line.

If you connect an oscilloscope to a not properly bypassed —5V line, you will probably see at least one-half volt of noise. Add large and small bypass capacitors until the noise signal voltage is radically decreased.

Bypassing will cure many ailments with the 16K memory board. The major cause of other difficulties is the addition of fast (200 or 250 ns) memory chips external to the keyboard if the chips inside the keyboard are the standard 450 ns speed. If all else fails to settle down operations of the 16K add-on memory board, swap those fast chips with the ones inside the keyboard.

W. R. Stanley
City Rt. 5, 204 Avery Ln.
LaGrange, GA 30240

Auto-Edit Bug

The program AUTO-EDIT 1 (Listing 1) in your February 1981 issue has a bug. The program should produce EDIT mode for consecutive program lines by typing RUN. As written, it skips every other line.

Here is the corrected program (corrections underlined):

```
1 CLP= PEEK(16620)+ PEEK(16621)*256 :X=
17129
2 LN = PEEK(X + 2) + PEEK(X + 3)*256
3 IF LN <= CLP THEN X=PEEK(X) + PEEK
(X + 1)*256 :GOTO 2
4 POKE 16620,PEEK(X + 2) :POKE 16621,PEEK
(X + 3)
5 EDIT
```

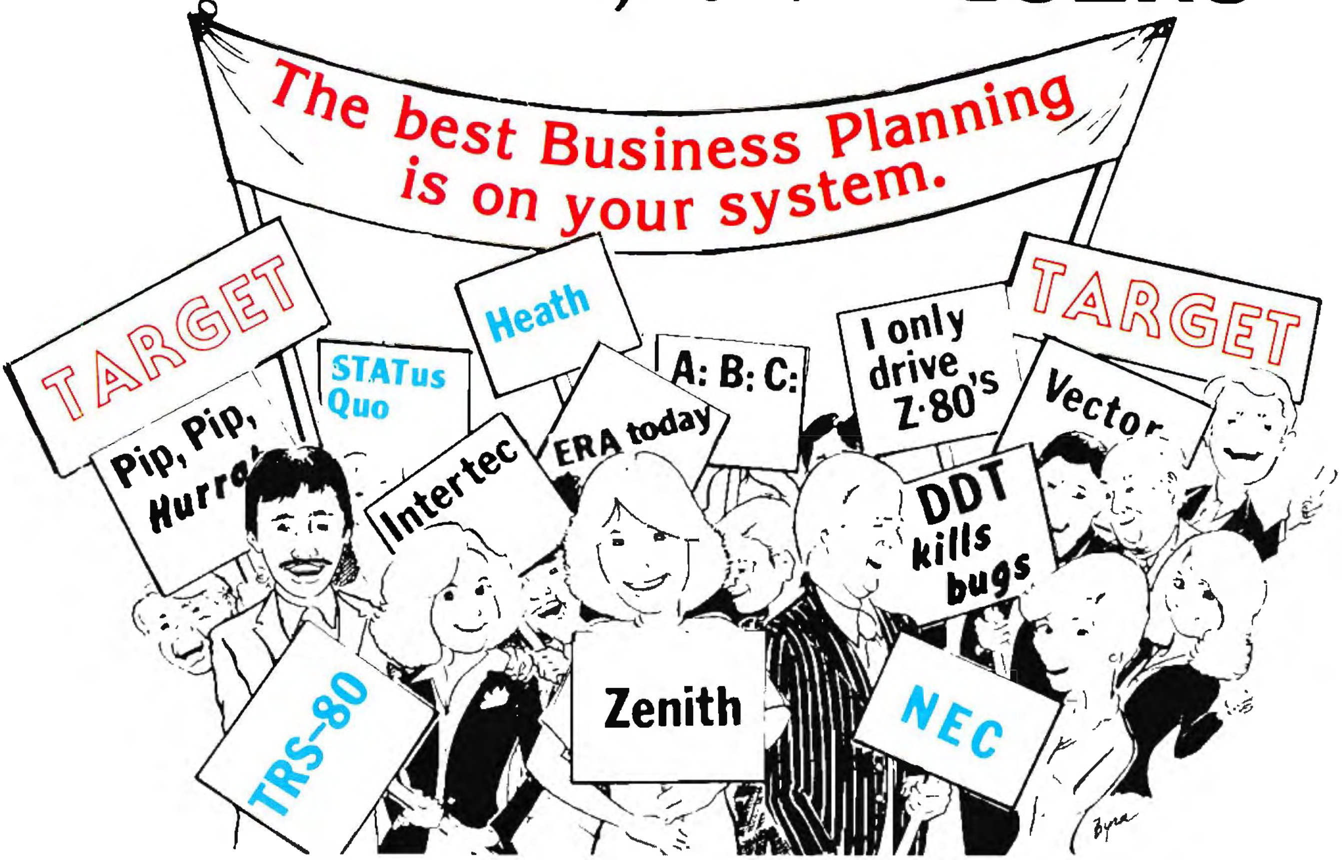
Dan Rollins
370 N. Cerritos, #15A
Azusa, CA 91720
Micronet ID# 70250,631

Triple Play Bug

In the article "Triple Play For T-BUG" in your October issue, Mr. W. H. Johnson describes how to move T-BUG to a new location in memory. I have tried to use the program he describes, and after spending some time trying to get it to work, I can say confidently that there is a mistake in the data. The data block in Table 1 shows that location 2B3B should be filled with a 73. That location should in fact be a 74 to make the resulting new T-BUG run correctly.

Allen Davidson
3502 Crystal lake Ave.
Crystal Lake, IL 60014

STAND UP, CP/M USERS



For the past few months you have patiently endured the indignity of watching your friends show off their flashy visible number cruncher on their game-playing computer and longed for something as slick.

You seriously considered buying *their* computer, but you just couldn't give up the benefits of the CP/M operating system.

Well, say hello to TARGET™, the best business planning system available on micros today (and some big computers besides).

Talk about a blank ledger sheet or columnar tabulator! TARGET™ is much more than that. Sure, just like their product, you can fill in the rows and columns as you see fit and the numbers will whiz by you on the screen. But, we let you create your report in English, not in some combination of reverse Polish notation and algebraic matrix languages.

Our TARGET on *Your* System. . .

```
LINE 1 SALES = 100 200 300 400
LINE 2 EXP   = GROW 50 BY 15%
LINE 3 NET   = SALES - EXP
```

Their Product on *Their* System . . .

```
SALES 100 200 300 400
EXP 50  +B2*1.15 +C2*1 +D2*1.15
NET  +B1-B2 +C1-C2 +D1-D2 +E1-E2
```

At least, that is what their product might look like if you could see all of your data and calculation rules at the same time, *which you can't.* If you think that it is an easy approach for debugging, guess again.

TARGET™ displays a full screen of results or data and calculation rules at your command. And, it runs on your system. How much more could you ask?

Lots! And TARGET™ delivers. We give you the ability to:

- Obtain hard copy printouts of *both* the results *and* your set of data and calculation rules used to create those results.
- Write the results of data and calculation rules to the disk as formatted files for word processing.
- Use full conditionals (IF . . . THEN . . . ELSE) in your set of rules and stack conditionals within conditionals.
- Build powerful models with commands such as GROW, MIN, MAX, AVE, CUM, GREATER, LESSER, SUM.
- Edit lines with ease.
- Enter rules and data in any order you wish. TARGET™ will automatically sort rows in ascending order *and* process your model correctly even if LINE 33 SALARIES needs to be calculated before LINE 3 EXPENSES.

There is so much more about the TARGET™ Planner that will impress you. Our manual explains not just what a command or function is, but how to use the functions in everyday business situations. Our manual and our newsletter give you illustrative examples of business planning problems and how TARGET™ can help you solve those problems. For example, we will show you how to quickly determine the payback period for a proposed project.

TARGET™ is a compiled system running under CP/M. There is no need to buy BASIC or FORTRAN or any other system software. And the price is *only $195* for the TARGET™ Planner, our basic system.

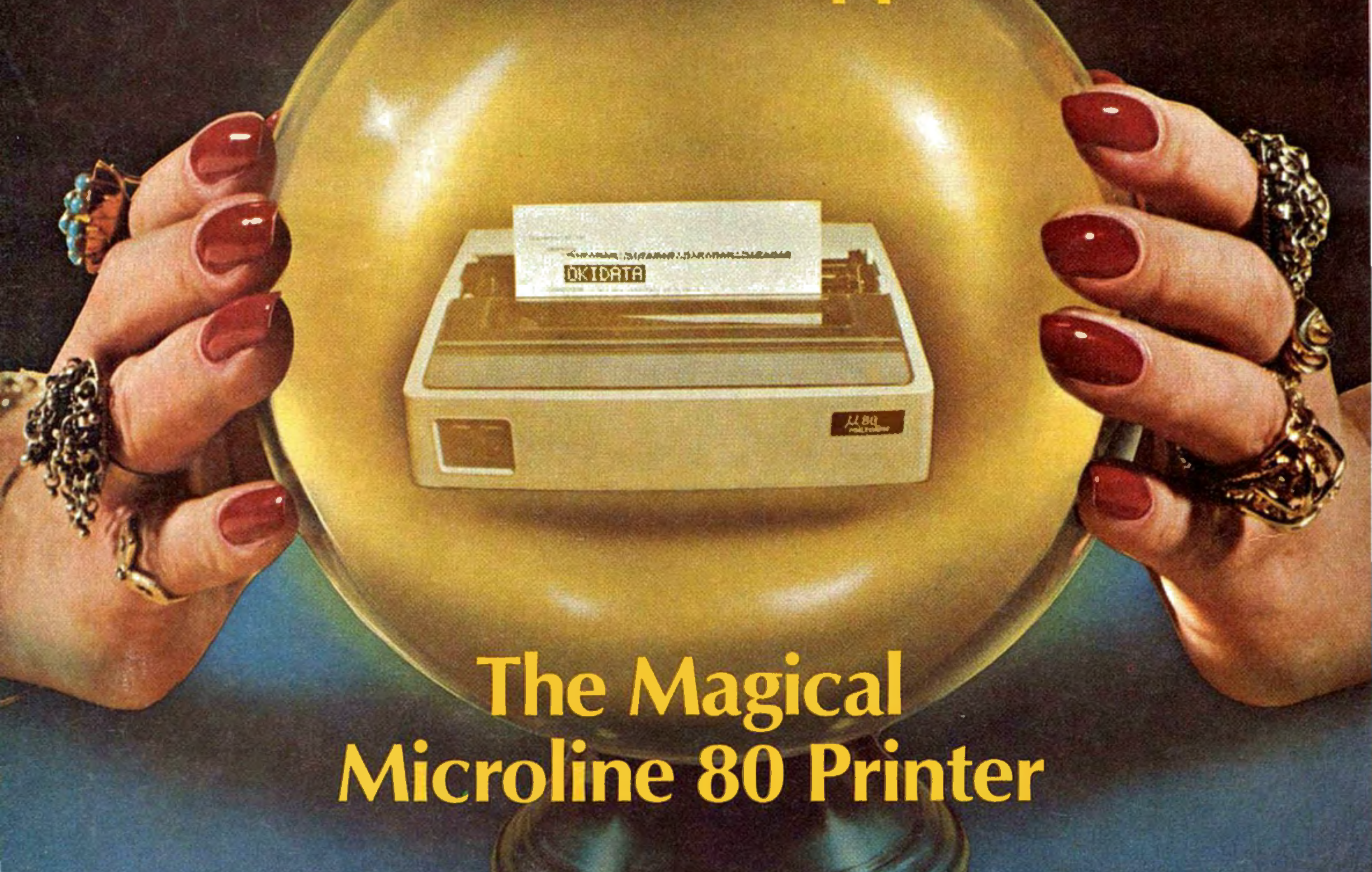
Stand up and be proud, CP/M users. The best business planning system runs on the best operating system on *your* computer.

Advanced Management Strategies, Inc.
1935 Cliff Valley Way, N E , Suite 200
Atlanta, Georgia 30329
404 634-9535 ↙260

P.S. When all of your friends start drooling over *your* product, tell them to cheer up. With Microsoft's great SoftCard and 16-K memory board, your friend's Apple can move up to your operating system and run TARGET™.

CP/M is a registered trademark of Digital Research
SoftCard is a trademark of Microsoft.
TRS-80 is a trademark of Tandy Corporation

What has nine lives, three forms, multiple faces and a price tag that almost disappears?



The Magical Microline 80 Printer

It's magic! Well, almost. The Microline 80 will run all day at 80 cps with no duty cycle limitations. The head is warranted for 200,000,000 characters. That translates to over nine years on your TRS-80,™ APPLE® or other small computer.

Want to change forms? The magical Microline 80 is three printers disguised as one. There is a whisper-quiet rubber platen for cut sheets and roll paper, pins on nine inch centers for pin feed stock and optional snap-on tractors that adjust to suit all your other forms. The Microline 80 also saves paper by letting you tear off as close as one inch from the last print line.

Want to change your image? The magical Microline 80 really does tricks. It prints upper and lower case, condensed and double width characters *and* block graphics for charts, graphs and diagrams.

The Microline 80 is not a toy. With two motors, a rugged cast aluminum base and a head you never have to throw away, the Microline 80 is built to handle the most demanding business applications.

Which brings us to the biggest magic of all, the price tag, the one that almost disappears. If we're not the lowest, we are so close that it doesn't matter. There are stocking Microline distributors throughout the country. Call or write today for the name of the one near you and the price of the Magical Microline 80.

OKIDATA

Okidata Corporation, 111 Gaither Drive, Mount Laurel, New Jersey 08054 609-235-2600

Okidata is a subsidiary of Oki Electric Industry Company, Ltd.

245

value in line six can be changed to CLP = CLP + 10 to speed things up. Alternatively, a break and LIST can identify the next segment and the CLP can be reinitialized by LISTing the first line.

I like my Model III very much, but I hope somebody writes an article which does a little better job of describing how the differences affect the use of Model I hardware and software. Right now I'm afraid to buy a lot of the attractive materials available for Model I for fear that I won't be able to make them work.

Richard H. Reitz
Midland, MI 48640

```
4 GOTO10
5 ON ERROR GOTO9:CLP = PEEK(16620) + PEEK
  (16621)*256
6 CLP = CLP + 1:MSB = INT(CLP/256):LSB = CLP - MSB
  *256
7 POKE 16620,LSB:POKE 16621,MSB
8 EDIT.
9 RESUME 6
```

Recommends MicroCompatible

A word of praise for MicroCompatible of Scaly Mountain, NC, for their Model I upgrade (advertised on page 206 of the February 1981 issue).

Before spending $200 for the device, I called MicroCompatible and finally ordered the upgrade kit. I was amazed to find that it arrived at my home within four days of my telephone order.

I immediately plugged it in and it played like a charm from the very first touch of a button.

The service is not only excellent, but the product appears to be very well made and performed in all respects as advertised. This is somewhat of a novelty to me after having purchased a lot of worthless hardware and software trying to weed out the good from the bad.

One thing MicroCompatible doesn't tell you is that they also provide a very convenient reset button in an accessible location. There is also a capability of high speed disk operation if you have a disk operating system that will support this.

My hat is off to a good product and I recommend it to anyone.

Larry R. Moore, P.E.
Moore-Bingham & Associates
Cedar Rapids, IA 52401

Micro Development

Our firm, Suncoast Micro-Systems, is a private, not-for-profit corporation with a primary objective to assist in the development and placement of microcomputer systems within community-based, public service and charitable organizations. In our view, development of microcomputer systems within this market sector has been impeded by existing funding regulations and a lack of familiarity with computers among governmental funders and local agency personnel alike. We intend to assist in bridging that gap by providing information and assistance to local public service agencies that have a preliminary interest in microcomputer applications.

We seek contact and collaboration with other non-profit groups that are trying to catalyze appropriate software and training materials for community based agencies.

Joseph J. Spatafora
Senior Associate
Suncoast Micro-Systems
1870 Sailfish Rd.
St. Petersburg, FL 33707

Zoom! Friend

We're all too familiar with companies whose interest ends with the sale. I wish to point out an exception.

After the purchase and use of a Zoom 3.6 from Zoom!, Nashua, N.H., I experienced a minor difficulty with the software. I placed a phone call to Martin Tobias of Zoom! and in the week which followed received four long distance calls from him.

The problem was rectified and the product exceeded my expectations. We wish to thank Mr. Tobias for impeccable service and a useful tool.

John Parsons
Nutrihealth Information Systems
Waxahachie, TX 75165

Short Pencil

For several months I have been using the cassette-based Electric Pencil for preparing manuscripts and letters using single sheets of 81/2 × 11 inch paper. Since the Electric Pencil does not automatically pause at the end of a page, I have to watch the printer carefully and stop the program manually so I can insert a new sheet of paper.

The tediousness of this task finally got the best of me, and I decided to investigate the program to see if I could modify it so it stops at the end of a page. After a lengthy search through the code, I located the spot at which the program tests for the final line of a page. At this point the program jumps to the page spacing routine. By changing only two bytes, I altered this jump to a return to the main system. Now when the printer reaches the end of a page, it stops, and I can load a fresh sheet of paper at my convenience.

The changes I made are detailed below:

Location	Initial Code	Disassembled	New Code	Disassembled
4F3B	CD	CALL 5283		
4F3C	83			
4F3D	52			
4F3E	35	DEC (HL)		
4F3F	28	JR Z,8	C8	RET Z
4F40	06		00	NOP
4F41	0D	DEC C		

Only bytes 4F3FH and 4F40H are changed, to C8 and 00 respectively. This change can be accomplished by using T-BUG relocated to high memory. After the change is completed, a new system tape may be written from T-BUG. The starting address is 4350H, the ending address is 5374H, and the entry point is 4350H.

This new version of Electric Pencil cannot print continuously, since the alteration forces it to stop at the end of a page. However, if I need continuous printing, I can always use the original program cassette.

Rodney Schreiner
Madison, WI 53715

ONE LINERS

The First One-Liner

From time to time readers have addressed themselves to the problem of finding the file name of System tapes. This has usually resulted in a lengthy program to find the name.

Jon Ledbetter showed me a one-liner that has proven very useful for determining file names for Model I TRS-80's:

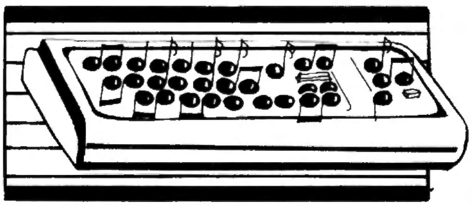
```
10 INPUT #-1,A$:PRINT A$
```

Robert Moran
Box 43
Sheldon, MO 64784

80 REVIEWS

Edited by Pamela Petrakos

"Synthesizer technology has advanced greatly... today we have synthesizers that are a fraction of the size and cost of the early models, yet with greater capabilities."



Music Master
Instant Software
Peterborough, NH
$7.95

Keynote
Hayden Book Co., Inc.
Rochelle Park, NJ
$9.95

by Jim Heid

The use of computer-like circuitry for music generation is not a new concept. The first electronic synthesizers were developed over twenty years ago. They were massive devices that looked like a cross between an organ and a telephone switchboard.

Synthesizer technology has advanced greatly since then, and today we have synthesizers that are a fraction of the size and cost of the early models, yet with greater capabilities.

Microcomputers can also be used to generate musical tones, using the same theory as large synthesizers. This article will describe this theory, and review two music oriented software packages for the TRS-80: Instant Software's Music Master, and Hayden's Keynote.

Sound Information

Sound is made up of variances in air pressure called sound waves. The more times per second these waves pulse, the higher the sound's pitch (Fig. 1). The greater the fluctuation of the waves, the louder the sound's volume (Fig. 2). The more variations there are in the waves, the harsher the sound's tone color (Fig. 3). Instruments like drums and cymbals produce harsh, erratic sound waves; instruments such as violins and flutes produce purer waveforms.

Microphones translate variances in air pressure into variances in electric current, very much like the human ear does. The minute current supplied by the microphone is fed into an amplifier to boost its strength (or amplitude), and then sent to a speaker which translates this current back into variances in air pressure.

By measuring a sound's waveform—its pitch, volume, and purity—at fixed intervals, it is possible to create a data table containing all of the sound's characteristics. The shorter the interval between measurements, the more accurate the information about the sound will be. Then, by instructing an electronic circuit to recreate the voltage variations at the same intensities and intervals as they occurred when they were measured, the sound can be accurately imitated, or synthesized.

The accuracy of the data and the extent to which it is recreated determine a synthesized sound's quality. The more accurately we duplicate what occurred when the sound was made, the more the synthesized sound will resemble the original.

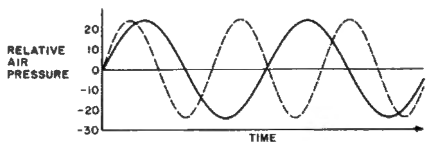


Fig. 1. The dotted line is a high frequency sound; the solid, a low frequency sound.

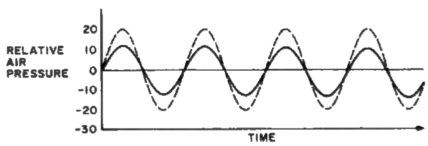


Fig. 2. The dotted line represents a louder sound; the solid, a softer sound. (The sounds are of the same frequency in this diagram.)

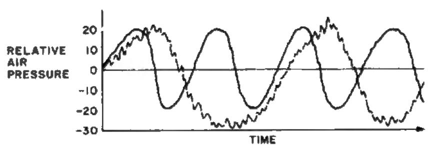


Fig. 3. The dotted line represents a hard sound, such as a drum or cymbol. The smoother line represents a purer sound, such as a violin or a piano.

The most modern electronic synthesizers use a digital representation of the soundwave. This representation is a highly accurate computer rendering of the original sound, and is often recorded on floppy disks and loaded into memory as needed. This method, while very complex and expensive, is the most accurate and often it takes an expert to discern between the original and the synthesized version.

Although precise sound synthesis requires more circuitry than is present in a microcomputer, the theory can be demonstrated with good results using software and, in some cases, a small hardware addition.

Music Master

The Music Master software package consists of a Level II cassette tape containing four programs: Micro Organ, Kaleidopy, Keymania, and Composer. All will run in 4K of memory, and require that the computer's cassette output cord be attached to an external amplifier.

The first program, Micro Organ, is a machine language program that turns the top two key rows of the computer into an organ keyboard on which one note or voice can be played. The user has a choice of three tone colors, which Instant Software calls piano, organ, and harpsichord. These names are, however, just labels, because the tone of each does not accurately represent its namesake.

Another drawback of this program is the lack of a keyboard buffer. The user must release the first note before the second note can be played. Failing to do this will cause the first note to sustain, even though the user may be pressing a different key. The user must adapt to this by releasing the first note, pausing a split second, then playing the second note. This results in staccatto sounding music.

This program provides a live keyboard, meaning that the music is played as soon as the keys are struck. There is no way to save each musical piece, except to record it with a microphone as it is played.

The second program, Kaleidopy, generates random musical notes, and produces a graphic design on the screen as it plays the notes. It is entertaining for a few minutes, but after that, each piece starts to sound the same.

The third program, Keymania, is more of a game than a music generation program. In Keymania, random notes are played and the players (up to four) are required to replay the notes in the same order that they were heard. The more notes that the player requests, the harder it is to recall each one. This game can be quite challenging and fun.

Composer is the final program in the package, and in my opinion, the least valuable. Like Kaleidopy, this program generates random musical notes, but plays them in user-selectable keys and tempos. It also repeats certain sections of the random musical piece, in order, according to the instructions, to relieve some of the boredom associated with totally random music. But to me, repeating random music at random intervals still gives random music, which is boring.

Music Master is a fair package, ideal for demonstrations and for the person who is interested only in occasional experimentation with computer music. Its price is very reasonable and its quality and documentation are good.

Keynote is Level II machine language/BASIC program which requires a minimum of 16K RAM to operate. The program permits the user to enter a musical piece using a special mini-language in which symbols are used to label a note's pitch and duration. The resulting piece can be edited and saved on cassette if desired.

To use the program, the sound generating machine language program is loaded into a memory area that has been protected using the MEMORY SIZE question. After the program is loaded, the BASIC program is loaded. The BASIC program contains the routines for entering, editing, and saving the musical piece.

The program's synthesizer features a five octave pitch range, three possible tempos (slow, moderate, or fast), and the ability to repeat certain sections of the piece. Keynote also features extensive error-trapping routines that result in smooth program execution.

After the musician has entered his/her song, he/she may choose from a multifunction options list to edit and review the piece, play the piece, or save the piece on tape. The edit option displays the contents of the piece in 40-note increments. The user may choose to view the next 40 notes, to change a certain note(s), or to repeat a certain part of the piece. The play option compiles the notes that have been entered and plays them through the computer's cassette output. The tape save option allows the user to save a piece and replay it at a later time.

This package has some negative points, however. The execution time of the BASIC program is slow, especially in the edit and review mode. The tape SAVEs and LOADs also get quite long. Execution time can make or break a program's effectiveness for frustration-free use, and Keynote is just too slow. The program also supports only one voice at a time, limiting the piece to melody—not enough for serious music programming.

Keynote is, however, an excellent program for the newcomer to computer music who would like to try it without committing himself to a large expense. The quality of Hayden's software is outstanding, and their documentation is good. For occasional use and for demonstrations, Keynote rates as a good buy.■

The Music Box
Newtech Computer Systems, Inc.
Brooklyn, NY
$149

by Terry Kepner and Erich Whitey

The Music Box claims to be "...a sophisticated facility for the composition and performance of music operating on the TRS-80 Microcomputer." Unfortunately, while it does fulfill this claim, it falls a bit short of actually being a truly versatile tool for the serious music student.

Documentation

The software manual is 71 pages thick, confusing, and incomplete. Written by the author, it falls into the trap in which most author-written manuals trip. Because he knows the software so well, he tends to give incomplete explanations. For example, a diagram is given (see Fig. 1) which, according to the manual "... shows the relationships among the five programs and the way in which files are passed from one to the other." Nothing further is mentioned about it. I am sure he knows how to interpret it, but we didn't.

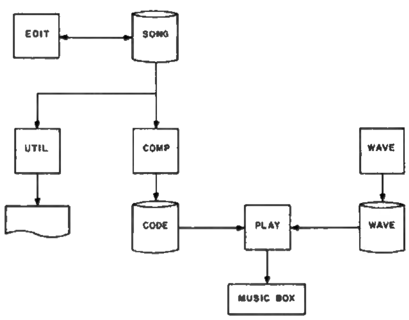


Figure 1.

Also, author-written manuals tend to overlook little details and tricks of use that make it easier to operate. In fact, he spends a good deal of space bragging about all the neat little things you can do with his software instead of giving examples of how to use the routines to achieve desired effects.

The hardware manual, a preface of eight pages added to the software manual, gives instruction on how to hook The Music Box up to your computer. These instructions are, by contrast, impeccably prepared. They not only tell you how to connect it, but also explain the electronics in detail, provide a complete listing of all the parts, include a schematic labeling of all the components, and even give the physical layout of the components on the circuit board!

Statistics

The Music Box's statistics are rather straightforward:

- It plugs into either the keyboard, or the expansion interface.
- It includes all the software needed, as well as all the extra hardware needed, except for a speaker (i.e., it contains its own audio amplifier in addition to its digital-to-analog electronics and power supply).
- It is well built.
- Songs may contain from one to four voices, each voice has an octave range of seven, starting at 27.5 Hz.
- The note range extends from the whole note down to the sixty-fourth note, including 1/5's and 1/7's of each of these.
- Notes may be dotted or double-dotted.
- Measures, repeats, refrains, triplets, and microtones are all possible.
- The wave form used to generate sound is accessible and can be changed at any time, including during the playing of a composition.

The software itself is divided into five programs: Edit, Comp, Wave, Play and Util.

Edit

The Edit program is used to enter music into the memory of the computer, and to change music already composed. The editor is very straightforward, supplying cursor, page, measure-step and home functions. Unfortunately this straight-forward approach is a liability, being a little too clumsy for any real professional, or even a beginning student of music.

For a programmer it is even worse (I've been spoiled by the R/S line editor). Entering a small 22 measure song in three-four time took over three hours, which included interpreting the manual instructions and experimenting. Each note is treated as a different line in the editor. However, all four voices for that note can be placed on the same line. See Fig. 2, a transposition of Bach's Invention #1.

There is one major restriction imposed by the Editor: If more than one voice is selected, all the voices called to play at the same time (i.e., all voices on the same line number) must play for the same duration. For example, when voice number one has an eighth-note duration, voices number two and number three, and number four are all played for an eighth note duration, even if the music calls for a longer or shorter time in these other voices.

The next restriction is not major, but it is very annoying. If for some reason you don't want to save the file just typed into the editor, the program will not let you exit unless you reset your computer, otherwise it will attempt to save your newly modified file (which will lock-up your computer if you're in Level II BASIC, or write the new file on top of your old disk file). In other words, you are often trapped in the editor, or even some sub-editor functions, involuntarily.

Compiler

The Compiler converts the music from the notation used by the editor to the numerical form required by the Play program. Although it doesn't take too much to operate, it is a slow and tedious process to wait through. I have seen (and reviewed) other music systems that compiled their data while in the editing section of the software, and a good deal faster too.

Wave

This program creates and stores new sound waveforms (also called colours, or instruments). These waveforms are used by the Play program as the sound by which notes are generated. For example, a harpsichord-like sound can be generated by specifying the proper waveform. Unfortunately, unless you have had courses in phase angles and relative amplitudes (heavy on the math), the Wave program will mean nothing other than a series of puzzling numbers.

Line	Duration	Voice 1	Voice 2	Line	Duration	Voice 1	Voice 2
	*V 2				*M 0001		
0003	S	R@4	R@3	0022	S	D@5	G@3
0004	S	C@4	R@3	0023	S	G@4	G@3
0005	S	D@4	R@3	0024	S	A@5	G@2
0006	S	E@4	R@3	0025	S	B@5	G@2
0007	S	F@4	R@3	0026	S	C@5	R@2
0008	S	D@4	R@3	0027	S	A@5	R@2
0009	S	E@4	R@3	0028	S	B@5	R@2
0010	S	C@4	R@3	0029	S	G@4	R@2
0011	S	G@4	R@3	0030	S	D@5	R@2
0012	S	G@4	C@3	0031	S	D@5	G@3
0013	S	C@5	D@3	0032	S	G@5	A@4
0014	S	C@5	E@3	0033	S	G@5	B@4
0015	T	C@5	F@3	0034	T	G@5	C@4
0016	T	B@5	F@3	0035	T	F@5	C@4
0017	T	C@5	D@3	0036	T	G@5	A@4
0018	S	B@5	D@3	0037	T	F@5	A@4
0019	S	C@5	E@3	0038	S	G@5	B@4
0020	S	C@5	C@3	0039	S	G@5	G@3

Figure 2.

This is a spot where the manual has failed to supply sufficient information. It tells you how to specify the fundamental and the harmonics, but does not tell you which ones will produce standard effects such as the harpsicord or organ. This is left to you to experimentally discover.

Play

Play brings everything together: the waveform, the music and the player. You actually have three modes of operation. The first is simply play the music, the second is rehearsal, the third is interactive.

Rehearsal lets you use the computer as a musical practice partner. Program your music to play the pieces you want to practice, leaving out those portions you will actually play yourself. Start the song and play your instrument, with the computer accompanying you along the way. At predetermined places you can stop the music, tell the computer where you want to pick up, and then continue, repeating portions as many times as necessary. Interactive mode allows you to assign (while in the Edit program) refrains of music to certain keys on the keyboard. When you play the music, you can force the computer to switch to any given piece by pressing its associated key. This is very, very good for composing avant-garde pieces.

Utility

This is supposed to be a special utility designed to use the RS-232 port. There were absolutely no instructions on how to use it (although the table of contents mentioned a subsection of Util, it wasn't anywhere in the manual), merely a fast list of its functions:

- Obtain a print-out of music in memory
- Receive a song from the RS-232
- Send a song to the RS-232
- Receive a waveform file from the RS-232
- Send a waveform file to the RS-232
- Set the baud rate of the RS-232 from 110 to 96 (this last function is puzzling. I believe the author means "set it anywhere from 110 to 9600"; but then what about stop bits, parity, etc.)

I did manage to get a listing of my song, but there were a few unexpected (and unexplained) stray symbols intermixed among the normal characters.

Overall

The main advantage of the program lies in its technical versatility. Microtones are useful in achieving Eastern musical effects, and along with the wide variety of

NEVER UNDER$OLD.

Thats right. If you can find a lower price in this magazine WE WILL BEAT IT! Period.*

SPECIALS

SPECIAL #1 . . . . $44.50
TRS-80 Disk and Other Mysteries Book, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #3 . . . . . . $149
Apparat NEWDOS/80, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #4 . . . . . . $190
Microsoft Basic Compiler, Box of Verbatim Diskettes, and a Plastic Library Case

SPECIAL #5 . . . . $49.95
Microsoft Basic Decoded and Other Mysteries, Box of Verbatim Diskettes, and a Plastic Library Case

PRINTERS

EPSON MX-80	CALL
MX-80 Graphics ROM	CALL
NEW EPSON MX-80 TF	CALL
Okidata Microline 80	$415
Okidata Microline 82	$625
Okidata Microline 83	$899
IDS Paper Tiger 460G	$1050
Anadex DP-9000,1	$1195
Anadex DP-9500,1	$1295
Centronics 737	$689
NEC Spinwriter	$CALL
NEC w/ Sellum Option	$CALL
Starwriter 25 cps	$1440
Starwriter 45 cps	$1770
Malibu 165	$1995
MPI 88G	$495

TRS-80 HARDWARE

TRS-80's	$CALL
Micropolis 77 track	$399
Percom Doubler	$199
Percom Separator	$27
Orchestra 80	$79
Shugart 40 tr.	$299
Pertec 40 tr.	$299
MPI 40 tr.	$299
16K Memory Kit	$19.95
Cat Modem	$145
D-Cat Modem	$155
Lexicon Modem	$125
Verbatim 525-01	$26.50
Verbatim 8" DD	$44.00
Isolator ISO-2	$49.95

NEW! Double Density Special

A Percom Doubler, Apparat's NEWDOS/80, Double Zap II [to convert the NEWDOS/80 to single and double density], a box of Verbatim Datalife diskettes, and a Plastic Library Case. A $436 Value...for

$359

Send For Our New Free Catalog!

*Include $2.00 shipping and handling on all orders. Visa and Master Card accepted. Never undersold offer applies only to items currently deliverable from other retail vendors at advertised price. All never undersold offers good as supply lasts. Please add $2.00 for all COD orders. Please call for items not listed. We gladly answer any questions on all of our hardware, software, and supply needs. Quanitity discounts available. School purchase orders accepted. Please remember to figure competitors shipping and handling charges when arriving at never undersold price.

POWERFUL SOFTWARE

Racet's New DISCAT $45
This New machine language disk catalog program will help you keep track of your programs in a neat, orderly library. It can catalog up to 1900 programs in each of 9 catagories for a total of 17000 program location records.

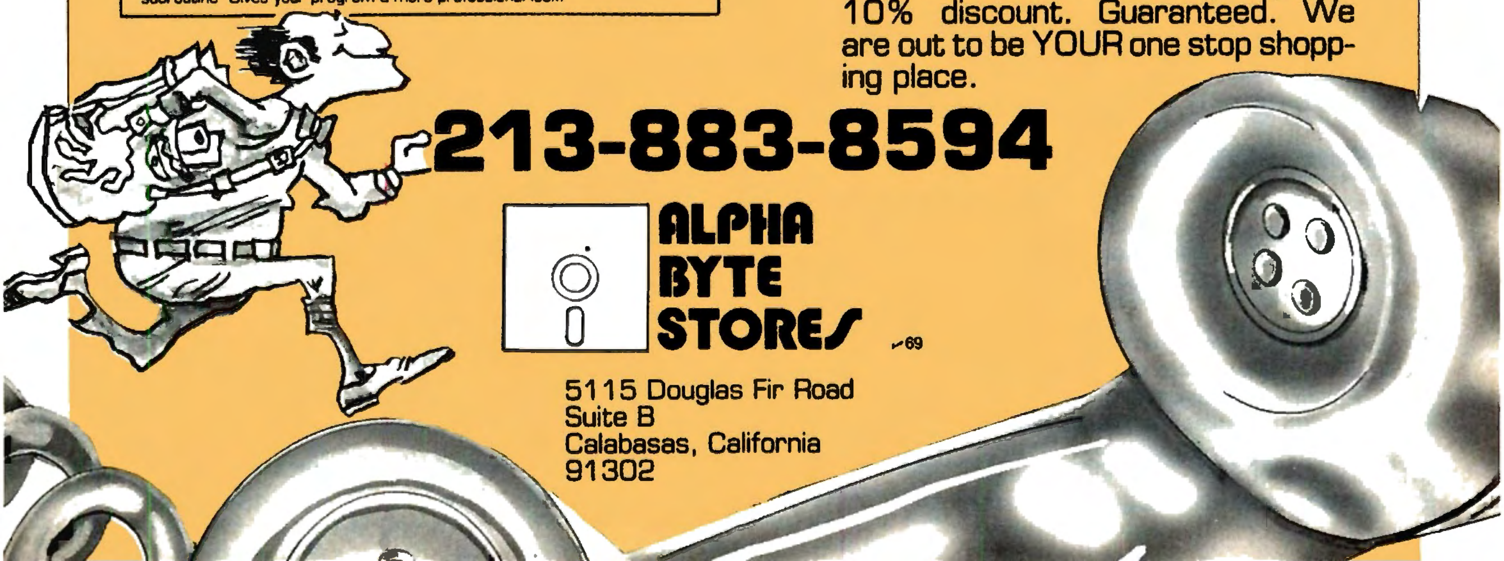
Snapp Spooler $90
Fully automatic print spooler for the model I. Increases your computers efficiency by freeing the computer while printing

TRACKSESS $24
This is the complete disk zap utility, capable of copying standard and non standard formatting. The dynamic editing allows you to change any byte in any sector with amazing ease

SCRINPUT $29
Format your screen displays and input routines with this machine language subroutine Gives your program a more professional look.

INTRODUCING ALPHA BYTE'S 24 HOUR EXPRESS SERVICE..

It's fast and reliable. Just dial our number. If the product is in stock, it's sent on it's way--ALWAYS WITHIN 48 HOURS--If we don't ship it within 48 hours you get a 10% discount. Guaranteed. We are out to be YOUR one stop shopping place.



213-883-8594

ALPHA BYTE STORES

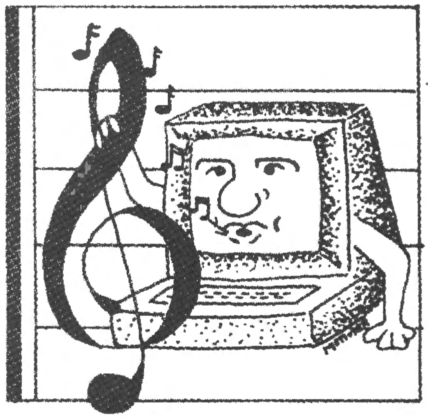
~69

5115 Douglas Fir Road
Suite B
Calabasas, California
91302

possible note lengths, can be extremely valuable in creating original, unconventional compositions. In addition, the rehearsal and interactive modes of the Play program greatly enhance its usefulness in the creation of music.

Its main disadvantage lies in its poorly written manual. The first music student to whom I showed The Music Box gave up after reading only halfway through. She objected strenuously to the repeated use of computer jargon and to the poor organization of the manual. Although she is not a computer hobbyist we have worked together on other music packages for the microcomputer and produced rather thorough reviews, so she is not unqualified to evaluate this type of package. Consequently, I must warn the non-computerist, this package is not easy to learn.

Yet, if you are computer music enthusiast and you want to experiment with unusual sounds and special effects, then you might want to try this package. ■



Orchestra-80
Jon Bokelman
Software Affair
Santa Clara, CA
$79.95

by Jim Held

Sometimes a product is marketed that is of such good quality and value that it soon becomes the standard in its class.

Orchestra-80 is this kind of program. This hardware/software combination for computer music generation is so versatile, so easy to use, and so reasonably priced that it deserves to become the standard in its class.

The hardware half of Orchestra-80 consists of a one and a half by two inch printed circuit board that attaches to the TRS-80's keyboard edge connector, or to the expansion interface's bus extension. It is then connected to an external amplifier, using a standard RCA type cord. The circuit board consists of an IC, a capacitor, and a dozen resistors, making up a digital to analog converter that converts the binary data coming out of the computer into a high quality audio signal. The board requires no additional power supply, and can be left in place when not in use. It must, of course, be detached in order to use the edge connector for any other peripheral.

"The compiler will accept music written in any key, any time signature, and any note value from whole to sixty-fourth notes."

The Orchestra-80 software, a 16K machine language program, contains:

- A Digital Synthesizer
- A Music Language Compiler
- A Text Editor
- A File Management System
- An Initialization Routine.

The synthesizer features a six-octave range and four simultaneous voices (the ability to play up to four notes at the same time). Although the synthesizer voices are preprogrammed to simulate a trumpet, an oboe, a clarinet, and an organ, the voices can easily be re-programmed using the Initialization Routine.

Orchestra-80 uses its own language to input musical pieces. In this language, every note within the system's range is given number or letter, along with a few other codes signifying key, tempo, and other codes signifying key, tempo, and voice, a musical piece can be entered very quickly and easily. The language is extremely simple, yet has all the capabilities needed for serious computer music.

The compiler will accept music written in any key, any time signature, and any note value from whole to sixty-fourth notes. Notes may be single, double, or triple-dotted and/or played as triplets. Accidentals, staccatto, and pizzicatto notes are available. Also provided are two forms of articulation and the provision for repeats, second endings (with or without retard), and modulation.

One of this program's best features is its text editor. The Text Editor provides a fourteen-line text display area and a two line status display area. The status lines are located at the top of the screen, and the file scrolls up and down below them. Fourteen lines of text are viewable at one time, and a blinking cursor can be positioned anywhere within the file by using the arrow keys on the keyboard. The keyboard is fully debounced and all keys will repeat when held down. Additional functions include insert or delete character, insert or delete line, and a global character string search. The contents of a file can be listed on a line printer, and the computer will not lock up if a printer is not attached or not ready.

The File Management System is an equally impressive feature of Orchestra-80. This system provides for orderly storage and retrieval of user files on tape or disk. All files are named, and the disk version adds the extension "/ORC" to the filespec. By typing D, the program will display a directory of all files on a given disk with the /ORC extension. The disk and tape files are compressed when written, allowing fast retrieval and reduced disk/tape space requirements. A special command lets the user enter the names of as many files as he would like the system to perform. The computer will then call the files to the screen and play them successively.

The Initialization Routine allows the user to alter the tone colors of one or more of the four registers, to achieve special effects or special dynamics. The altered program can then be saved on tape or disk with its own file name, eliminating the need to go through the initialization dialogue in the future. The disk version adds the extension "/CMD" to the user-selected filespec.

Orchestra-80 is also extremely well documented. The package comes with a 39 page typeset manual—a refreshing change from the typewriter and mimeograph manuals so many companies are providing. The manual gives full instructions on set-up and use of the Orchestra-80 system, including an important section for non-musicians on reading sheet music.

Software Affair's Orchestra-80 represents a best buy in computer music systems. Its price is less than half that of comparable systems, its documentation is thorough, and its operation is bug free and enjoyable.■

FLOPYCAT/BAS
DISK NAME/BAS
Marvin Plunkett
Microcomputer System Consultant
Roseburg, ORE
$30.00
$15.00
$40.00 (for both)

by G. Michael Vose

Anyone who has struggled through three or four boxes of floppy disks trying to spark a memory as to where you put that great program you wrote, will be happy to learn about this new disk library program from Marvin Plunkett.

The program has been pegged with the rather inelegant moniker, FLOPYCAT/BAS, but the inelegance stops there. Everything else about this program is indeed elegant.

First Rate Presentation

First of all, it is fast. Using machine language subroutines relocatable for sorting and other high level tasks, this BASIC program works as fast as any high quality word processor. These subroutines are tucked into REMark statements to make the program easy to use while saving disk and memory space.

Second, FLOPYCAT/BAS is easy to use. The screen presentation is first rate, all prompts are well laid out and easy on the eye. The screen is never cluttered. Marv Plunkett understands that it is often the little things, like attention to detail, that make the difference in a quality product.

Most importantly, FLOPYCAT/BAS is versatile. You'll need a 32K minimum system to use this software and you'll be able to store 429 disk file names in that 32K. If you've got 48K, you can store up to 1084 file names. You'll be able to use manual input or let the program read the files automatically. You can read the entire library on screen, at any time, or you can print out the entire list. You can change file and disk names and delete files, as necessary. There are search routines to help you find a specific file and you can sort your disk library by disk name or file name.

The features of FLOPYCAT/BAS that I especially admired include:

- As it reads a disk into the library, each file name is flashed briefly on the screen.
- As each disk is read into the library, the used and vacant library space is listed at the bottom of the screen; you'll always know how much more data you can tuck away into that particular library file.
- When you print a list of your file library, the program lets you input a title for the list so that you can have separate lists for work disks, masters, game disks, Scripsit files, etc.

There is extensive error trapping built into the program. I particularly enjoyed being warned, as I was trying to exit to BASIC, that the file had been changed but not written to disk. I slapped my knee and chuckled at my limited memory space and wrote that file to the disk, pronto!

There are several disk directory programs on the market now. All are reasonably fast and efficient. Some hold more data than FLOPYCAT/BAS. But I give FLOPYCAT/BAS the edge because of the author's attention to detail, the overall polish of the screen displays and the user prompts. The documentation exhibits these same qualities and never bogs you down with unnecessary details. If you are interested in the details, the program documentation is available from the author for $30.

As a plus, for $10 more, you can purchase a companion program, equally elegant, called DISKNAME/BAS. This BASIC program allows you to change the names of your disks. You can choose whatever name you like (anything's better than TRSDOS 2.3!). For example, you could name your disks Games1, Games2, Education, etc., and then use FLOPYCAT/BAS to store the files using these names. Using these programs, your disks will be much easier to keep track of, and will result in instant organization! ■

BASCOM
BASIC Compiler for Model I
Microsoft
Bellevue, WA
$195

by Bill Sholar

BASCOM, as Microsoft calls their compiler, arrived with a loose-leaf binder containing several hundred pages of documentation, and two five-inch diskettes.

Interpreters are handy, especially in developing a program in BASIC. But interpreters have to perform an interpretation of the BASIC line every time it is encountered. That can be time-consuming, especially in loops.

A compiler, on the other hand, goes through the BASIC source program and generates a machine language version which can be saved on the disk. When that version, called the object program, is executed, there is no need for interpretation. The program executes at machine language speed. Microsoft promised an increase in speed of anywhere from three to 30 times over the interpreter's execution, depending on the program. (They explain that integers work faster, and offer an example. The routine FOR I = 1 TO 10:A(I) = 0:NEXT I will run 30 times faster if the user merely replaces I with 1%.) To test these claims, I ran a statistical program I occasionally use with the interpreter (Radio Shack's Disk BASIC 2.2). It took 23 seconds to complete the computations, as measured by the internal TRS-80 clock. For example, typing:

```
BASCOM filespec = filespec
L80 filespec, filespec-N-E
```

compiled the program. The compiled version should work faster, and it did. It ran in less than a second. That is some improvement!

The manual suggests that only programs which are debugged and are to be used several times should be compiled. It is, after all, a time-consuming process to compile a program, taking from five to ten minutes. But there may be occasions where you will want to compile a program for a one-time run.

To use BASCOM, it is necessary to write a normal program for the TRS-80 Disk BASIC interpreter (or use one you have already written). There are only a very few incompatabilities, which do not seem to be too important. (For example, my line IF Y = 0 THEN RUN had to be changed to IF Y = 0 THEN 10.) Then the program is saved in ASCII format. Two commands will generate, on your disk, a machine language version of your BASIC program. (With a single drive, you will need to swap disks a couple of times.) The process takes about five minutes.

Some Limitations

You need a 48K TRS-80 with at least one disk drive to use BASCOM. Two drives, especially 40 track ones, are better. (Although Microsoft doesn't mention it, you can save time by putting all of the compiler programs and data on a single 40 track data disk, and using a system disk for creating and storing your source and object code. This will save some disk swaps even for dual drive systems.)

At the moment, you can only run programs of about 16K or less in length. Although the compiler itself reads source code from disk (and could, therefore, handle even an 80K program), the linking loader loads the program into memory. The object code and the run time system (this replaces the BASIC in ROM and RAM usually used by your program), requires substantially more memory than the original source code. With memory area already taken up by the loader, little room is left for lengthy programs.

Microsoft has promised, however, to send all registered owners of BASCOM a new linking loader disk which will allow larger programs to be compiled and loaded. They have also promised to send registered owners who request it a copy of their permanent manual as soon as it is off the presses, to replace the original loose-leaf manual.

The only other limitation is the inability of the USR(n) function to pass arguments to machine language subroutines. But there is a fix for that.

A Few Surprises

Included with the BASCOM documentation is a BASIC-80 Reference Manual. BASIC-80, also called BASIC 5.1, or MBASIC, is the BASIC you've probably seen advertised for CP/M and other operating systems. You may have noticed that it costs about what a new disk drive would cost. Well, smile, because purchasers of Microsoft's BASIC Compiler for TRS-80 Model I get BASIC-80 at no extra cost.

The BASIC-80 they provide has all of the features, including the ability to reset the width of your screen or printed display, WHILE...WEND loops, variable names with 40 significant characters, SWAP, and CALL (a fix for the USR(n) problem). However, you don't get the interpreter, meaning that you have to write the source code and then compile it before you can run it. (The Radio Shack Disk BASIC doesn't understand BASIC-80, needless to say.) To compile BASIC-80 instead of Radio Shack's Disk BASIC, all you do is type a one character signal indicating which form of BASIC you are using.

Errors in standard BASIC can be detected by running the program using the interpreter, and then correcting any detected errors. Mistakes in either Disk BASIC or BASIC-80 will show up, when being compiled, with error codes, the line containing the error, and an arrow pointing to the error!

Considering this compiler is the same software that is being sold for Model II and CP/M users for about twice that price, we Model I users are getting a real break! I heartily recommend BASCOM. It's like going from a cassette player to a disk drive again. ■

Kim's Game
General Computer Co.
Woodbridge, VA
$19.95 cassette
$24.95 disk

by Eric Maloney

Put this program, a keg of beer and 20 friends in your living room, and what have you got? The last party of yours that *those* people will ever come to.

Kim's Game, the "mind-expanding game of memory and recall," combines the excitement of math drills with the gaiety of Roget's Thesaurus to provide you and your guests with hours of unrelenting boredom.

In all fairness to General Computer, Kim's Game is far from shoddy. It is well-produced, well-packaged, and comes with a 36-page loose-leaf manual that includes the BASIC source listing. And it will probably do exactly what it promises—improve your memory.

The name is taken from the Rudyard Kipling novel, *Kim*, whose title character plays a memory game in which he must glance at a group of items on a tray then try to recall them. Kim the program substitites words for objects, and gives the player points for each one that he remembers correctly. The words increase in number, length, difficulty and value to make the game more challenging.

But there is a huge difference between remembering physical objects—as Kipling's character does—and remembering words. The former is visually exciting, and offers players far more mnemonic options. Two people will perceive the same object differently, and these perceptions will change dynamically with variations in, for example, the object's environment, its distance from the player, its juxtaposition to other objects and its angle to the eye. In contrast, a word on a computer screen, is dry, static, flat, and lifeless.

The game does have one redeeming feature—it lets players add their own words. This provides a unique avenue of expression for the sexually deviant and socially depraved. Perhaps the game's true value lies in its potential psychotherapeutic applications.

Ultimately, though, Kim's Game quickly becomes Kim's Endurance Test. An improved memory may have its advantages, but this program makes forgetfulness look like an attractive alternative. ■

Astrology
Tandy/Radio Shack
Ft. Worth, TX
16K Level I & II
$29.95 cassette

by Dennis Thurlow

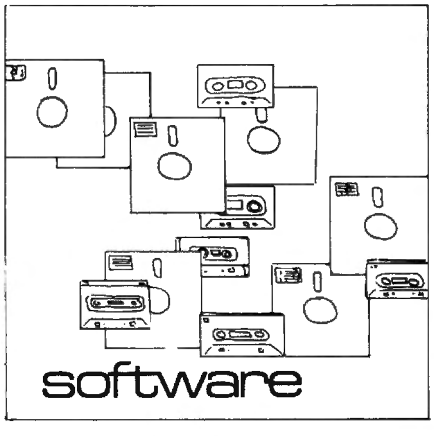
If your only previous experience with astrology is the column in the newspaper or the scales at the dime store, you probably won't be able to use most of the information produced by this addition to the TRS-80's software line. However, if you are a professional charter, you're going to wonder what happened to some very important information.

The non-professional user, who is less likely to have a printer, will find it hard to assimilate the long lists of data. That's why charts were developed. Yet the documentation to this particular program doesn't tell you how to make one. The professional, who is more likely to be able to afford a printer, knows how to make his own charts. So why write a program that only generates a chart on the printer?

When a subject's exact time of birth is not known, the list of houses and ascending sign (which is important to a complete astrological reading) is not given. The professional knows this is because estimating the time is not accurate enough for houses, but Radio Shack gives no explanation to the novice.

If the program was written with the professional in mind, then why doesn't it give heliocentric (sun-centered) aspects along with the geocentric (earth-centered) ones? Where is the median sign? The sidereal (star relative) time? Why are the calculations only accurate for the 20th century?

The program shortchanges both the professional and the beginner. It appears that Radio Shack software developers only got halfway on this one. ■



Cross References Utility
Robert Kilgus
Tandy/Radio Shack
Ft. Worth, TX
$14.95

by Stephen F. Nowak

Some people say that the TRS-80 is the most popular microcomputer because it has such a broad base of software support from so many different sources. There is a version of every game program on the market for the TRS-80, and most business programs are available in many forms for Tandy's favorite child.

Nevertheless, another important attribute of the TRS-80 is its ability to be easily programmed via its powerful BASIC interpreter. Most users buy some programs and write others. At first, programs are short, sweet and simple, but as the user gains familiarity with what BASIC can do, the programs get longer and more complex. We get more confident and freely modify existing programs to "improve" them.

Unfortunately, while writing a new program or improving an old one, we tend to lose track of which variables go where, and how we got to line 2050. TRON helps, but if there's a CLS between where we were and where we're going, it can become frustrating. The problem is further complicated if we reassign variables throughout the program after they've completed FOR-NEXT loops.

Fortunately, there is now a good solution available for this problem. Cross Reference Utility generates lists of references for the following important items:

Line numbers: It lists line numbers that are called by GOSUB and GOTO commands and the lines where those references exist.

Variables: It lists all variables that are defined or used, and each line where that variable appears.

Reserved words: It lists all reserved words and the lines where they appear.

If an item appears more than once in a line, the line number is listed as many times as the item appears. This is especially useful where there are multiple statements in a line with all of the extra spaces squeezed out for memory efficiency.

The program is used in Level II BASIC by loading it with the System command, after which programs can be examined by being read from cassette by the utility program. With a disk system, the utility is called from TRSDOS by its filespec, XREF/CMD, after which BASIC language programs are examined after being read from either disk or cassette.

Once the program is loaded, a series of prompts appear on the screen which must be answered with either a Y or an N:

PRINTER ?
LINE-NUMBERS ?
VARIABLES ?
RESERVED WORDS?
FROM DISK ?

Any combination of cross references may be chosen. The menu is almost idiot proof. If you inadvertently answer Y to the printer query without having a printer connected, the computer will tie up just as if you had entered LPRINT without a printer.

The program prompts you to ready the cassette or to enter the disk filespec. If you specify a non-existent file, it will respond with CAN'T READ! TRY AGAIN. If you attempt to load a file which is not a BASIC language program it will return the statement WRONG TYPE FILE! TRY AGAIN.

If you don't own a printer, the listing can be stopped by pressing the shift and @ keys simultaneously. If you wish to enter a new file, pressing the enter key will return the program to the "From Disk" query. To restart the program press shift and @ simultaneously and then enter.

At the end of a listing, the program allows you to re-examine the same file, load a new file, or exit.

The machine language program is sold on cassette, and comes with a well-written instruction manual, which includes a step by step (or, should I say, key by key) procedure for operating the program. The manual also includes a section with concise directions for transferring the program from cassette to disk using the TAPEDISK utility.

The manual looks larger than the information it holds, since someone decided to print the instructions on only one side of the page. Also included are directions for backing up a diskette, but the directions assume that the user has a two-disk system.

The shortcomings in the documentation are very slight, but are glaring when compared with the high quality of the program itself.

Radio Shack released this program in the fall of 1980, but the program itself displays a 1979 copyright date.

I feel that Tandy needs to catch up on quality in the software department, and this program is a giant leap in the right direction. ■

The Wizard
Richard Taylor
Programs Unlimited
Jericho, NY
$19.95 disk

by Joe Simon

As I was walking past a dealer's booth at the New York Computer Show I heard a pssst... Never being one to pass up a pssst... I immediately went over to check it out.

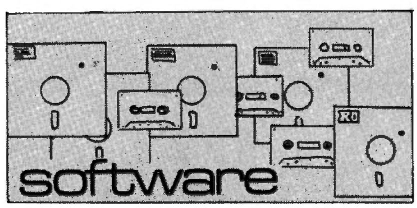
There was this evil looking fellow who asked me if I was tired of computer games that never went anywhere. With an affirmative nod of my head he sat me down and told me to choose a subject.

Doubtful that this program was going to be different from games I have played and shelved in the past, I reluctantly selected a subject. Suddenly, all hell broke loose; the TRS-80 started to make strange noises, the monitor began to flash and I found myself in a a battle against time to answer the questions correctly.

Needless to say, I was there for some time trying to prove I was smarter. It was a draw. I was so involved in the challenge that I didn't even notice the fellow looking over my shoulder, laughing and wringing his hands.

Anyway, I got the program and here is what it was. It's a program called the Wizard and is reminiscent of the coin arcade games that ask you a series of questions on various categories in exchange for points and a free game. It gives you a choice of five subjects: presidents, television, trivia, Star Trek, the movies, and general knowledge. you can play The Wizard alone, against the clock, or against another player.

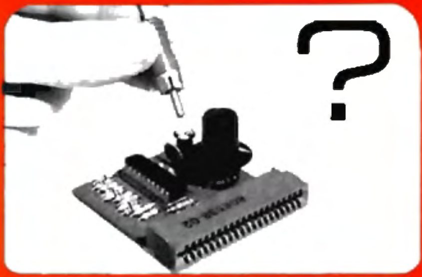
When two people play, you are not only pitted against the clock, you also play against your opponent who can steal your questions or force you to answer questions he rejects. When you answer correctly, points are added to your score. A wrong answer results in points subtracted from your score, and if you steal a question from your opponent and answer it wrong, double points are subtracted from your score.



NEW

Joystick versions of the Fantastic Games by BIG 5 (see page 175) and Software Innovation (see page 49). Available on tape or disk, same price as plain version. One "Stick 80" works with all. Money back guarantee. In stock now.

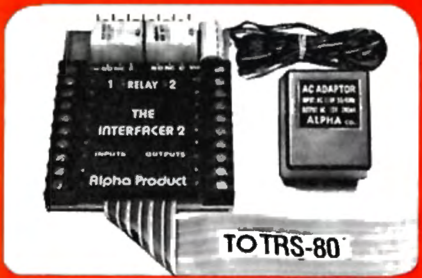
STICK-80 MAKES KEYBOARD OBSOLETE.
Features the famous ATARI Joystick. 8 directions + fire control. Simple instructions to make joystick versions of most action games. Plugs into keyboard or expansion int... Price includes ATARI joystick with ALPHA interface and instructions. FREE "MAGIC ARTIST" program.... $39.95
Super Real Time Action Graphic Sound games for Stick-80 by Software Innovations: ALIEN INVASION, COSMIC INTRUDERS, BREAKOUT Each $9.95
STELLAR ADVENTURE: super action with sound. $14.95
Software authors and distributors: contact us for joystick conversion package for your existing games



MUSIC-80 MUSIC-80 MUSIC-80 MUSIC-80 MUSIC-80
Use existing software or write your own. With this low cost 8 bit digital to analog converter you can synthesize up to 5 music voices. Built-in volume control handy when stereo not near TRS-80. Simply plug the "MUSIC-80" into the keyboard or the E/I screen printer port and connect the output (RCA jack) to any amplifier. The Radio-Shack $12 speaker/amplifier works fine.
Fully assembled and tested, 90 day warranty $39.95

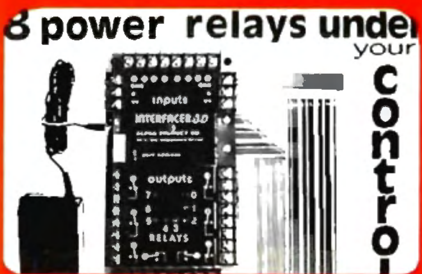
YOU ASKED FOR IT: "EXPANDABUS" X1, X2, X3 AND X4.
CONNECT ALL YOUR TRS-80 DEVICES SIMULTANEOUSLY on the 40 pin TRS-80 bus. Any device that normally plugs into the keyboard edge connector will also plug into the "EXPANDABUS". The "X4" is shown with protective covers (included). The TRS-80 keyboard contains the bus drivers (74LS367) for up to 20 devices, more than you will ever need. Using the E/I, it plugs either between KB and E/I or in the Screen Printer port. Professional quality, gold plated contacts. Computer grade 40 conductor ribbon cable.
X2...$29. X3...$44. X4...$59. X5...$74.
Custom configurations are also available, call us.

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



INTERFACER 2: LOW COST INPUT/OUTPUT MODULE.
Still the best value in sense/control devices. Use it for energy control, burglar alarm, darkroom, selectric drive, model trains, robots, Skinner box...
—8 latched TTL outputs, 2 relays SPDT 2A, 125V, contacts
—8 TTL/CMOS inputs. Input 0 and 1 are optically isolated.
—Neat and compact design, very easy to use:
10 A=INP(0) 'Reads the 8 inputs (if A=0: all inputs are low) 20 OUT 0,X 'Controls the outputs and the relays
Assembled & tested, 90 day warranty. Price includes power supply, cable to KB or E/I, superb user's manual, free phone dialer program: $95. Manual only: $5.

LET THE "CHAIN BREAKER" FREE YOUR MINI-DRIVES.
End the daisy-chain mess once and for all. Fits all mini-drives: Percom, Aerocomp, Shugart, Micropolis, MTI, Vista, Pertec, Siemens, BASF. Easy to install: just remove the drive cover, plug in the "CHAIN BREAKER" and replace the cover. Voila!!!
Now you can change and move your drives around without disassembly. Keep the cover on and keep the dust out. High reliability gold plated contacts, computer grade 34 conductor cable. Tested and guaranteed.
Get one for each drive.....only.................. $13.95



INTERFACER-80: the most powerful Sense/Control module.
- 8 industrial grade relays, single pole double throw isolated contacts: 2 Amp. @ 125 Volts. TTL latched outputs are also accessible to drive external solid state relays.
- 8 convenient LEDs constantly display the relay states. Simple "OUT" commands (in basic) control the 8 relays.
- 8 optically-isolated inputs for easy direct interfacing to external switches, photocells, keypads, sensors, etc. Simple "INP" commands read the status of the 8 inputs. Selectable port address. Clean, compact enclosed design. Assembled, tested, 90 days warranty. Price includes power supply, cable, connector, superb user's manual..... $159.

GREEN SCREEN WARNING

IBM and all the "biggies" are using green screen monitors. Its advantages are now widely advertised. We feel that every TRS-80 user should enjoy the benefits it provides. But **WARNING:** all Green Screens are not created equal. Here is what we found:
- Several are just a flat piece of standard colored Lucite. The green tint was not made for this purpose and is judged by many to be too dark. Increasing the brightness control will result in a fuzzy display.
- Some are simply a piece of thin plastic film taped onto a cardboard frame. The color is satisfactory but the wobbly film gives it a poor appearance.
- One "optical filter" is in fact plain acrylic sheeting.
- False claim: A few pretend to "reduce glare". In fact, their flat and shiny surfaces (both film and Lucite type) ADD their own reflections to the screen.
- A few laughs: One ad claims to "reduce screen contrast". Sorry gentleman but it's just the opposite. One of the Green Screen's major benefits is to increase the contrast between the text and the background.
- Drawbacks: Most are using adhesive strips to fasten their screen to the monitor. This method makes it awkward to remove for necessary periodical cleaning. All (except ours) are flat. Light pens will not work reliably because of the big gap between the screen and the tube.

Many companies have been manufacturing video filters for years. We are not the first (some think they are), but we have done our homework and we think we manufacture the best Green Screen. Here is why:
- It fits right onto the picture tube like a skin because it is the only **CURVED** screen **MOLDED** exactly to the picture tube curvature. It is Cut precisely to cover the exposed area of the picture tube. The fit is such that the static electricity is sufficient to keep it in place! We also include some invisible reusable tape for a more secure fastening.
- The filter material that we use is just right, not too dark nor too light. The result is a really eye pleasing display.

We are so sure that you will never take your Green screen off that we offer an unconditional money-back guaranty: try our Green Screen for 14 days. If for any reason you are not delighted with it, return it for a prompt refund.
A last word: We think that companies, like ours, who are selling mainly by mail should: •list their street address •have a phone number (for questions and orders) •accept CODs, not every one likes to send checks to a PO box •offer the convenience of charging their purchase to major credit cards.
How come we are the only green screen people doing it?
Order your **ALPHA GREEN SCREEN** today...$12.50

✓ 210

ALPHA Product Co.

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

85-71 79 St. WOODHAVEN N.Y. 11421 ☎ Info and order: (212) 296•5916

GAME BREAKTHROUGH:

THE STICK 80

- ☑ FEATURES THE FAMOUS **ATARI JOYSTICK**.
- ☑ WORKS WITH ANY LEVEL 2 AND DISK SYSTEMS.
- ☑ PLUGS DIRECTLY INTO KB. OR E/I (SCREEN PRINTER PORT).
- ☑ INCLUDES SIMPLE, DETAILED INSTRUCTIONS TO MAKE JOYSTICK VERSIONS OF MOST ACTION GAMES.
- ☑ COMPATIBLE WITH **ANY** OTHER TRS-80 ACCESSORIES.
- ☑ FULLY ASSEMBLED AND TESTED. READY TO PLUG IN AND ENJOY.
- ☑ FUN TO EXPERIMENT WITH IN **BASIC**: SIMPLY USE INP(0).
- ☑ 90 DAY PART AND LABOR WARRANTY.

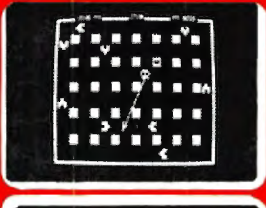
UNCONDITIONAL **MONEY BACK GUARANTEE:** Try your STICK-80 for 14 days. If for any reason you are not delighted with it, return it for a prompt and courteous refund (including shipping and handling). PRICE INCLUDES ATARI JOYSTICK + ALPHA INTERFACE + DETAILED INSTRUCTIONS + DEMO PROGRAM. ONLY $39.95

MODEL 3 OWNERS: WE HAVE A **MODEL 3 TO MODEL 1 ADAPTER**

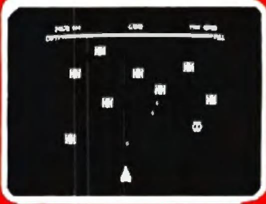
BIG FIVE SOFTWARE SUPER FAST MACHINE LANGUAGE ACTION GAMES

If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate! ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ **SUPER NOVA®**

The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects! ◆◆◆◆◆◆◆◆◆◆◆◆ **GALAXY INVASION®**



Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects! ◆◆◆◆◆◆◆◆◆◆◆◆◆◆ **ATTACK FORCE®**



With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound! ◆◆◆◆◆◆◆◆◆◆◆◆◆ **COSMIC FIGHTER®**

The second **Big Bang** has occurred and the galaxy is full of stray asteroids and meteors. As you look through your space port you see a belt of asteroids drifting across the screen blocking your path to the safety of the space station above. But be careful because meteor showers, exploding suns and invading aliens may strike your ship and send it hurtling back to ground level. How many times can you and your opponent maneuver through those obstacles before time runs out? With sound effects! ◆◆◆◆◆ **METEOR MISSION II®**

SPECIAL

INTRODUCTORY OFFER:

Order the STICK-80 with one or more games and choose one:

•FREE ALPHA GREEN SCREEN
OR
•FREE MAGIC ARTIST PROGRAM

IN ADDITION YOU MAY:
ORDER 2 GAMES AND TAKE 10% OFF
ORDER 3 GAMES AND TAKE 15% OFF

All games are written in machine language and supplied on cassette

EACH GAME IS AVAILABLE IN 2 VERSIONS

A SIDE 1: LEVEL II 16K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 3 $15.95

B SIDE 1: DISK 32K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 1
THE DISK VERSION SAVES THE HIGH SCORES ON DISK $17.95

• ALL THESE NEW JOYSTICK GAMES WILL ALSO WORK USING THE KEYBOARD WITHOUT ANY MODIFICATIONS.

GOOD NEWS: If you already have a non-joystick version of these BIG FIVE SOFTWARE games, send the original tape(s) with your STICK-80 order + $3 per game. We will send you a new Joystick version of your game(s).

ALSO AVAILABLE **DOUBLE-STICK 80**: 2 ATARI JOYSTICKS + ALPHA DUAL INTERFACE + INSTRUCTIONS + DEMO PROGRAM: $59.95

ALPHA Product Co.

~262

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

85-71 79 St. WOODHAVEN N.Y. 11421 phone: (212) 296 5916

Announcing "Hellfire Warrior", a fantastic new Dunjonquest™ computer game… that's really not for everybody: Beginners are likely to be gobbled up in the first room…and there are over 200 rooms on four levels

"Hellfire Warrior." Really not for everybody: newcomers to Dunjonquest should begin with something easier. Here the monsters are deadlier, the labyrinths more difficult, the levels far more challenging…

But for the experienced Dunjonquest game player there are more command options, more potions (131), more magical items (including — at last — magical armor), more special effects, more surprises. And an innkeeper, an armorer, apothecary and magic shops.

In part a sequel to The Temple of Apshai, up until now the greatest of all the Dunjonquest games, Hellfire Warrior can also be played completely on its own.

Now the character you've created, representing the highest level of role-playing to date, can explore the four new lower levels:

Level 5 — "The Lower Reaches of Apshai." With the giant insects and other nasties that swarmed through the upper levels of Apshai. With rooms your hero can get into, but not out of.

Level 6 — "The Labyrinth." The only exit is hidden within the Labyrinth. And man-eating monsters can thwart your hero.

Level 7 — "The Vault of the Dead."…And of the undead — skeletons, ghouls, mummies, specters… invisible ghosts — lurking in the rooms, doors, secret passages, ready to reduce your hero to a pale shadow of himself. Permanently.

Level 8 — "The plains of Hell." In an Underworld of lost souls and shades of dead, of dragons and fiery hounds, of bottomless pits and blasts of hellfire, our hero must rescue the beautiful warrior maiden lying in enchanted sleep within a wall of fire. And bring her past unbelievable dangers and monsters… even Death itself…to sun and air and life itself

Hellfire Warrior. The most exciting game yet from Automated Simulations, the leading producer of computer fantasy games.

Guaranteed: If it's not the most exciting computer game you've played, return it within ten days for a complete refund.

Available on disk for the Apple II and Radio Shack's TRS80, or on cassette for the TRS80 and for the Commodore PET. **The cassette or disk: $39.95.** Boxed with a magnificent instruction manual. Some of the drawings in the manual are reproduced here (in greatly reduced scale).

Hellfire Warrior. From the people who design and make some of the very, very best computer games, games that thinkers play. Including the great role-playing Dunjonquest series — The Temple of Apshai, Datestones of Ryn, Morloc's Tower, and now Hellfire Warrior. And Invasion Orion and Starfleet Orion. And more.

Hellfire Warrior. Like every EPYX computer game from Automated Simulations, it has the unique EPYX triple no-questions-asked warranty.

Hellfire Warrior. Now available on disk or cassette for TRS80, the Apple and the PET. At better computer stores everywhere. If you can't find it at your local friendly dealer, insist he order it for you.

Or you may order directly from Automated Simulations. **If you want to use your MasterCard or VISA, use our toll-free phones:** In the United States: operator 861 **(800) 824-7888;** In California: operator 861 **(800) 852-7777;** In Hawaii and Alaska: operator 861 **(800) 824-7919.** We'll also send you our new catalogue, hot off the press.

Or write Department F2, P.O. Box 4247, 1988 Leghorn St., Mountain View, CA 94040.

Please specify if you wish the **Disk** for TRS80 (32K, TRSDOS) and Apple (48K with Applesoft in ROM or the **Cassette** for the TRS80 (16K, Level II) or the Commodore PET (32K, old or new ROMs). Only $39.95, disk or cassette.

48

EPYX
COMPUTER GAMES THINKERS PLAY

TRIPLE WARRANTY

10 DAY FULL MONEY BACK GUARANTEE: If you don't like the game for any reason whatever, return it intact within 10 days of receipt for a complete refund. No questions asked.

30 DAY DEFECTIVE WARRANTY: Cassette or disk not functioning within 30 days of receipt; return it for exchange. No charge, of course.

FOREVER — LIMITED LIFETIME WARRANTY: No matter what happens to your cassette or disk; the dog chewed it…you left it out in the rain…whatever. No matter **when** it happens. Return the remains to us (with $5.00 to cover all handling and shipping) and we'll send you a brand new cassette or disk.

The Wizard is a program that really doesn't deserve to be called a game because it can also be educational. The real power behind The Wizard is the utility that comes with it allowing you to program in your own questions and answers on any subject you want.

This utility, I might add, is very easy to use. I have used it to program subjects on science and history. It is only limited by your imagination. A possible future addition might be separate category modules that would include questions and answers on a variety of different subjects.

There is one thing about The Wizard that could use improvement. The program takes a noticeably long time to load.

In closing I think the Wizard would be a compliment to any program library and is well worth the asking price. ■

Deluxe Personal Finance
Small Business Systems Group
Westford, MA
$39.95

by Reese Fowler

The Deluxe Personal Finance is an extensive budget program to be used in an individual household. The system is in the form of a mini-general ledger and includes an accounts payable program.

Using the system you can:

- Record 900 check transactions annually.
- Allow nine budget categories per check.
- Maintain an accurate checking account balance.
- Provide summaries of income vs. spending by month.
- Estimate and average monthly expenses.
- Assign expenses to 28 different categories.
- Assign income to three different categories.
- Generate end of year reports for tax purposes.

The package also includes an extensive manual that explains how to use the program. The manual also provides information on how to change the program to meet your specific needs.

The program runs on two disk drives, TRS-80 Model I with 32K of memory.

There are eight modules to the program that are called from a master menu. The data is written for a separate data disk that is inserted only when a request for data is made. This requires changing disks, consequently a second drive helps to expedite the process.

One criticism I have of the program has nothing to do with its operation. The program maintains a good set of records, but only if they are utilized. Often, there is a tendancy in a budget program to use it for awhile and then to stop.

However, if you do use Deluxe Personal Finance, all the data you need at income tax time will be right at your fingertips. ■

Quill Driver: A Text Formatting Program
The Alternate Source
Lansing, MI
Model I 32K
Model II 64K
$39.95 disk

by Dennis Thurlow

A Quill Driver is not a word processor: it is a text formatter. This information, and its ramifications, are not explicitly layed out in its documentation and this may lead you to believe it is actually a word processor. After the table of contents, introduction, and preface, there is one line in the overview stating that text files are prepared using the BASIC editor, EDTASM, or any program that creates unnumbered files of text.

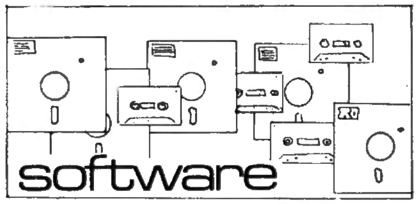
Back to Front

Chapter 2 briefly touches on this fact again, but you will find no examples or discussion of what form the files must have until you reach Appendix B.

I spent two frustrating hours reading through the manual, waiting to find some reference to input. When I reached the end, it occurred to me that the manual would make much more sense read back to front. Accordingly, this review will cover the manual—from back to front.

First we find the registration form. I suggest you send this in because I did find one bug in the system and there's no way they can inform you of the fix unless they know who you are.

Next is Appendix D, which tells us about a machine language driver that debounces the keyboard; keeps LPRINT from hanging up your machine; LPRINTs compression codes as proper tabs; gives you a repeat key function; allows several lowercase options; and sends line feeds and control characters when and where they are appropriate. Loading and use of the driver on a variety of systems is very clearly explained.



Dawning of the Light

Appendix C explains how to patch different versions of EDTASM to accept lowercase input. Why would you want to patch EDTASM? The light suddenly dawns! You can use EDTASM to make text files!

Appendix B shows the format for those files using two clear examples that also demonstrate most of the available printing features. On my disk, the EDTASM files differed slightly from the example, but this didn't cause any problems.

In Appendix B you learn how to imbed the codes to the formatter. Appendix A lists the codes, gives a brief description of each function, and the location of a complete description found in the text of the manual.

The last chapter contains the operating instructions for the program. It is easy to deduce from the operating instructions that text prepared with the BASIC editor can be used, as can files written to disk with no line numbers.

The program, which is written in BASIC, performs all the standard word processing functions when printing, is amazingly versatile and works perfectly with files prepared in the methods described in Appendix A and B. It had some trouble keeping track of the .AL counters in EDTASM prepared text. Normally, the .AL code allows you to switch back and forth between two files of text opened simultaneously. This is great for typing form letters while adding a different heading or personalized comments on each.

Once you figure out how to use the Quill Driver you will be delighted at how well it works, but please, sdrawkcab noitatnemucod eht daer! ■

"The program, which is written in BASIC, performs all the standard word processing functions when printing . . ."

THE ASSEMBLY LINE

by William Barden, Jr.

"...if the machine language output is relocatable, it can be embedded in BASIC...in dummy strings, arrays and other formats."

In assembly language as in real estate, it's sometimes better to remain where you are and not move around. Sure, I know that Zilog literature touts the Z-80 as producing relocatable code with the JR and DJNZ instructions—but how hard is it to make *all* code relocatable? About as easy as it is to sell your "3 BR, FDR, view, near Radio Shack store" home with the sagging rain gutters and cutworm-ridden dichondra...

You know what's coming next: I'm going to gripe and moan about how hard it is to use the Z-80 for relocatable code! If you can tolerate the lamentations, we'll look at code which will provide 50 percent more disk or memory capacity for text data, to illustrate the problems of relocatability. We'll also review some books and, for you puzzle buffs, we'll have the Fourth Great Assembly Line Programming Contest.

Why Relocatable Code?

There are many reasons assembly language code should be relocatable. The first to come to mind has been discussed many times in this column—if the machine language output is relocatable, it can be embedded in BASIC programs in the form of dummy strings, arrays, and other formats. This way machine code can be part of the BASIC program and loaded as one module.

A second reason for relocatable code is to make the object module free to move anywhere in memory without having to reassemble. Most of us have had cassette versions of programs that were not usable in DOS because they overlaid portions of the DOS. Finding the starting and ending addresses of the program is easy enough, but how to relocate the code to higher memory so it could be stored and read from disk is a stumper. In most cases the source code is proprietary. Fully relocatable code would avoid this problem.

Another reason: With a number of relocatable object modules in a library, it would be a simple matter to use them together to create a larger applications program without having to worry about reassembling. The disk Editor/Assembler will automatically relocate object code to any area of memory, but why not work directly with a module instead of a sophisticated loader? It might even be possible to emulate the actions of a large system by allocating memory dynamically for a number of separate tasks. The tasks could be moved without having to adjust the code for relocation; possibly even some multiprogramming could be done—running several programs concurrently in memory.

"In assembly language as in real estate, it's sometimes better... not to move around."

A Base 40 Converter

According to Zilog literature, it's a relatively simply matter to spew out relocatable code. Let's take a look at a typical program and see what's involved in making it completely relocatable. The program I chose to write for this exercise is a Base 40 converter.

The Base 40 converter has been around for a long time. The idea is this: If we work with a subset of 40 characters—the letters A through Z, the digits 0 through 9, and four special characters—it's possible to compress what would be three ASCII bytes into two bytes! This means we could increase our diskette or memory capacity by 50 percent as long as we are willing to use only the 40 characters.

Why 40 characters? If each character is assigned a code 0—39, a 16-bit register pair or pair of memory locations can hold three digits of a base 40 number. If codes 0—9 represent the digits 0—9, codes 10—35 represent the letters A—Z, and codes 36, 37, 38, and 39 represent four special characters, the base 40 number 29, 27, 28 would represent "TRS". The base 40 form would be 29*48**2 + 27*40**1 + 28*40**0, where * represents the multiplication operator and ** represents exponentiation. This is the same as 29*1600 + 27*40 +27*1 = 47507, which can be held in 16 bits.

The largest number would be 39*1600 + 39*40 + 39*1, or 63,999, which could be held in 16 bits with no problem. To answer your next question—no, base 41 would have a maximum value of 68,920, too large for 16 bits!

The program is shown in Listing 1. It consists of two parts, Encode, that converts three ASCII characters into a two-byte base 40 number, and Decode, that reconverts a two-byte base 40 number into three ASCII characters.

Enter Encode from BASIC with the usual Level II, Level III, or Disk BASIC call. The CALL 0A7FH puts the argument into HL. The argument points to the three characters to be converted. On return, the two-byte result is put into the first and second bytes. If any of the characters is not one of the 40 characters, a zero is used as the code.

Entry to Decode is made with the argument pointing to the two bytes of a base 40 number. On return, the three character result is put into the first, second, and third bytes of the string.

Listing 2 shows a test of Encode/Decode from Disk BASIC. An input string of three characters is converted to two bytes, the two bytes are displayed, the two bytes are reconverted, and the result is again displayed. USR0 is for Encode and USR1 is for Decode.

Program Listing 1 is not relocatable. If it were moved from the 7F00H area to BF00H, for example, it would bomb. The CALL 0A7FH is relocatable. It would remain the same, as a call must always be made to pick up the argument from the BASIC interpreter. Everything else is relocatable down to CALL XLATE, which involves an address within the Encode/Decode code, and must be changed if the program is relocated.

Continuing down the listing, we find no more non-relocatable CALLs or JPs; the

JRs are always relocatable, as they are referenced to the contents of the program counter which changes according to the instruction being executed. The displacement in the second byte of the JR is fixed, no matter where the Encode/Decode set of code is moved.

There are four other non-relocatable instructions: The three LD IX instructions that pick up TABLE – 1 or TABLE + 39 and the LD DM TABLE are not relocatable. The immediate address in bytes one and two of these instructions change based on where TABLE is located in memory. In all, one CALL and four immediate LDs must be made relocatable before Encode/Decode can be moved.

Possible Relocation Schemes

Look at the crib sheet of the Z-80 instructions. There is *no* way to read the contents of the program counter that doesn't involve a relocatable instruction! There are some other approaches, however, that might be used.

We could have the calling routine pass us the base argument of Encode. If this code is in a dummy string or array, presumably we could always find the address in BASIC by a VARPTR. This address could be passed as a parameter to the routine itself. If the routine were at A$, for example, the following code would pass the base address to the routine:

```
100 B=VARPTR(A$)
110 B=PEEK(B+1)+PEEK(B+2)*256:IF B>32767
THEN B=B-65536
120 DEFUSR0=B
130 A=USR0(B)
```

Another way we could pass the base address to the routine is by executing a CALL to some fixed location in memory. If the CALL address were permanently at FF00H, CALL 0FF00H would be relocatable. The code at FF00H could then get the return address from the top of the stack and return it in a register pair. Something like this would suffice:

```
POP HL
JP (HL)
```

This code puts the return address into HL from the top of the stack, then jumps to the return address using HL. The return address in HL is not destroyed in the return, as it is in RET. If we have the address of the instruction following CALL 0FF00H, we can easily compute any address in the program block.

The two instruction sequence above is in ROM at location 0BH. As ROM never changes, 133 revisions notwithstanding, we can use the POP HL, JP (HL) as a "where am I" function from any assembly language program.

Adding Bias to a Base Address

Calling location 0BH in ROM returns the address of the instruction following the call in HL. Knowing that address, it is easy to find the address of any location forward or backward from the address. The scheme that can be used for CALLs is to add a bias to the base address to compute the address of the return point, PUSH that return point, and then to JR to the

Listing 1. Encode/Decode

```
7F00                      ORG     7F00H
                   ;**********************************************************
                   ;*  BASE 40 ENCODING/DECODING PROGRAM.                    *
                   ;*    ENTER AT ENCODE FOR ENCODING:                       *
                   ;*      ENTRY: HL=> 3 ASCII BYTES FOR ENCODING            *
                   ;*      EXIT:  HL DESTROYED. 2 ENCODED BYTES IN BUFFER    *
                   ;*    ENTER AT DECODE FOR DECODING                        *
                   ;*      ENTRY: HL=> 2 ENCODED BYTES FOR DECODING          *
                   ;*      EXIT:  HL DESTROYED. 3 ASCII BYTES IN BUFFER      *
                   ;**********************************************************
7F00 CD7F0A        ENCODE CALL    0A7FH            ;GET BUFFER LOCATION
7F03 E5                   PUSH    HL               ;TRANSFER TO IY
7F04 FDE1                 POP     IY
7F06 210000               LD      HL,0             ;CLEAR ENCODE RESULT
7F09 0603                 LD      B,3              ;SETUP LOOP COUNT
7F0B 29            ENC010 ADD     HL,HL              ;*2
7F0C 29                   ADD     HL,HL              ;*4
7F0D 29                   ADD     HL,HL              ;*8
7F0E E5                   PUSH    HL                 ;SAVE
7F0F 29                   ADD     HL,HL              ;*16
7F10 29                   ADD     HL,HL              ;*32
7F11 D1                   POP     DE                 ;RESTORE *8
7F12 19                   ADD     HL,DE              ;*40
7F13 FD7E00               LD      A,(IY)             ;GET ASCII BYTE
7F16 CD667F               CALL    XLATE              ;CONVERT TO 0-39 CODE
7F19 FD23                 INC     IY                 ;BUMP BUFFER PNTR
7F1B 19                   ADD     HL,DE              ;MERGE CODE
7F1C 10ED                 DJNZ    ENC010             ;GO THREE TIMES
7F1E FD75FD               LD      (IY-3),L         ;SAVE LSB
7F21 FD74FE               LD      (IY-2),H         ;SAVE MSB
7F24 C9                   RET                      ;RETURN TO CALLING PROG
7F25 CD7F0A        DECODE CALL    0A7FH            ;GET BUFFER LOCATION
7F28 E5                   PUSH    HL               ;TRANSFER TO IY
7F29 FDE1                 POP     IY
7F2B 0603                 LD      B,3              ;SETUP LOOP COUNT
7F2D FD6E00               LD      L,(IY)           ;GET LSB
7F30 FD6601               LD      H,(IY+1)         ;GET MSB
7F33 DD217A7F             LD      IX,TABLE-1       ;TABLE START TO IX
7F37 114006               LD      DE,1600          ;40**2
7F3A B7            DEC010 OR      A                  ;CLEAR CARRY
7F3B ED52                 SBC     HL,DE              ;SUBTRACT TIL NEGATIVE
7F3D DD23                 INC     IX                 ;BUMP TABLE PNTR
7F3F 30F9                 JR      NC,DEC010          ;GO IF NOT NEGATIVE
7F41 19                   ADD     HL,DE            ;RESTORE RESIDUE
7F42 DD7E00               LD      A,(IX)           ;CONVERT TO ASCII
7F45 FD7700               LD      (IY),A           ;PUT IN BUFFER
7F48 112800               LD      DE,40            ;40**1
7F4B DD217A7F             LD      IX,TABLE-1       ;TABLE START TO IX
7F4F B7            DEC030 OR      A                  ;CLEAR CARRY
7F50 ED52                 SBC     HL,DE              ;SUBTRACT TIL NEGATIVE
7F52 DD23                 INC     IX                 ;BUMP TABLE PNTR
7F54 30F9                 JR      NC,DEC030          ;GO IF NOT NEGATIVE
7F56 19                   ADD     HL,DE              ;RESTORE RESIDUE
7F57 DD7E00               LD      A,(IX)             ;CONVERT TO ASCII
7F5A FD7701               LD      (IY+1),A           ;PUT IN BUFFER
7F5D 117B7F               LD      DE,TABLE         ;TABLE START TO DE
7F60 19                   ADD     HL,DE            ;POINT TO CHAR
7F61 7E                   LD      A,(HL)           ;GET LAST CHAR
7F62 FD7702               LD      (IY+2),A         ;SAVE IN BUFFER
7F65 C9                   RET                      ;RETURN TO CALLING PROG
7F66 DD21A27F      XLATE  LD      IX,TABLE+39      ;TABLE START+39
7F6A 1E28                 LD      E,40             ;# OF BYTES IN TABLE
7F6C DDBE00        XLA010 CP      (IX)               ;COMPARE A TO TABLE ENTRY
7F6F 2806                 JR      Z,XLA020           ;GO IF MATCH
7F71 DD2B                 DEC     IX                 ;BUMP TABLE PNTR
7F73 1D                   DEC     E                  ;DECREMENT COUNT
7F74 20F6                 JR      NZ,XLA010          ;GO IF MORE
7F76 1C                   INC     E                ;ADJUST POINTER
7F77 1D            XLA020 DEC     E                ;ADJUST INDEX
7F78 1600                 LD      D,0              ;INDEX NOW IN DE
7F7A C9                   RET                      ;RETURN
7F7B 30            TABLE  DEFM    '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ;,()'
     31 32 33 34 35 36 37 38
     39 41 42 43 44 45 46 47
     48 49 4A 4B 4C 4D 4E 4F
     50 51 52 53 54 55 56 57
     58 59 5A 3B 2C 28 29
0000                      END
00000 TOTAL ERRORS
```

Listing 2. Encode/Decode Exerciser

```
100 DEFUSR0=&H7F00:DEFUSR1=&H7F25
110 INPUT A$
120 B=VARPTR(A$):C=PEEK(B+1)+PEEK(B+2)*256:IF C>32767 T
    HEN C=C-65536
130 A=USR0(C)
140 PRINT PEEK(C),PEEK(C+1)
150 A=USR1(C)
160 PRINT A$
170 GOTO 110
```

Listing 3. Relocation Version

```
7F00            00100           ORG     7F00H
                00110 ;*****************************************************
                00120 ;*  BASE 40 ENCODING/DECODING PROGRAM.                *
                00130 ;*    ENTER AT ENCODE FOR ENCODING:                   *
                00140 ;*      ENTRY: HL=> 3 ASCII BYTES FOR ENCODING        *
                00150 ;*      EXIT:  HL DESTROYED. 2 ENCODED BYTES IN BUFFER *
                00160 ;*    ENTER AT DECODE FOR DECODING                    *
                00170 ;*      ENTRY: HL=> 2 ENCODED BYTES FOR DECODING      *
                00180 ;*      EXIT:  HL DESTROYED. 3 ASCII BYTES IN BUFFER  *
                00190 ;*****************************************************
7F00 CD7F0A     00200 ENCODE  CALL    0A7FH           ;GET BUFFER LOCATION
7F03 E5         00210         PUSH    HL              ;TRANSFER TO IY
7F04 FDE1       00220         POP     IY
7F06 210000     00230         LD      HL,0            ;CLEAR ENCODE RESULT
7F09 0603       00240         LD      B,3             ;SETUP LOOP COUNT
7F0B 29         00250 ENC010  ADD     HL,HL           ;*2
7F0C 29         00260         ADD     HL,HL           ;*4
7F0D 29         00270         ADD     HL,HL           ;*8
7F0E E5         00280         PUSH    HL              ;SAVE
7F0F 29         00290         ADD     HL,HL           ;*16
7F10 29         00300         ADD     HL,HL           ;*32
7F11 D1         00310         POP     DE              ;RESTORE *8
7F12 19         00320         ADD     HL,DE           ;*40
7F13 FD7E00     00330         LD      A,(IY)          ;GET ASCII BYTE
7F16 E5         00340         PUSH    HL              ;SAVE RESULT
7F17 CD0B00     00350         CALL    0BH             ;GET ADDRESS
7F1A 110700     00360 ENC020  LD      DE,BIAS1        ;DISPLACEMENT
7F1D 19         00370         ADD     HL,DE           ;COMPUTE ADDRESS OF XLATE
7F1E E5         00380         PUSH    HL              ;SAVE RETURN
7F1F 1855       00390         JR      XLATE           ;CALL XLATE
7F21 E1         00400 ENC030  POP     HL              ;RESTORE RESULT
7F22 FD23       00410         INC     IY              ;BUMP BUFFER PNTR
7F24 19         00420         ADD     HL,DE           ;MERGE CODE
7F25 10E4       00430         DJNZ    ENC010          ;GO THREE TIMES
7F27 FD75FD     00440         LD      (IY-3),L        ;SAVE LSB
7F2A FD74FE     00450         LD      (IY-2),H        ;SAVE MSB
7F2D C9         00460         RET                     ;RETURN TO CALLING PROG
7F2E CD0B00     00470 DECODE  CALL    0BH             ;GET ADDRESS
7F31 115F00     00480         LD      DE,BIAS2        ;DISPLACEMENT
7F34 19         00490         ADD     HL,DE           ;COMPUTE TABLE START
7F35 E5         00500         PUSH    HL              ;SAVE
7F36 CD7F0A     00510         CALL    0A7FH           ;GET BUFFER LOCATION
7F39 E5         00520         PUSH    HL              ;TRANSFER TO IY
7F3A FDE1       00530         POP     IY
7F3C 0603       00540         LD      B,3             ;SETUP LOOP COUNT
7F3E FD6E00     00550         LD      L,(IY)          ;GET LSB
7F41 FD6601     00560         LD      H,(IY+1)        ;GET MSB
```

Program continues

subroutine. This sequence is shown in Listing 3, lines 350—390. CALL 0BH returns the address of ENC020 in HL. DE is loaded with BIAS1 which is equal to ENC030-ENC020, the difference between the base address and the return point. This address is PUSHed onto the stack, and a JR is done to XLATE.

The same scheme could be used for JPs, but it's much simpler to use a JR in place of the JP. If the jump is beyond the range of the JR (−126 to +129 bytes), use a linking JR within the range.

Listing 3 also shows the bias scheme for LDs of addresses at lines 470—500. In this case we want to find the address of TABLE−1. CALLing 0BH returns the adress on the next instruction after Decode. Adding BIAS2 results in DECODE + 3 + TABLE − 1 − DECODE − 3 = TABLE − 1. This is PUSHed into the stack for future reference. A similar scheme is used at XLATE to compute the address of TABLE + 39.

Listing 3 is fully relocatable even though it is arbitrarily assembled at 7F00H (see Listing 4). It moves the code upward one byte so that the program starts at 7F01H, and then repeats the sequence.

You may wish to investigate other schemes to make all code relocatable. There is a great deal of work involved in calculating the CALL and immediate addresses; whether or not it's worth it depend on your application. An interesting sidenote—the Color Computer 6809 microprocessor easily produces fully relocatable code. You might take a look at its structure to see the methods it uses to reference the program counter.

The Comments Didn't Come Out . . .

I have rarely seen a computer manufac-

turer that released the source listings of an assembler or compiler. So what do you do when there are bugs in the utility software? You scrounge around in the code and correct the bugs and add the necessary patches.

There is a similar situation with software. Should a software house that has invested a huge chunk of time in software development release source listings? It's very frustrating to users not to have the source for specialized patches.

At this point the question appears to be academic. There is at least one book available providing a fully commented source listing of Level II ROM, and a series that goes a long ways towards explaining the internal workings.

The book is *Microsoft BASIC Decoded and Other Mysteries* by James Farvour (IJG Computer Services). It is a 310-page book that takes an Apparat disassembly and comments every line. You supply the operands of the instruction; everything else is printed. You will have to disassemble the code and put the pages of the disassembly alongside the comments, a formidable but worthwhile task. Every line is commented, with some expanded comments appearing on the back of the page. In addition to the disassembly, there are 61 pages of general text about ROM calls, the structure of ROM, variables, and disk operation.

The book is accurate, more so than any other book written about ROM thus far, with the exception of the next book I'm going to mention.

Farvour's book is rather steep, at $29.95. It is a fully annotated source listing of ROM, but don't expect to get a tutorial on the internal workings of the BASIC interpreter. It's not a workbook on designing BASIC interpreters, although it certainly eliminates a lot of research.

The Book, Volume I, by Insiders Software Consultants covers math operations in ROM. It is the first of three books describing Level II ROM. *The Book* is approximately 136 pages long, about 75 percent ROM disassembly.

If you would like a detailed, accurate description of all types of BASIC math operations (particularly floating-point operations) together with ROM calls and examples, this is the book for you. I would definitely recommend *The Book* at $14.95 for these subjects.

I'm often asked what assembly language books I would recommend. One that I would highly recommend, but with some warning, is *Practical Microcomputer Programming: The Z80* by Walter J. Weller, Northern Technology Books. First the bad news: The book does not use standard mnemonics. If you can get past the shock of using BZ ZILCH instead of JP NZ,ZILCH, the material in it is well-written and interesting. The book contains a listing of a complete Z-80 assembler, and is not inexpensive at $32.95.

One of these days my book *More TRS-80 Assembly Language Programming* will emerge from the depths of Radio Shack. It covers the Disk Assembler and other intermediate assembly language topics.

```
7F44 DDE1      00570        POP    IX           ;GET TABLE START-1
7F46 DDE5      00580        PUSH   IX           ;SAVE TABLE START-1
7F48 114006    00590        LD     DE,1600      ;40**2
7F4B B7        00600 DEC010 OR     A              ;CLEAR CARRY
7F4C ED52      00610        SBC    HL,DE          ;SUBTRACT TIL NEGATIVE
7F4E DD23      00620        INC    IX             ;BUMP TABLE PNTR
7F50 30F9      00630        JR     NC,DEC010      ;GO IF NOT NEGATIVE
7F52 19        00640        ADD    HL,DE        ;RESTORE RESIDUE
7F53 DD7E00    00650        LD     A,(IX)       ;CONVERT TO ASCII
7F56 FD7700    00660        LD     (IY),A       ;PUT IN BUFFER
7F59 112800    00670        LD     DE,40        ;40**1
7F5C DDE1      00680        POP    IX           ;GET TABLE START-1
7F5E DDE5      00690        PUSH   IX
7F60 B7        00700 DEC030 OR     A              ;CLEAR CARRY
7F61 ED52      00710        SBC    HL,DE          ;SUBTRACT TIL NEGATIVE
7F63 DD23      00720        INC    IX             ;BUMP TABLE PNTR
7F65 30F9      00730        JR     NC,DEC030      ;GO IF NOT NEGATIVE
7F67 19        00740        ADD    HL,DE          ;RESTORE RESIDUE
7F68 DD7E00    00750        LD     A,(IX)         ;CONVERT TO ASCII
7F6B FD7701    00760        LD     (IY+1),A       ;PUT IN BUFFER
7F6E D1        00770        POP    DE           ;GET TABLE START-1
7F6F 13        00780        INC    DE           ;NOW TABLE START
7F70 19        00790        ADD    HL,DE        ;POINT TO CHAR
7F71 7E        00800        LD     A,(HL)       ;GET LAST CHAR
7F72 FD7702    00810        LD     (IY+2),A     ;SAVE IN BUFFER
7F75 C9        00820        RET                 ;RETURN TO CALLING PROG
7F76 CD0B00    00830 XLATE  CALL   0BH          ;GET ADDRESS
7F79 113F00    00840        LD     DE,BIAS3     ;DISPLACEMENT
7F7C 19        00850        ADD    HL,DE        ;COMPUTE TABLE ADDRESS
7F7D E5        00860        PUSH   HL           ;TRANSFER TO IX
7F7E DDE1      00870        POP    IX
7F80 1E28      00880        LD     E,40         ;# OF BYTES IN TABLE
7F82 DDBE00    00890 XLA010 CP     (IX)           ;COMPARE A TO TABLE ENTRY
7F85 2806      00900        JR     Z,XLA020       ;GO IF MATCH
7F87 DD2B      00910        DEC    IX             ;BUMP TABLE PNTR
7F89 1D        00920        DEC    E              ;DECREMENT COUNT
7F8A 20F6      00930        JR     NZ,XLA010      ;GO IF MORE
7F8C 1C        00940        INC    E            ;ADJUST POINTER
7F8D 1D        00950 XLA020 DEC    E            ;ADJUST INDEX
7F8E 1600      00960        LD     D,0          ;INDEX NOW IN DE
7F90 C9        00970        RET                 ;RETURN
7F91 30        00980 TABLE  DEFM   '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ;,()'
     31 32 33 34 35 36 37 38
     39 41 42 43 44 45 46 47
     48 49 4A 4B 4C 4D 4E 4F
     50 51 52 53 54 55 56 57
     58 59 5A 3B 2C 28 29
0007           00990 BIAS1  EQU    ENC030-ENC020
005F           01000 BIAS2  EQU    TABLE-1-DECODE-3
003F           01010 BIAS3  EQU    TABLE+39-XLATE-3
0000           01020        END
00000 TOTAL ERRORS
```

Listing 4. Relocation Test

```
50 FOR I=254 TO 0 STEP -1
60 POKE 32513+I,PEEK(32512+I)
70 NEXT I
100 DEFUSR0=&H7F01:DEFUSR1=&H7F2F
110 INPUT A$
120 B=VARPTR(A$):C=PEEK(B+1)+PEEK(B+2)*256:IF C>32767 T
    HEN C=C-65536
130 A=USR0(C)
140 PRINT PEEK(C),PEEK(C+1)
150 A=USR1(C)
160 PRINT A$
170 GOTO 110
```

The Fourth Great Assembly Line Programming Contest

All right—you asked for it. Here's the fourth Assembly Line Programming Contest. In honor of the 8080 precursor of the Z-80, there are eight questions. Answer them briefly, send the results to me, and you may be mentioned in a computer magazine column as well as receiving a trivial prize. We'll have two winners, a novice class winner (less than 100 hours of assembly language experience) and a hacker winner (more than 100 hours). Send your entries to the address at the end of this column no later than the 20th of the month and be sure to specify what kind of fool you are.

Number One: What is the tightest timing loop that can be written for counts of

"Send your entries to the address at the end of this column. . . and be sure to specify what kind of fool you are."

1—65536? An example of a slow loop is:

```
        (HL) = count, 0-65535
LOOP    DEC     HL
        LD      A,H
        OR      L
        JR      NZ,LOOP
```

Number Two: What do these three instructions do to the contents of HL?

```
SR1     CALL    SR2
SR2     ADD     HL,HL
        RET
```

Number Three: What does the following code do? (Thanks to Ron Markel.)

```
LD      (HL),A
DEC     B C
LD      D , H
LD      E , L
INC     D E
LDIR
```

Number Four: A value of 0—15 decimal must be converted as shown below. What are the fewest number of instructions to do this?

0 to 30H	4 to 34H	8 to 38H	C to 42H
1 to 31H	5 to 35H	9 to 39H	D to 43H
2 to 32H	6 to 36H	A to 40H!	E to 44H
3 to 33H	7 to 37H	B to 41H	F to 45H

Number Five: Find the integer portion of log base 2 of a value in the A register, with the result in any register. Examples:

1 produces a value of 0
2 produces a value of 1
3 produces a value of 1
4 produces a value of 2
5 produces a value of 2
15 produces a value of 3
25 produces a value of 4
37 produces a value of 5

Number Six: What does the following code do?

```
        JR      NEXT+1
NEXT    JR      NEXTP1-1
NEXTP1  .
        .
```

Number Seven: A contains a signed value. What does it contain after:

```
NEG
CPL
```

Number Eight: What is the purpose of TWDEE and TWDUM? (Thanks to Bill Schroeder of Galactic Software.)

```
SR      CALL    TWDEE
        .
        .
        .
TWDEE   EX      (SP),HL
        PUSH    DE
        PUSH    BC
        PUSH    AF
        LD      DE,TWDUM
        PUSH    DE
        JP      (HL)
TWDUM   POP     AF
        POP     BC
        POP     DE
        POP     HL
        RET
```

That's it for this month. Keep sending those cards, letters, and contest entries to: William Barden, Jr., Department Infinite Loop, 28182 Palmada, Mission Viejo, CA 92692. ■

✓See List of Advertisers on page 306

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

"I watched the... salesman trying to save a... sale.... he lost and the customer wandered out... convinced that the 'ugly box' could have little use in his office."

I don't like the color of the box." That comment, overheard at a local Radio Shack Computer Center made me drop what I was doing and strain to hear the rest of the conversation.

"If it were only blue, or a more stylish color, instead of that ugly grey color, I would buy it." The speaker was referring to the Model II on which a General Ledger demo was running.

Fighting my impulse to join the discussion, I watched the young salesman desperately trying to save a potential sale. Unfortunately, he lost and the customer wandered out completely convinced that the "ugly box" could have little use in his office.

In computers as well as books, a pretty cover can be very deceiving. The value of a popularly priced, mass market computer cannot be seen until you consider the wealth of software that has become available for it. With the flexibility of the Model II, one can move out of the TRSDOS environment into the business standard, CP/M. With this operating system, thousands of useful business programs are available at exceptionally low costs.

VisiCalc User Report

The VisiCalc program dramatically demonstrates this usefulness. This program can pay a user back almost immediately. Brian Gaines, a friend and stock market Registered Representative, put it to use immediately.

Brian uses VisiCalc as a portfolio monitor in an interesting way, and has offered to share his experiences with our readers. I'll turn the column over to him.

One of the most often promised and seldom delivered offers of a stockbroker is the portfolio review. Writing out a client's portfolio list and calculating a variety of totals and percentages is time-consuming and tedious work. When finished, it has to be refigured for each proposed change and updated to reflect constantly changing prices.

Typing all this data requires a good two hours for each ten positions, just to get a first draft, then the draft must be proofed and re-typed. Frankly, the typos alone will discourage you from prospecting for new portfolios.

Of course, there had to be a better way, and VisiCalc and the Model II appear to be the answer. Anything you currently calculate in tabular form can be handled by this combination. Some examples include:

- Covered Options Writing
- Portfolio Profit and Loss Schedules
- Commissions Due Schedules
- Margin Interest Charged Reports
- Portfolio Evaluations

Portfolio evaluations alone have already paid for my own system many times over. The ability to produce such alternative evaluations for clients have resulted in transactions that have netted me commissions of up to $29,000. With VisiCalc I can evaluate 40 possible transactions for my client in a short time; by hand the best I could have done was only four or five.

To set up my portfolio evaluation worksheet, I load VisiCalc and type the column headings as shown in Fig. 1.

Items labeled "None" require manual entry, and all others are automatically calculated by the VisiCalc program. Once the position is entered, only the current market figures have to be changed to update the position instantly. As an extra bonus, if position data is entered by category, such as equities, munis, corps, converts or options, subtotals can be created for each category.

Every time you change the information in any of the elements all figures will be recomputed.

Once the portfolio has been set up, it should be saved on disk for future use. VisiCalc can handle even the largest portfolios—up to 64 column headings and 100 lines of positions.

With the worksheet and my Lineprinter IV, I can print a copy of the worksheet on embossed stationery. No typos, no wasted time. With VisiCalc, I can tell a client that he has, for example, only 18 percent of his portfolio in utilities and that his overall yield would increase 200 basis points (2 percent) if he buys another 400 Con Ed.

Visicalc's speed allows me to display a portfolio, update its current market value and discuss alternatives with my clients over the phone. Best of all, we can discuss "what if" strategies and see the results dynamically displayed.

The Target System

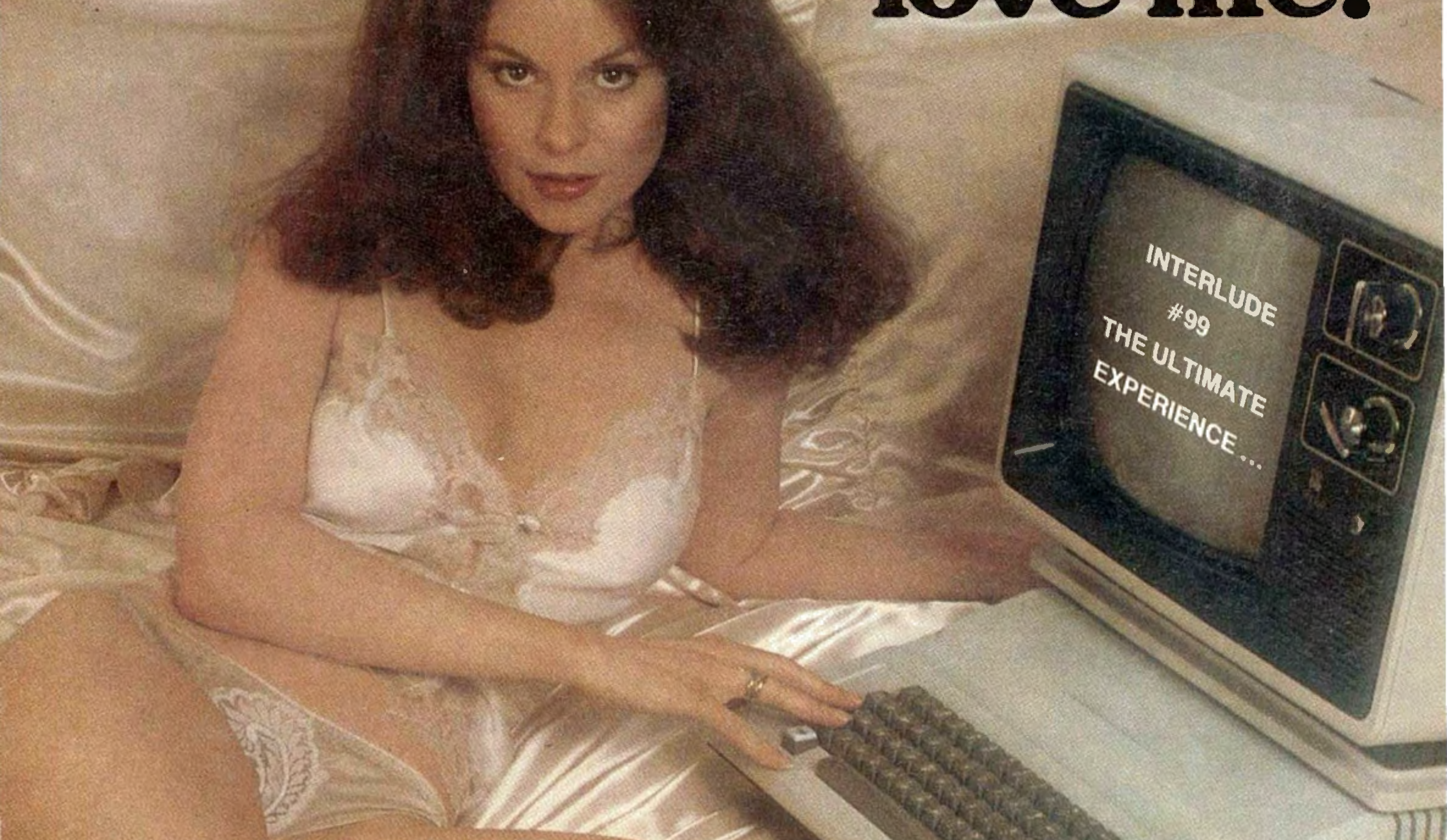
Brian has illustrated one effective use of VisiCalc, but it's not the only management tool that can have an immediate payback. In last month's column, I suggested that VisiCalc could be used for management planning. No sooner had I posted the column, when I received an intriguing flyer from a firm called Advanced Management Strategies, Inc., Atlanta, GA.

This organization has developed a planning system called Target. Target is intended to be the first element in a series of management planning assists to run on a CP/M system. Needless to say, I request-

COL	HEADING	CALCULATION	DESCRIPTION
A	Security	None	Name or ticker symbol
B	No. of units	None	No. of shares or the face value of bonds.
C	Total cost	None	Original cost or sales proceeds (negative)
D	CMV/unit	None	Current market value
E	Total CMV	B x D	Extension col. B x col. D
F	% to CMV	B/totEx100	% of holding at current market value
G	Current P&L	E-C	Unrealized gain or loss
H	% P&L	G/C x 100	Current unrealized gain or loss as a % of total
I	Inc/unit	None	Dividend per share or interest per $1.00 face
J	Annual Inc	B x I	Income for position
K	Current Yld	J/E x 100	Yield for position
L	LT or ST	None	Long term or short term or transaction date

Fig. 1. Portfolio Evaluation Worksheet

How's your love life?



A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

Interlude is: romantic...playful...outrageous...a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).

More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?

Interlude™
The Ultimate Experience.

INTERLUDE, 10635 Richmond, Houston, Texas 77042. I'm really ready. Send my Interlude today.

Apple II (16K)*
☐ Cassette ($16.95)
☐ Diskette ($19.95)
☐ Diskette—Pascal or DOS 3.3 ($19.95)
Add $1.50 for shipping and handling.

TRS-80 (Level II-16K)**
☐ Cassette ($16.95)
☐ Diskette ($19.95)

Poster
☐ 20"x 24" reproduction of this ad without ad copy ($4.95—includes shipping charges)

Available for immediate shipment.

Please enclose your check payable to INTERLUDE or complete the charge information:

☐ MASTERCARD ☐ VISA All charge customers must sign here. ____

Account No. ____ Expiration date ____ MasterCard Bank Code ____

CHARGE CUSTOMERS: Order by phone toll-free! **1-800-231-5768 Ext. 306** (Texas: 1-800-392-2348 Ext. 306)

Name ____ Age ____

Address ____

City ____ State ____ Zip ____

*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

ed a copy immediately.

Within a week, I received a 60-plus page manual, an eight inch disk, and a reference card. Compared to some, the manual is a model of clarity. It contains four color-coded sections. The first contains step by step instructions on how to use the system and an index. The second section contains specifications for the version of Target supplied.

In a special section are models which illustrate the features of the Target system. (These models were thoughtfully reproduced on the system disk furnished.) The remaining section highlights enhancements added to the version of Target supplied, which have not been included in the first section of the manual.

The manual suggests reading the second section on specifications *before* trying to use Target. This recommendation should be observed, especially if the user has never used a CP/M (Control Program Microcomputer) operating system before. Attempting to boot up a CP/M disk without that operating system results in an RS error rather than a No System message.

The Target, version 2.1 supplied to me is designed to work with Lifeboat Associates, New York, NY CP/M version 2.2 or greater. It consists of five programs and three files. Each program is a machine language compilation of a BASIC program. Therefore, you will be unable to list the source code.

One major advantage of a compiled BASIC program is that its execution speed is considerably greater than that of an interpreted program. Target runs considerably faster than the T/Maker system I reviewed previously. Another major advantage, or disadvantage depending on your point of view, is that you cannot tinker with the source code. Therefore, you will be completely dependent on the vendor for program enhancements or corrections should bugs occur.

Compared to VisiCalc

Advanced Management Strategies takes their obligation to support their customers seriously. During the evaluation phase, when I had difficulty with a program, their program support director had a copy of the system disk sent to me. Together we worked on and resolved the problem.

Once up and running, Target bears a strong resemblance to VisiCalc. Like VisiCalc, data is presented in tabular form. Calculations can be made to sum lines, generate data, create percentage relationships and sum columns. However, Target goes beyond VisiCalc by allowing logical tests of these calculations. IF tests, for example, can be made by the computer and the calculation of data altered dynamically.

To illustrate this, let's consider calculating an income tax. Typically, income taxes are calculated using a table of values, with the tax bite getting progressively greater at each table break. Target's logical tests can identify a step point and use a higher tax rate should projected income increase. This feature will be particularly handy projecting purchases where volume discounts are a factor.

Unlike VisiCalc, data and resulting calculations are not entered directly on a worksheet. Target treats this information as a separate file. To place a display file of information in a format suitable for printing, the file must be run. After a display file is created, it can be examined at any time by hitting the ESC key.

. . . "you will be completely dependent on the vendor. . . should bugs occur."

After becoming accustomed to VisiCalc, data entry to a file seems archaic, however, there are significant advantages. The most important advantage is that all calculations and information are obvious in reviewing the file. With VisiCalc, it is easy to lose track of which figures are input and which are calculated. Target gets around the forgetfulness factor by requiring row labels be used as descriptors when defining calculations.

Row labels are limited to eleven characters; any more are ignored. If more than one word is used as a label, an apostrophe (') must be entered in place of the space. The apostrophe is printed as a space. While this does not result in very descriptive row titles, careful abbreviations generate acceptable reports.

Target is loaded from the CP/M by simply typing "TARGET". The initial display prompts you to enter the code of the data disk. In a CP/M system the drives are coded A,B,C,D, corresponding to the TRSDOS 0,1,2,3. Once the disk is named, Target requests the number of columns desired. If the return key is depressed, a default of three is selected.

The system next prompts you for the number of decimals to be displayed on the output reports. If return is selected, the system will default to an integer output. Finally, you are asked to specify the output report date. Selecting return suppresses the date and displays the main menu.

Change Command Available

From the menu, the user can create a new model, or recall a previously saved model. To facilitate recall, a directory command is available which displays file names of models stored on the disk. The menu also allows the operator to save a memory file or erase a previously saved disk file. Should you decide to alter a model's size or number of decimal places in the output report, a change command is available on the menu.

Once an option is selected, the system displays the data entry screen. Its chief feature is the guideline displayed on line one that uses the reverse print highlighting feature of the Model II. Within the guideline is exhibited the model name, number of characters available for model storage, column counter, decimal indicator and mode indicator.

Target uses historical data to calculate initial model values. These historical values are entered in parentheses. The historical values do not appear on the output reports.

Line commands can contain data, arithmetical operations (^,*,/,+,−), logical statements (IF. . . THEN. . . ELSE) or functions. Like VisiCalc, Target supports line summing functions, MIN, MAX, and AVE. Unlike VisiCalc, there is no present value function. Target has the unique functions of CUM, GREATER OF, LESSER OF and GROW BY.

The CUM function enters the cumulative total of all the lines to that point into the current column. GREATER OF and LESSER OF allow a choice of values to be entered into the report and subsequent calculations. GROW BY allows a programmed growth factor to be applied to each value calculated for a column within the row.

Target has column commands which are applied to the entire column vector. Arithmetic, logical and some functional operations are permitted.

To further dress up your report, heading, skip, underline and overline commands are accepted. The heading can contain up to 252 characters. After formatting is complete and the model is run, the report can be saved or printed. You can print the entire report or sections.

With its advanced features and quick execution time, Target is worth a good long look at your nearest computer store. ■

80 APPLICATIONS

by Dennis Kitsz

"This month's column presents a miniature window similar to those in early microcomputers and minicomputers."

Hmmmmm, let's see. S-Y-S-T-E-M. Okay, enter that. Good, there's the prompt. Now, what was that address? Aha, here it is. Type the slash, then 2-0-4-8-0. Now enter that. Gone. Damn! Electronic never-never land again! Where is it? What's it doing? Just locked up... maybe I'll hit reset. Seems to...-Arrgh! MEMORY SIZE?

If you've been entering machine language programs—your own, or those from *80 Microcomputing* or elsewhere —chances are you have experienced maddening moments like these. Positive the coding is correct, you direct the computer to execute the routine, and it promptly hangs up. If you're skilled at using machine language monitors and breakpoints, it's possible (very painstakingly) to trace the course of the program until you find a flawed area. The process is time-consuming and unrewarding.

"The micro front panel is simple to build, and can be completed in an evening..."

Certain machine language monitor programs offer a kind of window into the computer's operation by emulating a program rather than letting it run under its own control. It seems like the program is running, but it is actually being run by the emulator. Instructions appear slowly disassembled on the screen and, if you've got the time, the program will eventually reach the trouble spot.

Earlier microcomputers had few of these software comforts. Programs were entered a byte at a time by selecting addresses through a bank of 16 switches, and entering data using eight other switches. It was all binary. But that method *did* provide a hardware "window"—each address and data bit was displayed on a row of LEDs which winked and blinked as the computer performed its operations. This switch-laden window was called the front panel.

If the computer happened to hang up, the address area would be flashing away before your eyes. With a quick calculation, the binary digits could be converted to hexadecimal (or octal in those misty past days), the computer recalled from its locked up state, and the incorrect code changed.

This month's column presents a miniature window similar to those in early microcomputers and minicomputers. You may watch the address and data change while the CPU reads from memory, writes to memory, inputs from a port, or outputs to a port. The entire project fits on a one-inch by four-inch piece of perfboard, which plugs into the edge card connector of the TRS-80 (Photo 1).

For this project, you will need four integrated circuits (one 74LS20 and three 74LS373), twenty-four 270-ohm resistors, four 1K-ohm resistors, one four-position DIP switch, and a 5-volt power source. A transistor battery and a 7805 (also known as a 340T-5) voltage regulator will do for the 5-volt source or a 9-volt battery eliminator together with the voltage regulator. The display will also require 24 micro LEDs (sold in packs of four for 88 cents at Radio Shack, catalog number 276-032) or three 10-step LED bar displays (also at Radio Shack, $3.79 each, catalog number 276-081).

Fig. 1 presents the complete circuit for the micro front panel, and Fig. 2 shows two power supply options for it. The sixteen address lines and eight data lines are fed to the inputs of the 74LS373s. These integrated circuits contain eight latched flip-flops in each; when a rising clock pulse is sent to pin 11 of these circuits, the data present at their inputs is latched to the outputs.

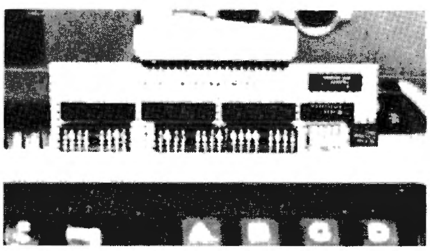


Photo 1. Micro front panel connected to the TRS-80. It is slightly wider than the numeric keypad.

The read (RD), write (WR), input (IN), and output (OUT) lines lead through a switch to the input of a 74LS20 four-input NAND gate (Z4). When the switches are open, the inputs to Z4 are all high, holding its output low. When a switch is closed, the output of Z4 remains low until the respective input line is activated. A read signal, for example, switches the RD line from its normal high (one) state to a low (zero) state. The presence of a low on any input of the NAND gate causes its output to swing high, latching data and address into the 74LS373s, Z1 to Z3.

"In any case, enjoy. Just watching those LEDs is a delight."

The outputs of Z1, Z2 and Z3 should not normally be expected to light LEDs. But that is their only function in this circuit, so the current demanded by the LEDs does not overexert the 74LS373s.

The micro front panel is simple to build, and can be completed in an evening using wire-wrapping technique. The individual LEDs can be inserted into integrated circuit sockets to avoid soldering them or wire-wrapping their smooth, round leads. When ten-segment bar displays are used, they can be inserted into sockets; two segments of each package are left unconnected, breaking the display into three groups of eight digits.

To have the displays read properly in binary, the least significant bit of the address bus should be on the right, and the most significant bit on the left. The pat-

tern applies to the data bus as well.

An Experiment

When the micro front panel is complete, plug it into the edge card of the computer, apply power, and turn the WR switch on. The LEDs will probably come on randomly, but as soon as the TRS-80 receives power they will stabilize into a discernible flicker surrounding one area of memory. Disk users, make sure you are in Level II BASIC, and press enter in response to MEMORY SIZE?

The addresses will read straight up through available memory, incrementing rapidly in binary. The effect will be most dramatic if you have a full 48K system.

Disk users can power up the system with no disk in place, watch the disk be commanded to spin, and the computer hang up. The address line will probably read 0100001000000001—that is 4201 hex, where the program expects to find the disk boot program.

From Level II BASIC, try this short routine:

```
POKE 20480,24 : POKE 20481,254   <ENTER>
SYSTEM                           <ENTER>
/20480                           <ENTER>
```

The system, as expected, hangs up. This program is the simple endless loop JR FE—that is, jump back to itself. Switch the WR line off and the RD line on. The address display will flicker in one area, but will mainly read 0110000000000000, which is 5000 hex (20480 decimal—see box). The least significant bit will be dim, meaning both 5000 and 5001 are being accessed alternately.

Press the reset button to get out of this loop, and put this routine in place:

```
POKE 20480,60 : POKE 20481,50  <ENTER>
POKE 20482,0  : POKE 20483,60  <ENTER>
POKE 20484,24 : POKE 20485,250 <ENTER>
SYSTEM                         <ENTER>
/20480                         <ENTER>
```

The video screen will jitter and shake, and at the top left corner a rapidly changing character will appear. With the RD line still switched in place, the display will flicker around 0110000000000000 (5000 to 5005 hex). Now switch the RD line off and the WR back on.

The display should show 0011110000000000, which is 3C00 hex, the first location of the video screen. In other words, the program is reading from 5000 to 5005 hex, but writing to the screen at 3C00 hex. The program in assembly language reads:

```
INC A
LD  (3C00),A
JR  FA
```

Again, it is an endless loop, each time incrementing the value of A and writing it to the top left corner of the screen.

For the third demonstration, insert a cassette into the player as if you were going to load a program. Actually, the cassette may be any cassette, so long as something is recorded on it. Now enter this program:

```
POKE 20480,58  : POKE 20481,61   <ENTER>
POKE 20482,64  : POKE 20483,211  <ENTER>
POKE 20484,255 : POKE 20485,219  <ENTER>
POKE 20486,255 : POKE 20487,50   <ENTER>
POKE 20488,0   : POKE 20489,60   <ENTER>
POKE 20490,24  : POKE 20491,244  <ENTER>
SYSTEM                           <ENTER>
/20480                           <ENTER>
```

In assembly language, this translates to:

```
LD  A,(403D)
OUT (FF),A
IN  A,(FF)
LD  (3C00),A
JR  F4
```

An explanation is in order here. At location 403D hex (16445 decimal), the status of port FF (255 decimal) is stored for use by BASIC during CHR$ (23), CLOAD, and SYSTEM. This program reads that status and sends it to port FF. When this information is written to that port, it resets the data input flip-flop without disturbing the cassette relay or screen character size.

In other words, when data comes into the cassette port it sets the flip-flop trigger. To read the presence of data, the

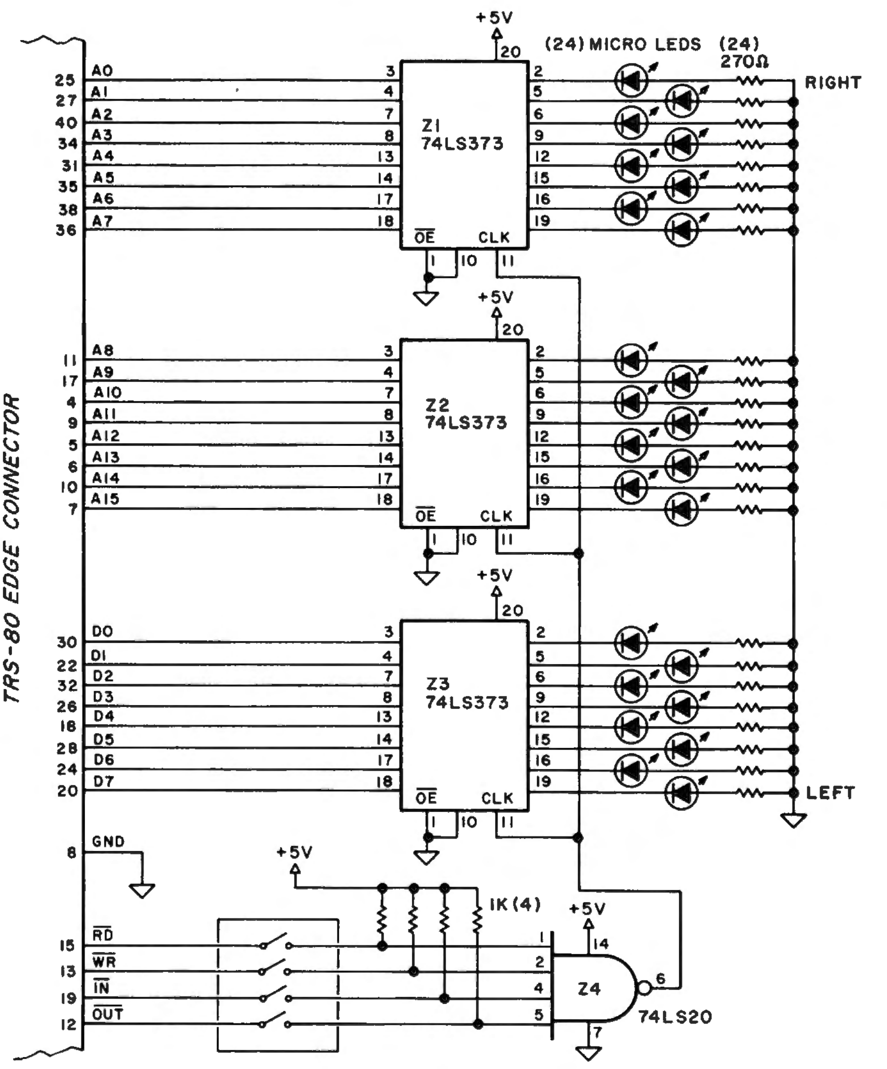


Fig. 1. Micro Monitor

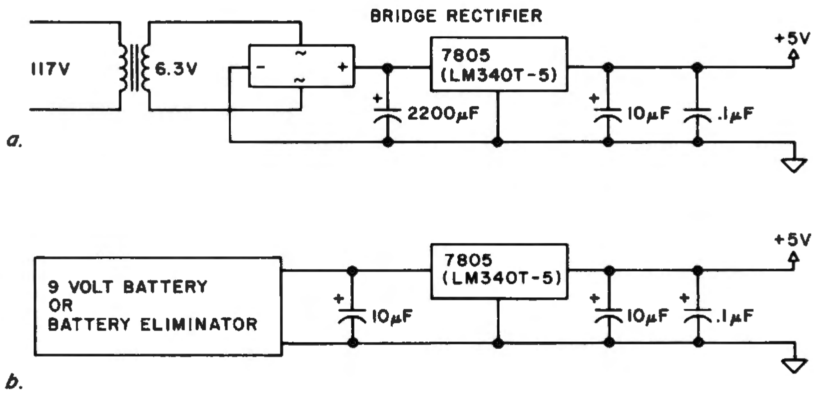
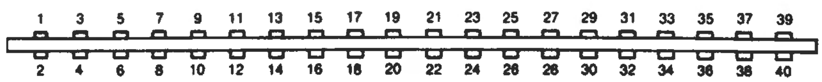


Fig. 2. Power Supplies for Micro Front Panel

P/N	SIGNAL NAME	DESCRIPTION
1	RAS*	Row Address Strobe Output for 16-Pin Dynamic Rams
2	SYSRES*	System Reset Output, Low During Power Up Initialize or Reset Depressed
3	CAS*	Column Address Strobe Output for 16-Pin Dynamic Rams
4	A10	Address Output
5	A12	Address Output
6	A13	Address Output
7	A15	Address Output
8	GND	Signal Ground
9	A11	Address Output
10	A14	Address Output
11	A8	Address Output
12	OUT*	Peripheral Write Strobe Output
13	WR*	Memory Write Strobe Output
14	INTAK*	Interrupt Acknowledge Output
15	RD*	Memory Read Strobe Output
16	MUX	Multiplexor Control Output for 16-Pin Dynamic Rams
17	A9	Address Output
18	D4	Bidirectional Data Bus
19	IN*	Peripheral Read Strobe Output
20	D7	Bidirectional Data Bus
21	INT*	Interrupt Input (Maskable)
22	D1	Bidirectional Data Bus
23	TEST*	A Logic "0" on TEST* Input Tri-States A0–A15, D0–D7, WR*, RD*, IN*, OUT*, RAS*, CAS*, MUX*
24	D6	Bidirectional Data Bus
25	A0	Address Output
26	D3	Bidirectional Data Bus
27	A1	Address Output
28	D5	Bidirectional Data Bus
29	GND	Signal Ground
30	D0	Bidirectional Data Bus
31	A4	Address Bus
32	D2	Bidirectional Data Bus
33	WAIT*	Processor Wait Input, to Allow for Slow Memory
34	A3	Address Output
35	A5	Address Output
36	A7	Address Output
37	GND	Signal Ground
38	A6	Address Output
39	+5V	5 Volt Output (Limited Current)
40	A2	Address Output

NOTE: * means Negative (Logical "0") True Input or Output



Mates with AMP P/N 88103-1 Card Edge Connector or Equivalent

Fig. 3. Edge card connector, from the TRS-80 Technical Reference Handbook

computer must first reset the trigger without messing up the screen or turning off the cassette motor. The first two instructions perform that function. The command IN A,(FF) reads that information into the accumulator, from where it is subsequently written to the screen.

Flip off the RD and WR lines of the micro front panel, and turn on the IN line. Start the cassette moving, and watch as the cassette data bit flickers in concert with the incoming audio signal. Try RD and WR in turn as well, which will show the program being read from the 5000 area and data being written to 3C00.

Finally, turn off all three lines and flip the switch to OUT. You can now see the steady data from 403D that is being output to port FF.

What Earthly Use?

Naturally, a front panel is not for every-

Reading the Binary

Looking at 24 blinking LEDs is a treat. It also gives you a feeling that you've got a *real* computer there. But to use the little flasher, you have to read the binary information.

When the LED is on, count that as a one; when it's off, you've got a zero. Break the string of ones and zeros into blocks of four, and use this table to convert the binary number to hexadecimal.

0000	0
0001	1
0010	2
0011	3
0100	3
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Easy as can be. 1001011011010101 breaks into 1001 0110 1101 0101, which you can convert quickly to 9 6 D 5. To determine the decimal equivalent of this number, if you need it, refer to my "Applications" column in the March *80 Microcomputing.*

COLLEGE BOARDS
for TRS-80 PET, APPLE

The best way to sharpen your skills for the College Boards is to work on actual examinations. Each of these program sets confronts the user with a virtually limitless series of questions and answers Each program is based on past exams and presents material of the same level of difficulty and in the same form as used in the College Board examination. Scoring is provided in accordance with the formula used by College Boards.

SAT, PSAT, N.M.S.Q.T., set includes 7 programs covering Vocabulary, Word Relationships, and Mathematics. Price $79.95

EDUCATOR EDITION - SAT, PSAT Includes all of the above programs plus detailed solutions and explainations for each problem plus drill exercises. SAT set includes 14 programs. $149.95

GRADUATE RECORD EXAMINATION set includes 10 programs covering Vocabulary, Word Relationships, Mathematics, Logical Diagrams, Analytical Reasoning. $139.95

EDUCATOR EDITION - Graduate Record Exam Set includes 20 programs. $199.95

SWORD OF ZEDEK

Fight to overthrow Ra, The Master of Evil. In this incredible adventure game, you must confront a host of creatures, natural and supernatural. To liberate the Kingdom, alliances must be forged and treasures sought. Treachery, deceit and witchcraft must be faced in your struggles as you encounter wolves, dwarves, elves, dragons, bears, owls, orcs, giant bats, trolls, etc. Each of the 12 treasures will enhance your power, by making invisible, invulnable, more eloquent, more skillful in combat etc., etc., as you explore the realms of geography both on the surface and underground. Dungeons, temples, castles, mountains etc., are all a part of the fantastic world of Ra. Each game is unique in this spectacular and complex world of fantasy. $24.95

KRELL GAME PAK

Incredible bargain, 15 unique and challenging games!!!

Hostage	Primary Fight
Prime Time	Black Gold
The Black Death	Wordsworth
Star Clipper	Hard Scrabble
Bulls & Bears	Bible Quode
Banzai	Shakespeare Quode
Banzai II	Bill of Rights Quode
Super Banzai	*TRS-80 only, $39.95*

SUPER STAR BASEBALL

ALL TIME SUPER STAR BASEBALL Sample Lineup

B. Ruth	T. Williams
L. Gehrig	J. Foxx
J. DiMaggio	H. Greenberg
J. Jackson	R. Hornsby
G. Sisler	H. Wilson
S. Musial	B. Terry
T. Cobb	M. Mantle
W. Mays	H. Aaron
C. Young-P	W. Johnson-p

SUPER STAR BASEBALL Sample Lineup

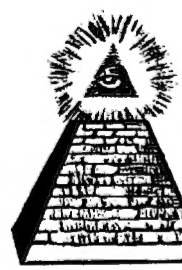
D. Parker	J. Rice
W. Stargell	H. Aaron
W. Mays	L. Brock
P. Rose	R. Carew
O. Cepeda	H. Killebrew
C. Yazstremski	R. Allen
W. McCovey	R. Leflore
R. Jackson	R. Zisk
G. Brett	B. Madlock
R. Guidry-P	T. Seaver-p

Performance is based on the interaction of actual batting and pitching data. Game can be played by one or two players with the computer acting as a second player when desired. Players select rosters and lineups and exercise strategic choices including hit and run, base stealing, pinch hitting, intentional walk, etc. Highly realistic, there are two versions, ALL TIME SUPER STAR BASEBALL, and SUPER STAR BASE—BALL featuring players of the current decade. Each includes about 50 players allowing nearly an infinite number of roster and lineup possibilities.

$14.95

TIME TRAVELER

The best of the adventure games. Confronts player with complex decision situations and the demand for real time action. Using the Time Machine, players must face a challenging series of enviornments that include; the Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, The Crusades, Machiavelli's Italy, the French Revolution, the American Revolution, and the English Civil War. Deal with Hitler's Third Reich, Vikings, etc. At the start of each game players may choose a level of difficulty... the more difficult, the greater the time pressure. To succeed you must build alliances and struggle with the ruling powers. Each game is unique. $24.95



✓375

*ALL PROGRAMS AVAILABLE FOR TRS-80, APPLE II & PET

*Programs for APPLE or TRS-80 are on disk or cassett, please specify.

All programs require 16K●TRS-80 programs require LEVEL II BASIC●APPLE programs require Applesoft BASIC

KRELL SOFTWARE

Send check or money order to
21 Milbrook Drive, Stony Brook, NY 11790
(516) 751-5139

one. If your forte is BASIC, chances are you'll hardly ever need it. But then, it's reassuring to be able to see BASIC working when it pauses for one of those seemingly interminable string sorts.

In any case, enjoy. Just watching those LEDs is a delight. If there is enough interest, a printed circuit board will be made available for this project. Please write to me at Roxbury, VT 05669. Enclose a self-addressed, stamped envelope.

Reminder: The Edge Connector

I've recommended that you all purchase and read the *TRS-80 Technical Reference Handbook*, which contains the pin assignments of the edge card connector. But just as a yearly reminder, it's presented in Fig. 3.

High-Speed Update: I am still receiving many letters about the speed-up modification which appeared in the February 1980 issue of this magazine. This speed-up works well for all versions of the TRS-80 Model I, but some expansion interfaces (those manufactured since the spring of 1980) may need a few changes. These are the expansion boxes that don't need the buffered cable.

The changes involve the way memory is selected in the expansion box, because these interfaces were fixed to run at 1.77 MHz and no faster. First of all, be sure you have RAMs rated at 300 nS or better, which is usual for newer expansion interfaces.

There are seven steps:

1. Find Z37 and Z38.
2. Cut the trace leading from Z38, pin 9. Call the end furthest from Z38 "loose end A".
3. Cut the trace leading from Z38, pin 8. Call the end furthest from Z38 "loose end B".
4. Cut the trace leading from Z37, pin 4. Call the end furthest from Z37 "loose end C".
5. Attach loose end A to Z38, pin 11.
6. Attach loose end B to Z37, pin 4.
7. Attach loose end C to Z38, pin 9.

With these changes in place, memory will be selected faster and asked to produce valid data from the computer much faster. Technically, the Radio Shack design slowed the Row Address Strobe (RAS) from the computer by a few nanoseconds to produce a special signal for the expansion box, MRAS. By using an inductance network, the design also produced delayed signals which were used for select multiplex (MMUX) and column address strobe (MCAS).

These alterations retain the original RAS, and the delayed MMUX and MCAS follow that. Normal speed on the computer is unaffected by these changes.

Survey Results

Well, the votes have been cast, few of them though there were (23 out of 50,000 readers!), and here are the results:

Everything's okay:	15
More hardware:	3
More software:	3
More of both:	1
Five page complaint:	1

Among all the software presented, only KBEEPFIX is in regular use; two readers built the real-time clock, one the small interface, and two performed a few suggested modifications. To the readers who responded, many thanks. To the 49,997 others. . . where did I go wrong?

Memory Sidecar Update: Apparently, initial shipments of this device from The Peripheral People have been delayed, but should be in your hands by the time you read this. Also, another company has developed an inexpensive ROM add-on, and several utilities to go with it. Write to Computer Accessory Technology, Frank Kwong, 1307 Bagley Dr., Kokomo, Indiana 46901.

Interview with Dr. Lirpa: Thanks and credit were inadvertently omitted last month. Appreciation to *Audio Magazine* for greasing the path to Dr. Lirpa's villa.

Next month (with luck): 384×192 high-resolution graphics add-on. Plug it in, turn it on.

EDUCATION 80

by Earl R. Savage

I have long maintained that teachers can learn to write instructional programs much easier than programmers can learn to instruct. That's another way of saying it is much easier to make a programmer out of a teacher than a teacher out of a programmer! I don't mean to sell programmers short—it's simply that extensive training and experience are necessary before anyone can become a teacher, and therein lies the problem with much of today's instructional programming.

It is a rare stroke of luck when a programmer turns out a good CAI (Computer Assisted Instruction) program. This is true even when he knows the subject well—he just doesn't know enough about how people learn or the most effective ways to present material to various types of students.

There is a very real shortage of teacher-programmers. When you consider that the best CAI writer is a combination of master teacher and expert programmer, where do you find one?

A second reason for scarcity of good CAI is financial. Writing a good program from scratch requires many hours; to be worthwhile, the writer must be able to anticipate a reasonable number of sales at a fair price. Schools, however, are increasingly short of funds: As they buy less, writers produce less.

The solution is to find some method by which teachers can write programs quickly without knowing anything about programming. Fortunately, such a solution is as close as your mailbox. There are writer's programs available which will take care of all the "computer stuff" and let you concentrate on the learning material.

Instructional Goals

Let's have a quick review of the fundamentals of planning an instructional lesson or topic. This is nothing new—it's the same process you've been using for years when planning instruction. Planning for computer instruction is exactly like planning for regular classroom instruction. The only difference is that it is easier to accomplish some important objectives with CAI.

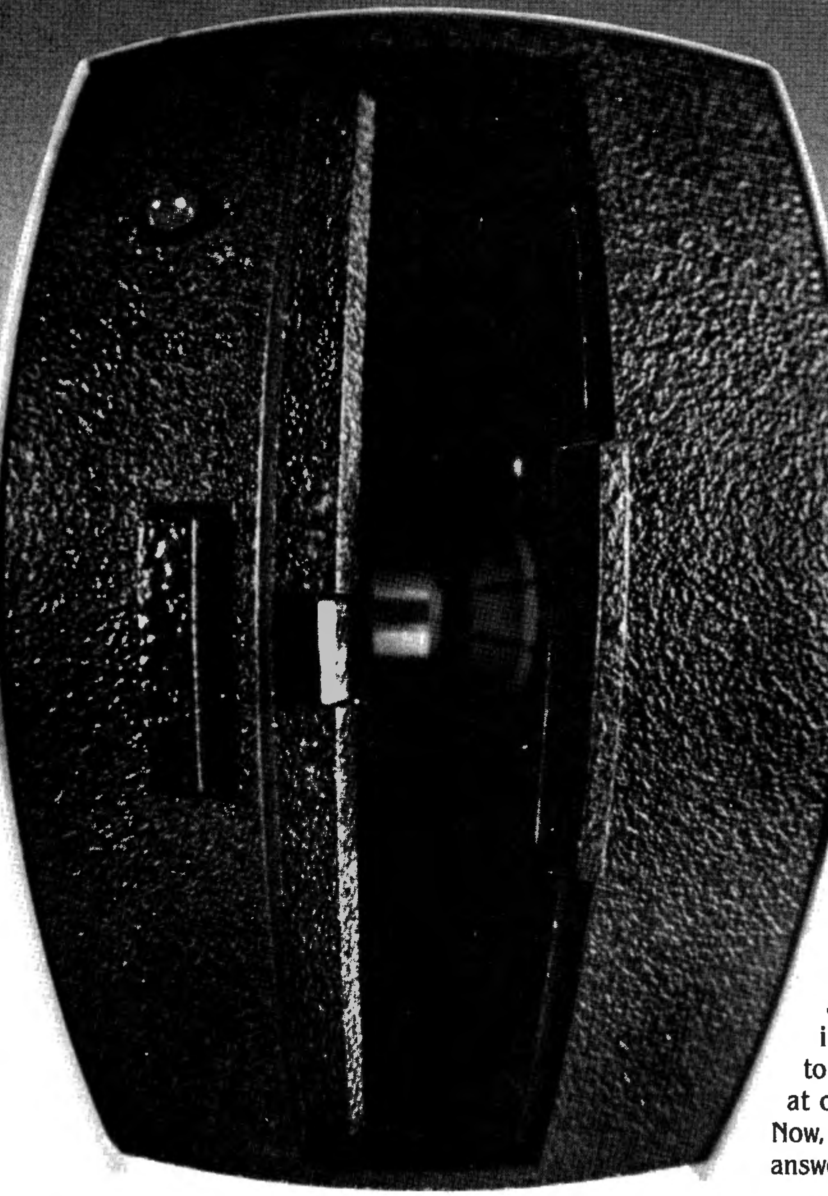
After setting goals and performance objectives for student achievement, select the instructional materials and processes which you expect will lead to attainment of those objectives. From a practical standpoint, you end up with a "plan of attack" (on ignorance) suited to the students, the material and the situation.

CAI Organization

The simplest type of CAI programming is organized in a linear manner. It is similar to the pattern used in the lecture/discussion presentation of material by the classroom teacher. Linear programming is represented by the diagram in Fig. 1.

APPARAT INTRODUCES

Now even more bytes per buck

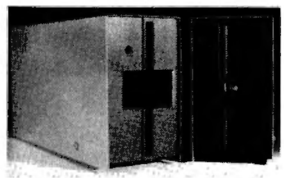


405 K/bytes of storage. Apparat has combined its Newdos/80 operating system and a dual-sided 80 track mini-floppy drive to give you 405,000 bytes of storage in a single volume. Modification patches to Newdos/80 expands the capability of single density drives, so you'll have greater applications for your TRS-80 model 1.®

Drives plug directly into an expansion interface with no modification required so you can now have over 1 megabyte of storage on-line with standard mini-floppy diskettes. Each drive has 316 free grans, for a total of 948, on a maximum of three 80 track drives, which can be added to a TRS-80.

Upgrading to double density is possible by running under most double density controllers, (e.g. Percom's Doubler).

These drives can "read" standard 35 or 40 track diskettes with special software that skips every other track.



Drives come complete with case, power supply, interface cable and documentation including patches to Newdos/80. The drive mechanism is priced at only $699 (Cat. No. 1-705, specify I or III). Now, at 579 bytes per buck, it just might be the answer to your storage problems.

Apparat, Inc.

4401 So. Tamarac Parkway, Denver, CO 80237 (303) 741-1778

VISA master charge

Toll Free Order Line
1-800-525-7674
(Except CA, AL, HI)
For Technical Service
303-741-1778

Now available for model III, 728K bytes of storage.

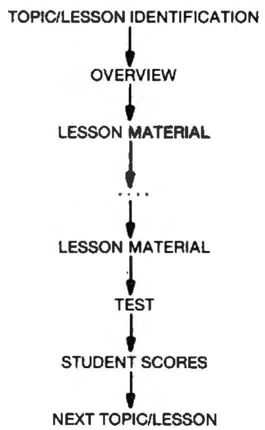


Fig. 1. A Linear Instructional Program

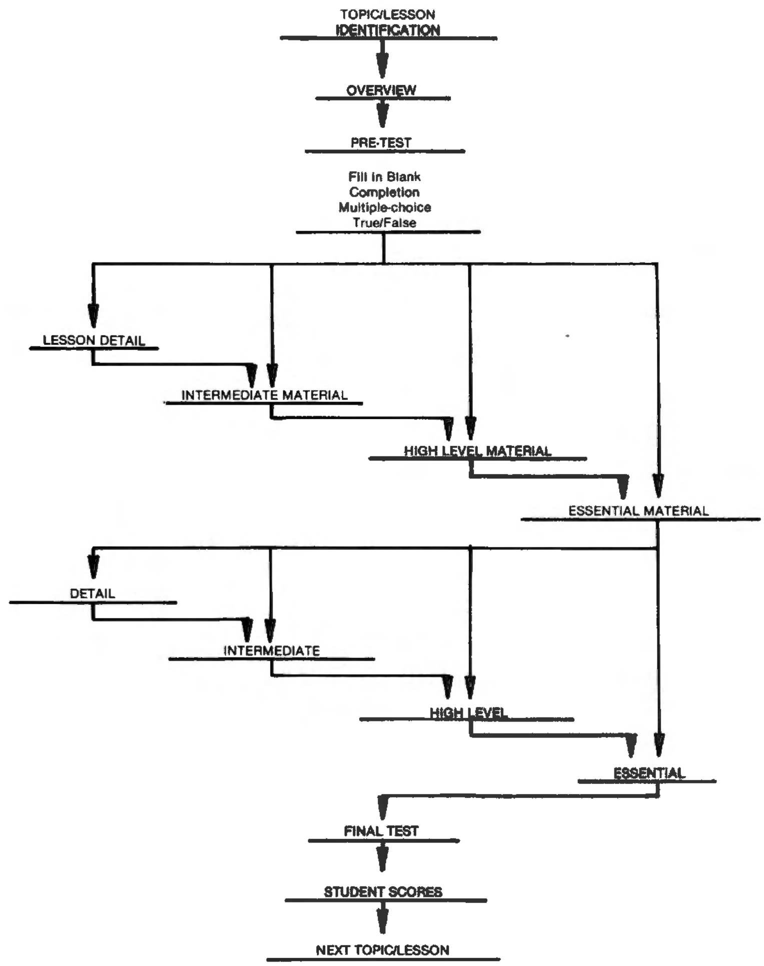


Fig. 2. A Branching Instructional Program

The program (or teacher) presents a series of informational frames which may or may not include a significant amount of student interaction. All students are exposed to the same material in the same sequence.

A more advanced type of CAI programming is called branching. In branching programs, all students are not presented the same material. They get only that which, by some criterion, they seem to need.

Fig. 2 shows one example of an almost infinite variety of branching possibilities. In this case, each student takes a "pretest" and then receives material which is appropriate in detail and scope to his demonstrated knowledge of the subject. A typical branching program contains frames of instructional material which are skipped by students in various categories. There may be as many levels as the teacher believes necessary, though seldom does one find more than four levels in a program.

The better your plans, the better your instruction. We will leave the subject of planning with the well known axiom that an effective instructional activity requires careful planning regardless of the means used to present that activity.

Easy Programmer

So, there you are with your lesson plan in hand, ready to write your CAI program. Even if your plan is linear, you must know something about programming that computer. If it is branching, you must know much more—*unless*... unless you have an "author" program to take care of the nuts and bolts of putting your plan into the computer.

Programs called CAIware and Super-CAI will handle those nuts and bolts for you. They are available from Micrognome, Fireside Computing, Inc., 5843 Montgomery Road, Elkridge, MD 21227. As amazing as it may seem, all the teacher has to be able to do is load the author program and follow simple instructions.

CAIware consists of three parts: an author program, a student program and a detailed manual. The author program is a manager-formatter, used by the teacher to create lesson tapes according to his own specifications. The student program is used by the student to call up the lessons. Both programs are re-used as often as desired; they are unchanged by the lesson presentation.

You, the teacher, need know absolutely nothing about programming to create top quality CAI lessons. The author program takes care of that automatically. You simply answer questions and fill in the blanks on pre-formatted screens.

I'm aware that this sounds like a straight-jacket, allowing little variation in lesson format. Nothing could be further from the truth. The CAIware author gives you a large number of choices throughout, allowing you great variety and latitude formatting each lesson to match the requirements of the subject and the needs of the students.

Those of you who are beginning or even advanced programmers should not overlook the value of CAIware and Super-CAI: you should consider the time and effort an author program can save you. If you don't have to slow down to figure out the nuts and bolts for each of your lessons, you can spend more time on the essential planning and end up with more good CAI programs for your students.

Don't wait for someone else to write CAI programs suitable for your students. You have already done most of the work—look at all those lesson plans in your file cabinet! I bet that with a little plan modification and some nuts and bolts assistance, you can be on the way. ■

THE EXCLUSIVE ORACLE

by Dennis Kitsz

Q: I have built your interface (April, May and June "Applications") and it works fine. How about another couple of chips that would decode seven of the data-out pins as a peripheral device's address. Then you could use the eighth address bit as your logic high or low signal. That way, couldn't one port control two to seven devices on or off?

Also, a disturbing thing with the interface so far is that with the ribbon cable connected to either the computer or the expansion interface, I get some kind of noise that appears on letters displayed from the keyboard. Would this disappear if I enclosed it in a metal case? Do I need to shield the ribbon?

Lenny Greenberg
Toledo, OH

A: The idea of controlling peripheral devices that way is a good one that I never considered. There's an old psychology term called "functional fixedness," and lots of us fall into that trap. Using data lines as an address is in that category. So here's part of a circuit you might use:

The 74LS154s select one of 16 devices when given a binary input; the three other lines can select one of eight 154s, giving a total of 128 options. (What are you running, Lenny, all the traffic lights in town?) The on-off signals can be run to the 5-volt relays as shown, or to optical isolators where you are working with higher voltage. (Maybe a train set?)

As for your noise problems, you might consider a shorter cable, braided cable, or shielding, but first check the cleanness of the wiring or soldering on the small interface. Also, if your expansion box does not use the buffered cable, any peripheral devices that connect directly to the edge-card bus increase the overall system noise. Tandy's technical handbook only rates the TRS-80's fan-out (the ability to drive additional circuits) as "one"! So clean the edge-card contacts, keep the TRS-80 away from noisy electrical equipment (fluorescent desk lamps are notorious), and connect peripheral devices only when you need them.

Q: Well, here I am, sitting at the good old TRS-80 using the Word Processor from May's *80 Microcomputing*. Getting the LNW Research Interface up and running has given me a problem. The real-time clock won't work, I have all chips in the board except for the disk controller. Any ideas?

John W. Schrage
Chester, New Jersey

A: Oddly, the disk controller is an integral part of the real-time clock in the LNW board. The chip can be obtained from Advanced Computer Products in Irvine, California, for about $17.

Q: Single-stepping a BASIC program (September "Applications") intrigued me, as it seems to be an excellent way to debug a program. I have a 48K TRS-80 with two disk drives, and use either TRSDOS 2.3 or NEWDOS+. I thought I would give the BASIC version of the single-stepper program a try. I cannot get beyond line 30, at which point my screen reads ILLEGAL FUNCTION CALL AT 30. Could you make some suggestions?

```
30 POKE 16526,228 : POKE 16527,127 : M% = USR(0)
```

Dr. George L. Haller
Naples, FL

A: Your disk operating systems provide you with ten USR options rather than just one we Level II users are allowed. That means you must define the entry points for each of these USR calls. In this case, the line would read:

```
30 DEFUSR1 = &H7FE4 : M% = USR1(0)
```

By the way, since you have a 48K machine, you might wish to relocate the single-stepper to higher memory.

Q: I'm having a lot of trouble with my TRS-80. First, when my computer scrolls, it prints spaces as zeros, Gs as Ws, and so on. It is very regular, and this does not occur until the computer gets to the bottom of the screen and scrolls. The last line is unchanged. Second, the computer is ignoring some cassettes, most of which I've loaded many times successfully. Either I get a stuck asterisk or no asterisk at all. Third, I've been getting some arbitrary characters on the screen, and often the computer will unexpectedly lock up. Other times the screen will fill with garbage and begin printing, from the bottom to the top, a pair of alphabetic or numeric characters. What's happening?

Dave Hildebrandt
Walnut Creek, CA

A: As you know by now, we solved these problems together. Here's what happened. Some solder hairs you left when you put in a lower-case mod were causing part of the random garbage and most of the lockups. Moving the CPU away from some nearby electrical equipment cleared them up completely.

The scrolling problem was more subtle. Bit 4 of the video memory seemed stuck high all the time, which would account for the very regular change in letters. Yet, since the screen data was printed correctly the first time (before scrolling), it

✓See List of Advertisers on page 306

would seem that the video memory itself was fine. What could this mean? We need to know something about the scrolling routine itself, which works like this:

```
LD      HL,3C40H
LD      DE,3C41H
LD      BC,37FH
LDIR
```

In other words, it transfers what it reads from the bottom 15 lines of the screen to the top 15 lines. Simple. But the secret is the statement, "what it reads from." This means bit 4 is fine going into the video RAM, but messing up on the way out. It could indicate a bad read line on the video RAM, a blown buffer, or a broken trace on the line (causing it to "float" high). In your case, the trace was torn during your modifications, and the problems cleared up when the trace was repaired.

Finally, your cassettes load now that you've replaced the ones damaged by your CTR-80 tape recorder. Radio Shack still offers the free fix to early CTR-80s, which could record a glitch onto tapes, even in the play mode.

Q: I am having a great deal of trouble communicating using my CAT modem. It works great with other TRS-80s, and in fact I've exchanged info from Spokane to Chicago. There is a Computerland store in my neighborhood that is left up all night. The employees swear other TRS-80 users are on the system, but I can't seem to get going. Are there any tricks to interfacing a TRS-80 and Apple? Baud rate? Stop bits?

Paul Knechtel
Hinsdale, IL

A: Since you've been successful on line, I won't suggest you've been putting the phone receiver in backwards! I did that myself the first time. Your question is one that plagues computer users from the home user to the systems engineer: what communications protocol to use?

For hobbyist use, you would normally be sending your data at 300 baud, 8 bits, with parity disabled. Use full duplex, and set your modem to the "originate" position. After you dial the number and the host computer answers, you should hear a whine pitched approximately a seventh higher than yours (like the distance between C and B on the piano). When it hears your answer tone, it may send some requesting signal, such as a question mark, but don't count on it. Hit ENTER a few times, and wait. Keep trying that, but if there's still no response, or the carrier tone disappears, dial up again. Go through the same process, but listen in on an extension. Listen for a variation or jitter in the pitch at the far end. It may not be sending data at all, or your local phone lines may be weak. As a last resort, ask for a list of those "other TRS-80 users," and give one of them a call.

There are no tricks to interfacing an Apple and a TRS-80. The reason we use ASCII, standard modem frequencies, and so forth, is to provide compatibility.

Q: I would like to leave the small Radio Shack amplifier/speaker connected to the second cassette cable and not have to do a lot of unplugging, etc., for music and sound effects. Is there a POKE that would direct the output to the second cassette jack?

Joe Sholes
South Boston, MA

A: The switching relay for cassettes in the expansion box is mapped to address 37E4. To switch to cassette #1 (from cassette #0),

```
POKE 14308,1
```

Get back by POKEing a zero into that location. If this doesn't seem to work, check the cables attached to your expansion box. Radio Shack didn't mark their positions, and the order isn't logical. Check the installation manual.

Q: All I want is a routine to have the computer tell me how many digits there are after a decimal point for any given value. My simple program doesn't do the job:

```
10  FOR I = 0 TO S
20  Y = ABS(A - FIX(A))
30  IF Y <= 0 THEN 60
40  Y=A*10 : A=Y
50  NEXT
60  PRINT "I = ";I:END
```

What's the solution?

Dick Blayney
Mission Viejo, CA

A: There are a few problems that come up when using decimal positions. First of all, single-precision numbers have fifteen. Try this routine, which will work in most cases:

```
 10  DEFDBL A
 20  INPUT A
 30  A$ = STR$(A)
 40  B = LEN(A$)
 50  FOR X = 1 TO B
 60  B$ = MID$(A$,X,1)
 70  IF B$ = "." THEN 90 ELSE NEXT X
 80  X = X - 1
 90  Y = B - X : PRINT Y
100  GOTO 20
```

Defining A as a double-precision number will give you fifteen decimal places before going into scientific notation. Following zeros will not be counted by this program.

Q: There are several things I do not understand about machine language programs written as BASIC POKE statements. For example, in the BASIC single-stepper (September "Applications"), you say key it in and run it separately. Do you mean RUN the program in Listing 2?

When you say delete the program, do you mean delete the program in Listing 2? Is it necessary to protect memory if no other system programs are in memory? And if so should it not be done at power up?

If you say, "always protect memory," I still will not know what you may be trying to communicate. Also, how can an assembly language program be converted to a BASIC listing?

MULTI-USER OASIS HAS THE FEATURES PROS DEMAND. READ WHY.

(THEN COMPARE.)

NOW AVAILABLE FOR TRS80 MODII

Computer experts (the pros) usually have big computer experience. That's why when they shop system software for Z80 micros, they look for the big system features they're used to. And that's why they like Multi-User OASIS. You will too.

DATA INTEGRITY: FILE & AUTOMATIC RECORD LOCKING

The biggest challenge for any multi-user system is co-ordinating requests from several users to change the same record at the same time.

Without proper co-ordination, the confusion and problems of inaccurate or even destroyed data can be staggering.

Our File and Automatic Record Locking features solve these problems.

For example: normally all users can view a particular record at the same time. But, if that record is being updated by one user, automatic record locking will deny all other users access to the record until the up-date is completed. So records are always accurate, up-to-date and integrity is assured.

Pros demand file & automatic record locking. OASIS has it.

SYSTEM SECURITY: LOGON, PASSWORD & USER ACCOUNTING

Controlling who gets on your system and what they do once they're on it is the essence of system security.

Without this control, unauthorized users could access your programs and data and do what they like. A frightening prospect isn't it?

And multi-users can multiply the problem.

But with the Logon, Password and Privilege Level features of Multi-User OASIS, a system manager can specify for each user which programs and files may be accessed—and for what purpose.

Security is further enhanced by User Accounting—a feature that lets you keep a history of which user has been logged on, when and for how long.

Pros insist on these security features. OASIS has them.

EFFICIENCY: RE-ENTRANT BASIC

A multi-user system is often not even practical on computers limited to 64K memory.

OASIS Re-entrant BASIC makes it practical.

How?

Because all users use a single run-time BASIC module, to execute their compiled programs, less memory is needed. Even if you have more than 64K, your pay-off is cost saving and more efficient use of all the memory you have available—because it services more users.

Sound like a pro feature? It is. And OASIS has it.

AND LOTS MORE...

Multi-User OASIS supports as many as 16 terminals and can run in as little as 56K memory. Or, with bank switching, as much as 784K.

Multi-Tasking lets each user run more than one job at the same time.

And there's our BASIC—a compiler, interpreter and debugger all in one. An OASIS exclusive.

Still more: Editor; Hard & Floppy Disk Support; Keyed (ISAM), Direct & Sequential Files; Mail-Box; Scheduler; Spooler; all from OASIS.

Our documentation is recognized as some of the best, most extensive, in the industry. And, of course, there's plenty of application software.

Put it all together and it's easy to see why the real pros like OASIS. Join them. Send your order today.

OASIS IS AVAILABLE FOR

SYSTEMS: Altos; Compucorp; Cromemco; Delta Products; Digital Group; Digital Microsystems; Dynabyte; Godbout; IBC; Index; Intersystems; North Star; Onyx; SD Systems; TRS 80 Mod II; Vector Graphic; Vorimex.

CONTROLLERS: Bell Controls; Cameo; Corvus; Konan; Micromation; Micropolis; Tarbell; Teletek; Thinkertoys; X Comp.

Write for complete, free Application Software Directory.

PLEASE SEND ME:

Product	Price with Manual	Manual Only
OPERATING SYSTEM (Includes: EXEC Language; File Management; User Accounting; Device Drivers; Print Spooler; General Text Editor; etc.)		
SINGLE-USER	$150	$17.50
MULTI-USER	350	17.50
BASIC COMPILER/ INTERPRETER/DEBUGGER	100	15.00
RE-ENTRANT BASIC COMPILER/INTERPRETER/ DEBUGGER	150	15.00
DEVELOPMENT PACKAGE (Macro Assembler; Linkage Editor; Debugger)	150	25.00
TEXT EDITOR & SCRIPT PROCESSOR	150	15.00
DIAGNOSTIC & CONVERSION UTILITIES (Memory Test; Assembly Language; Converters; File Recovery, Disk Test, File Copy from other OS; etc.)	100	15.00
COMMUNICATIONS PACKAGE (Terminal Emulator; File Send & Receive)	100	15.00
PACKAGE PRICE (All of Above)		
SINGLE-USER	500	60.00
MULTI-USER	850	60.00
FILE SORT	100	15.00
COBOL-ANSI '74	750	35.00

Order OASIS from:

Phase One Systems, Inc.
7700 Edgewater Drive, Suite 830
Oakland, CA 94621
Telephone (415) 562-8085
TWX 910-366-7139

NAME________________
STREET (NO BOX #)________
CITY________________
STATE________ ZIP________
AMOUNT $________
(Attach system description; add $3 for shipping; California residents add sales tax)
☐ Check enclosed ☐ VISA
☐ UPS C.O.D. ☐ Mastercharge
Card Number____________
Expiration Date__________
Signature____________

OASIS

MAKES MICROS RUN LIKE MINIS

I always see POKE, FOR. . . NEXT, READ, DATA, which can presumably be read by BASIC. Where is it put in a BASIC program? How do I get to it? How do I get back to the BASIC program? I hope my questions are not too much!

George Dosa
Gardena, CA

A: You've asked all the right questions. . . almost enough for a book! There is a key to understanding how these programs work: a computer contains memory, which holds whatever you tell it to hold. When you load a BASIC program, memory holds human-oriented words like PRINT, INPUT, READ, etc., which cannot be used by the computer to do anything. Instead, a large block of permanently programmed memory interprets these words and symbols into the electrical impulses the computer can understand. In this case we call the permanent memory "Level II BASIC."

Machine language programs merely avoid the roundabout and time-consuming process of interpretation. When you load and run a SYSTEM tape, you're avoiding that interpretation process. But remember this: when you load that SYSTEM tape, it doesn't disappear. It is held somewhere in memory, where it can do its work.

TRS-80 users with programs like T-Bug and Editor/Assembler can build their own SYSTEM tapes. You may not have these programs, or you may find it is more convenient to load a single BASIC program with a machine language subroutine. That forces us to use BASIC, clumsy as it can be, to simulate loading a machine language program into memory!

The process is surprisingly arbitrary. To start with, high memory is chosen because Radio Shack lets us set Memory Size, which is the amount of space allocated to a BASIC program. So when program authors tell you to set memory size to 32739, they mean "let BASIC use all the space up to but not including memory cell number 32739." You might think that a short BASIC program wouldn't necessarily need all that memory, would you? Well, the fact is that Level II BASIC always keeps an electronic scratchpad just below that Memory Size boundary. We've got to prevent BASIC from using our machine language program as its scratchpad!

Once space has been set aside in high memory, a FOR-NEXT loop can be created which POKEs the machine language program (READ from the lines of DATA statements) into that reserved block of memory. When the work of POKEing the data is finished, the machine language program is ready to be used; from then on it will be unaffected by any BASIC command because we have protected it. RUN, CLEAR, NEW, and all the rest leave it untouched.

Consequently, DELETE also leaves the machine language program untouched. Once the work of the FOR-NEXT, READ, DATA and POKE lines has been completed, they are no longer necessary. They may be removed from the program (they need not be, but a long BASIC program might need the space when it executes).

On to your last question: how is a machine language program converted to BASIC statements, and how do you use it? The process is not complicated, but you can run into trouble. First of all, it's best to try to convert only those programs which are intended to work with BASIC, but for which the author has neglected to provide a BASIC version.

Next you must determine three things: where the program starts, where the program ends, and where it is entered. The starting point is usually marked ORG (origin) in an assembly language program. In a hex dump, it is the address presented at the upper left in the block of hex numbers.

Finding the ending point can be a bit tricky. In an assembly language listing, the last address presented is the location of the start of the final machine language instruction... which may or may not be the last byte of the program. Since that final instruction might be one, two, three, or four bytes long, look at the listing carefully.

The start and end points thus represent the two bounds of your BASIC FOR-NEXT loop. Naturally, you will have to convert the boundary information to decimal; check back issues of *80 Microcomputing* to learn how to accomplish this task. The machine language program you will be converting is then found either in the block-memory dump or the second column of the assembly listing. Each of the two-character hex bytes must be converted to decimal; these become the long list of DATA statements that are POKEd into place in the above FOR-NEXT loop.

Finally, the entry address must be found and put into place. The entry point is usually described in a program's accompanying text, or will be notated as an END statement in an assembly listing. An address is two bytes long, so it can't be squeezed into a single POKE. Instead, the address is split into two halves, and each of those converted to decimal. For example:

```
7F00 is split into 7F 00
7F and 00 are converted to decimal 127 and 0
```

These values are POKEd into the USR entry locations, 16527 and 16526. The high byte (7F or 127 in this case) goes into 16527, and the low byte into 16526. Whenever a statement containing USR is executed (M% = USR(0)) or PRINTUSR(5), for example), the BASIC program jumps to the machine language program. If the program has been well written, it will automatically come back to BASIC when it is finished.

George, there are many ways of making machine language programs work with BASIC. Some use fancy tricks, but this is the most direct. Good luck. ■

Address your questions about the TRS-80 to Dennis Kitsz, Roxbury, Vermont 05669.

80 NEWS

edited by Bert Latamore

"Three major files of the Dow Jones Information service have been reformatted for compatibility with all TRS-80s."

Dow Jones Offers Videotex Tie-In

Up to the second business news and continuously updated stock quotes can now come directly from Dow Jones to you through your TRS-80 or Videotex.

Under an agreement with Tandy, the nation's largest publisher of business news is making three data bases of its electronic news service, the Dow Jones Information Service, available for all models of the TRS-80, including the new color computer and the Videotex monitor.

For Dow this represents a major step in making the Dow Jones Information Service, originally designed for brokers, securities analists and others in the financial field, available to the general public.

For Tandy it is a second major step in a move to make the TRS-80 a major source of news and information, a process which saw its first success in the introduction of the Compuserve service to the TRS-80.

Dow is so confident it is making available a free hour to all present owners and purchasers of Videotex.

Direct Pipeline

Dow has reason for its confidence. The package gives what is in many ways an investor's and businessman's dream, a pipeline directly into the heart of Dow Jones, publishers of *The Wall Street Journal* and *Barron's National Business and Financial Weekly*.

Three major files of the Dow Jones Information Service have been reformatted for compatibility with all TRS-80s. The first contains most items on the Dow Jones News Service, the nation's largest business news wire service, from 90 seconds ago to 90 days ago.

The second is a constantly updated listing of stock quotes for more than 6,000 businesses from the New York, American, Midwest and Pacific stock exchanges and the NASDAQ over-the-counter market.

The third is a file of detailed financial statistics on the 3,200 companies.

All these are arranged for access by company or by 60 subjects areas so you can see what is happening with any particular firm or get a view of overall activity in a particular market.

News stories hit the electronic news service 90 seconds after they go over the Dow wire, according to Tim L. Turner, director of marketing information services for Dow.

For instance, Turner said, two minutes after the story on the Dow-Tandy agreement was announced over the Dow Jones wire he read the story off his computer terminal.

That day Tandy stock rose several points, a clear demonstrations of just how sensitive today's market is to such news.

Because of the speed of the service, a TRS-80/Videotex user is essentially reading over the shoulders of the reporters and editors of the *Wall Street Journal* as they write tomorrow's front page.

The news wire maintains a record of stories for 90 days, so you don't read a story in a vacuum. Instead, you can review what has happened to this company and its industry by summoning a menu of headlines by firm or subject area and scanning them to find stories that relate to your present interest.

The stock quotation files are delayed 15 minutes because of rules imposed by the stock exchanges. The 80/Videotex user can get these by company or subject area and can specify from which exchange he wants the quote.

He can also get the full financial picture of any of 3,200 firms in the Dow list. This file is updated weekly.

Early Electronic Publishers

Turner said Dow started electronic publishing in 1974 as a service for professionals in the investment field.

In 1977 they started a time-sharing service on a dial-up basis, but it wasn't until 1978 that they tried marketing it to the general public through the Apple microcomputer.

By means of the agreement with Tandy, which, sells more than 50 percent of the total microcomputers sold in the United States and Canada, Dow is trying to reach a much larger market with services that have proven to be popular, Turner said.

Tandy spokesman John Shirley, on the other hand, said Tandy's interest in the non-exclusive agreement is to make a new

Journal Not Going Electronic

The publishers of the prestigious business press, *The Wall Street Journal* and *Barron's* among others have joined a growing number of newpapers entering electronic publishing.

Dow Jones Spokesman Tim L. Turner said Dow's recent agreement with Tandy Corporation which puts the Dow Jones Information Service on Videotex is a move to make the service more available to the general investing public.

This public will be able to read many *Wall Street Journal* stories over their TRS-80s before *The Journal*, itself, can print them.

However, Turner said Dow Jones does not expect this to have a major impact on *The Journal's* circulation. He said Dow has no plans to turn *The Wall Street Journal* into an electronic newspaper.

Rather, he said, the two news services are seen as serving different, although complementary, needs.

"People use the electronic service and still like to read *The Journal*," he said. ■

and unique information service available to owners of TRS-80s.

Tandy-Radio Shack is basically a hardware producer in this area, he said. The firm hopes to promote its microcomputers and Videotex terminals by making their service more valuable.

Turner said the Videotex version of the Dow service is intended mostly for the small businessman who uses a microcomputer and for the individual investor.

He said in many cases the costs involved are tax deductable, and he said that the service is low cost.

The Dow service is compatible with the Videotex without any change in hardware. The major change to the three databases involved fitting them to the 32 characters across format on the Color Computer.

"A TRS-80/Videotex use is essentially reading over the shoulders of the reporters. . . of the Wall Street Journal."

Instructions and Costs

All the investor needs is the instruction books and his password. The books and a special password good for an evening hour's free time on the service are being included in the $29 Videotex package. Passwords and instructions are to be made available to those already owning Videotex at no cost.

To use the system, you simply call your nearest Tymnet telephone number. Several pages of these numbers, covering cities across the US and Canada ranging from Los Angeles to Moose Jaw, Saskatchewan, are listed in the Dow instructions. For most users the wire service is a local call.

You identify yourself, your equipment and the information you want by a series of codes detailed in the instructions and by a password.

The instructions contain all the codes you need including more than 100 pages of codes for individual firms, mutual funds, private and government bonds and foreign bonds.

The service is being offered to the general public at a different rate than that given to major investment firms. The $2.50 minimum charge those firms must pay has also been dropped.

The cost of the service depends on the database used and the time of day. Costs for non-prime time use, after 8 pm local time at the locale of the user, is much lower than prime time use except for the statistical and financial information, which costs a dollar a minute at all times.

The news service costs a dollar a minute during prime time and 20 cents a minute after 8 pm. Stock quotations cost 75 cents a minute during prime time and 15 cents at night. The cost of the Tymnet telephone system is included in these charges.

There is no start-up charge or minimum charge imposed, Turner said.

The service is available on weekdays from 6 am to 3 am EST daily. On holidays and weekends, which count as non-prime time all day, the system operates from 7 am to 3 pm EST, Turner said.

Dow does not charge extra for the privilege of making private copies of any information available in the databases, so you can store or print anything that interests you.

Turner said the costs, which may appear high at first glance, are actually very reasonable for the kind of service Dow is offering when compared to other electronic news services such as Compuserve or The Source.

This, he said, is because these other services are not designed to give you information by company or by stock or by business subject. To get such information from these sources you have to spend a great deal of time searching for it, and since the cost of the information is your search time, times the rate per minute, the charges increase.

Even at that, some of the information, such as the most recent market quotes, may not be available from the other services at all, he said.

"As you read this, someone may well be checking his stock portfolio from his study or office over his TRS-80."

As of press time the service was nearly ready for release to the general public. The only holdup seemed to be final printing of the special users guide for owners of TRS-80s and distribution of the guide to Radio Shack stores.

Officially both firms have promised the service to be available sometime in the "second quarter." Turner said he expected it to be available this month. So, as you read this, someone may well be checking his stock portfolio from his study or office over his TRS-80. ■

Bert Latamore
80 Staff

Coleco Patents Football Game, Seeks Court Block against Tandy

Coleco Industries, Hartford, CT, recently granted a patent on their hand-held version of electronic football, is now instigating suits for infringement against Tandy-Radio Shack Corporation, Fort Worth, TX; Mattel, Inc., New York, NY; and several other electronic game manufacturers and distributors.

Coleco's attorney, Michael Schwefel, said the suit is a straightforward action and does not set a precedent in the area of protecting software. He said the Coleco patent is for a physical device, not for a program.

Coleco is seeking an order stopping the game companies from what it says is an infringement of its patent and is seeking monetary damages based on a percentage of the total sales of football games by each offending firm. He said he did not know the exact amount sought against Tandy-Radio Shack.

Coleco maintains that aspects of its game allowing passing and blocking and allowing two people to play it interactively are being infringed on by other products in the field.

Schwefel said, however, that Coleco's patent does not prevent other firms from producing electronic football games.

This is not the first patent for an electronic game to be granted, Schwefel said. In fact Coleco is suing Mattel over a previous patent Mattel holds, arguing that that patent is invalid.

Attorney Gary Pat of Tandy's legal office declined to comment of the suit beyond acknowledging that Tandy-Radio Shack was named as a defendant. ■

Students to Work for Computer

Elementary school children across the country will have the opportunity to do something to improve their own educational opportunities this fall thanks to a link-up between Tandy Corp. and the QSP division of Readers Digest.

Under the new agreement children will sell magazine subscriptions to their parents to earn credits their school can use to buy TRS-80 hard and software.

"If I could dream up 12 good programs like this today, we'd put them into effect tonight."

QSP, Inc., Vice President William E. Drake emphasized that children would not be selling door-to-door and that the list of more than 300 magazines QSP offered would all be in "good taste."

"We aren't selling *Playboy* or *Hustler*," he said.

The sales effort will be sponsored locally by PTAs or similar groups, Drake said. The school systems who go for the program will distribute QSP literature and subscription forms to the students' homes through the children.

Credits for Hardware

QSP would return 50 percent of the gross magazine sales to the schools in the form of credits which could be used to buy any TRS-80 hardware or educational software at a discount.

"Basically, we're hoping the parents will buy their magazines through us instead of from someone else," Drake said.

Charles Phillips, vice president for special markets for Tandy, called the program "a dynamite idea.

"These people (QSP) have been in the business a long time, and they recognize the interest schools have in computers and the funding problems many schools have," he said.

Phillips said the program was not targeted at any particular portion of the educational market but was "another way for schools to get computers."

He said he hears daily from school systems who are interested in putting TRS-80s in their classrooms but lack the necessary funds.

"The schools are hard pressed," he said. "We have no way to help them find funds."

Drake said this was a general marketing program not connected to Tandy's educational department.

"We're always open to marketing opportunities," he said. "If I could dream up 12 good programs like this today, we'd put them into effect tonight."

Phillips said Radio Shack was giving the program "a standard volume discount" on products, but declined to give specific figures. They were determined in an agreement signed between Tandy and QSP.

Drake, however, indicated the discounts were substantial. Asked how many magazine sales a school would need to earn one computer, he said it would take about $1,200 in gross sales or about 120 subscriptions.

This would price the computer at about $480 in sales credits.

"Basically, we're hoping the parents will buy. . . from us."

Pilot Program

Drake said a pilot program was starting now in the midwest but that the "big roll out won't be until next fall."

QSP is the largest fund raising magazine subscription sales organization in the country and often works with middle school and high school students earning funds for class trips and other projects through magazine sales.

Both Drake and Phillips said the QSP agreement does not involve any other division of Readers Digest, and both specifically denied any involvement between Tandy and "The Source," A Readers Digest division that distributes information via a computer network to microcomputers.

However, Phillips said the present agreement came out of a meeting between himself and representatives of eight Readers Digest Divisions.

"They have a big educational department, also, which was our main interest in that meeting, I think," he said.

Readers Digest also has been involved heavily in market research for a variety of corporations for many years and has maintained a large installation of mainframe computers for this and other business and financial research for more than a decade.■

by Bert Latamore
80 *Staff*

Tandy Sues Personal Micro, Alleges Patent Infringements

Tandy-Radio Shack Corporation, Fort Worth, TX, is suing a rival microcomputer hardware distributor for patent infringement.

The suit, against Personal Microcomputers of Mountain View, CA., and five manufacturing and distributing firms, alleges the input/output programming of the PMC-80 microcomputer is copied from the TRS-80 and that the PMC name is confusingly similar to the TRS-80 registered trademark owned by Tandy.

Tandy is asking for unspecified damages and permanent injunctions against Personal Microcomputers, Eaca Electronics Corp. of America, Compumart, Consumer Computers Mailorder and Robitaille & Sons Enterprises, Inc. (The Software Exchange), and the Hong Kong based Eaca International Ltd., apparently the manufacturer of the product.

The PMC-80 is also being sold in Europe, and Australia under the names Video Genie and TRZ-80.

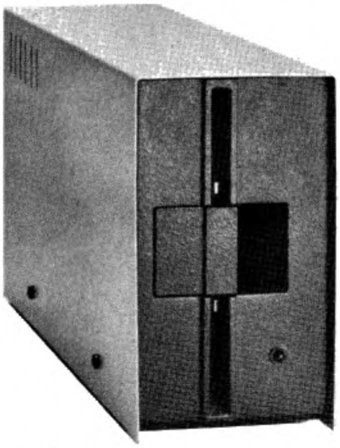
The PMC-80 is fully compatible with virtually all software designed for the TRS-80.

Officials for both Tandy and Personal Microcomputers declined to comment on the suit, which is entered in U.S. District Court in San Francisco, CA, although the spokeswoman for Personal Microcomputers said a statement might be released soon. ■

Mini-Disk Storage Systems for TRS-80* Computers

Access... Offers More. Charges less.

from $295



Company/Drive Model	Price 1/	Flippy	*Formatted* Diskette Storage Capacity: Dbl-Density	Sgl-Density	100% Tested	48 hr. Burn-In	Warranty	Trial Period
40-TRACK DRIVES								
Access Unlimited								
AFD-100[1]	$295.00	no	180 Kbytes	102 Kbytes	yes	yes	yes	yes
AFD-100F[1]	329 00	yes	360 Kbytes	204 Kbytes	yes	yes	yes	yes
MTI TF-5	359 00	no	?	?	?	?	?	no
Midwest Comp. & Per MPI.B-51	321.00	no	?	102 Kbytes	?	?	yes	no
Aerocomp Mdl 40-1	349 95	yes	?	?	yes	?	yes	yes
CPU Shop CCI-100	314.00	no	?	102 Kbytes	?	?	yes	no
AMI 40-track	325 00	no	?	?	?	?	?	no
80-TRACK DRIVES								
Access Unlimited								
AFD-200[1]	429 95	no	368 Kbytes	205 Kbytes	yes	yes	yes	yes
AFD-200F[1]	449.95	yes	736 Kbytes	410 Kbytes	yes	yes	yes	yes
MTI TF-8	639 00	no	?	200 Kbytes	?	?	?	no
Aerocomp 80-tk mdl	459.95	yes	?	?	yes	?	yes	yes
CPU Shop CCI-280	429 00	no	?	204 Kbytes	?	?	yes	no
AMI 80-track	560 00	no	?	?	?	?	?	no

1 As advertised in 80 Microcomputing, Jan 1981.

Check this line-by-line comparison of Access 5-inch mini-disk systems for Model I and Model III computers. *Nobody* offers more benefits, better service or lower prices.

TOLL-FREE ORDER NO.
1-800-527-4196†
(orders and literature only)

Free trial offer

Use your brand new AFD drive system for up to 15 days. If you're not *completely* satisfied, pack it in the original shipping container and send it back to Access. We'll refund the price of the system (less shipping charges), no questions asked. (No refund for misuse or improper handling.)

Operating burn-in test — too

Every drive that leaves Access is not only 100% electrically tested and double-checked for mechanical alignment, but it is also given a full 48-hour operating burn-in test. You'll find a test list — checked and signed by one of our competent technicians — in the drive carton. If a drive has latent defects, the burn-in life test will weed them out. The drives we ship just keep on running. And running. And running.

About our warranty

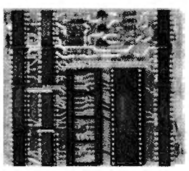
Venus de Milo has about the right number of fingers to count our warranty returns. Nevertheless, your new drive system is covered by our comprehensive 90-day limited warranty. The details are spelled out in the illustrated users manual included with each AFD drive system.

Software galore!

Games. Business programs. System software. We sell many of the best TRS-80* Model I and Model III programs, and at competitive prices. Make Access Unlimited your one-stop shopping center for all of your TRS-80* software, hardware and accessories. Save big! Call our toll-free order number, 1-800-527-4196† for free descriptive literature.

▶ USE YOUR CREDIT CARD AND SAVE! VISA AND MASTER CARD CHARGES ARE NOT DEPOSITED UNTIL THE DAY YOUR ORDER IS SHIPPED. CALL NOW TOLL-FREE, 1-800-527-4196†. ◀

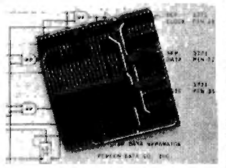
the DOUBLER™



Percom's new plug-in adapter for your Expansion Interface stores almost twice the data on a diskette track as a single-density system. You can store up to four times more data — depending on the type of drive — on one side of a diskette than you can store using a standard Model I mini-disk drive. Other features: Reads, writes and formats *either* single or double density minidiskettes. • Runs TRSDOS*, NEWDOS+‡ Percom OS-80™ or other single-density software *without* changing either software or hardware. Switch to double-density when convenient. • Includes DBLDOS™, a TRSDOS* compatible double-density operating system. • Includes on card, high-performance data separator circuit. • Installs *without* rewiring or trace cutting. • Introductory price, including DBLDOS and format conversion utility, only $219.95.

▶ *Permits Model III* software to be read on Model I computers. ◀

DATA SEPARATOR™



This PC board plug-in adapter for the TRS-80* virtually eliminates data read errors (CRC error — Track locked out!) which occur on high-density inner disk tracks, a problem that has plagued TRS-80* systems. The Percom Data Separator™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty: $29.95.

Disk System Interconnecting Cables

Improvement over RS cable design places drive 0, which includes the cable termination, at the end of the cable to eliminate the reflected noise of an unterminated cable. Better data integrity. Prices:

Two-Drive Cable $24.95
Four-Drive Cable 34.95

Power Line Filter

115/250 V, 50-400 Hz. Instructions included for easy installation in standard mini-box chassis: $19.95

Minidiskettes (Double-Density rated)

10 Disks in a convenient plastic organizer box $34.90
Single Disk 3.49

Disk Drive ID Tabs

1" x 1¼" self-adhering plastic drive identification tabs. Compatible silver with engraved black drive number. Two tabs (Nos. 0, 1): $2.50; three tabs (Nos. 0, 1, 2): $3.25; four tabs (0, 1, 2, 3): $4.50.

How to Order Order by calling Access Unlimited toll-free on 1-800-527-4196†. Mail orders also accepted. Orders may be charged to a VISA or MasterCard account or paid by a cashier's check, certified check or money order. We accept COD orders with 25% deposit. Sorry, we cannot accept personal checks. We pay shipping and insurance charges on orders over $1,000.00. Add approximate insurance and shipping charges for under $1,000.00. If in doubt about these charges, ask when you call in your order. Texas residents include 5% sales tax. Minimum order $20.00. Allow 2 to 4 weeks for delivery. †Texas residents call (214) 494-0206.

ACCESS UNLIMITED ✓229

315 N. Shiloh · Ste. D1 · Garland, TX 75042
(214) 494-0206

T trademark of Access Unlimited Company. *RADIO SHACK and TRS-80 are trademarks of Tandy Corporation. ™trademark of Percom Data Company, Inc. ‡trademark of Apparat Corporation.
ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE AND ALL OFFERS SUBJECT TO WITHDRAWAL WITHOUT NOTICE.

Author to Make New Microcomputer

In a change-of-state typical of today's micro industry entrepreneur, author/publisher Adam Osborne has become a computer hardware manufacturer. His metamorphosis has resulted in the Osborne Computer Corporation, headquartered in Hayward, CA.

Osborne will manufacture a portable, Z-80A based microcomputer called, predictably, the Osborne 1. The unit will sell for approximately $1600.

This single board midget micro conforms to airline carry-on baggage requirements and weighs only 28 pounds, including monitor. It offers a fast Z-80A processor from Zilog, 60K bytes of usable RAM, both IEEE-488 and RS-232 interfaces, modem electronics and a CP/M operating system. Two single density, 5¼-inch on-board drives are standard and double density drives usable with double sided disks optional.

A five-inch monitor with a 24-line by 50-character display format is also standard. A nine-inch monitor is optional. The character set of the Osborne 1 supports both upper and lower case as well as character underlining. The video display uses 4K bytes of RAM.

Author Osborn has not ventured into hardware manufacturing totally on his own. His corporate board of directors is made up of industry veterans, who have bought into the action. They include: Jack Melchor, a venture capitalist; Gary Kindall, designer of CP/M; Gordon Eubanks, designer of CBASIC; and Seymore Rubenstein, designer of Wordstar.

It is no coincidence that the above products are included in the Osborn 1 package. At the time of purchase the machine is supplied with a CP/M operating system, CBASIC, a Wordstar word processing program and two business programs, Mail Merge and a spread sheet program similar to Visicalc.

No In-House Programming

At present, software support for the Osborne 1 is limited to the programs supplied. Osborne Computer has no in-house programming staff and no plans to create one. Additional software support is expected from outside vendors who will be encouraged to write for the Osborne 1 system by an incentive program, the details of which were not available at press time.

In an interview with *80 Microcomputing*, Barbara Burdick, a marketing representative for Osborne Computer, indicated that several authors were already in touch with the company in hopes of writing software for the machine. In her opinion there will soon be no shortage of software.

Burdick was reluctant to comment on other aspects of the operation, however. When asked about the physical plant, she would say only that the final assembly is performed at the Hayward facility. Component level construction and chassis wiring may take place elsewhere, possibly the far east. When asked how many employees the facility had, she offered, "no comment," as she did when queried about the projected first year production run. She did say that McGraw-Hill, owners of Osborne's publishing operation, was not involved in the hardware venture.

The Osborne 1 will be required to meet the new RFI specifications the Federal Communication Commission has adopted for all microcomputing devices. Since it is being marketed for business environments, however, it will fall under class A limits for power-line conducted and radiated emissions. Certification is expected by mid-April.

While not an immediate threat to Radio Shack, the Osborne 1 microcomputer will have a measurable impact on the Shack's market. It is a carefully thought out machine that brings a level of sophistication to the micro marketplace that is unavailable from other manufacturers in the under $2000 price range.

But will the software, essential to its success be written in time to ensure the survival of the Osborne 1, or, like the Onyx, Sharp and Texas Instruments machines, will it fall victim to software starvation?

by Chris Brown
80 Staff

See last page of ad for ordering information.

SSM SOFT SECTOR MARKETING, INCORPORATED ✓494

& Victor Andrews

present

Quality Products That Will Make A Believer Out Of You!

POSTMAN DATA HANDLER

By Alger Software. ©1980 S.S.M., Inc. A machine language mailing list program that will do the following:

- 650 labels on a 35 track disk drive
- 750 labels on a 40 track disk drive
- 1534 labels on an 80 track disk drive
- 10 fields (2 user defined)
- Fast sorts (500 records in 30 seconds)
- Written for (1) drive system (capacities shown are for a single drive system)
- Most any label stock that is one up can be used (you save no money buying labels that are more than one up anyway)
- Print one label at a time or a sequence of labels
- Purge duplicates with or without user assistance
- 9 digit zip code
- Super fast search on any field - 3 second average
- Easy screen editing

Now Postman Data Handler has been upgraded with many new features. Now this very popular mailing package is not just the best way for most people and small businesses to do their mailing lists, but now we give you a way to uncramp, convert and formletter your mailing list. You need this package if one of the following is true: If your mailing package is memory dependent. This means that you must have more memory to handle more names in your machine; If you need a way to get rid of duplicate names in your mailing list; If you want to sort on more than just name or zip. (our package can sort on any or all 10 fields at once.); If you are waiting more than one minute for your sort to finish. (our package is in machine language so it runs very fast); If your present program will not handle the 9 digit zip code; If your program doesn't have full screen editing. This package is a machine language program, this is the reason for the super fast speed of all functions!

This is a random access disk based program and any name can be called to read, write, print or update in 3 seconds or less. Now along with it you get utilities that permit you to do the following: CONVERT 1; takes all the files from your old mailing list and converts them to our system. (Why should you change to our system if we made the change hard?) CONVERT 2; convert from our package back to ASCII files if you want to do something with them (like send them to another computer over the phone). MOD I (coming soon for the MOD III). This program runs on all quality operating systems. Requires 1 disk drive ad 32K memory.
.... only **$125.00**

Coming soon – MOD III version

MAILING LABLES 5000 .... **$14.95**
Plus $2.00 shipping

POSTWRITER

©1981 S.S.M., Inc. Now there is, at extra cost, a formletter package that permits inserting any of the 10 fields of imformation from "Postman Data Handler" into any part of a letter, (yes even in the body of the letter), and right and left justify the letter. This program is made to be used only with the Postman program and one of the following word processing packages: Lazy Writer or Electric Pencil.
.... **$49.95**

MINI POSTMAN

Sample Package. For you people who have a small mailing list (max. 26 names) and/or you can't believe that Postman is the best package and we must be overstating how well this program runs; try our sample package called Mini Postman. It comes as an up and running program and shows you how good our package really is. This package can be upgraded to the full package within 30 days of purchase by calling and saying "I want the full package". and paying the difference between the mini and the full package. Note: Mini Postman does not include the convert utilities. If not everything you expect, return sample disk for full refund (less shipping). You can't lose.
.... **$25.00**

Special

If you purchase both Postman and Lazy Writer at the same time, at list price, (or if you already own one of the packages above and now want to purchase the other), we will at no extra cost include the Postwriter package FREE. There are 2 ways to purchase this package. The first way is to say "I want the best package on the market so send me the full system at one time". This includes the Postman Data Handler, Convert 1 and Convert 2. The package works on all quality operating systems based on the Tandy disk format (TRSDOS, VTOS, NEWDOS, NEWDOS-80, etc.)

$250.00

This price is only good till June 1, 1981.
Save **$49.95**

If purchased separately **$299.95**

GRAPHX

By Steve Skindell. ©1981. This is a program that is for the person who does reports or requires some sort of plotted output to show gains or losses, or any type of output that needs graphs. This program puts to the screen or to a printer the plotted points in bar graph format and plots the mean, and averages of input. This is a very important program for accountants, CPA's and the average businessman to evaluate, at a moments glance where he is, was, or where he is going. Files saved to disk can be recalled at any time to be reexamined, modified, or just reprinted. An extra feature: if you have the Microline 80 printer, by Okidata or Epson MX-80, your output is in true graphics. Information is supplied for the user so he can modify this program for other printers. Comes complete and ready to run. Requires MOD I or MOD III, 48K disk. Printer optional (132 col.) ....
only **$49.95**

SPECIAL – Receive this program for only **$25.00** if you purchase an Okidata Microline 80 or 82 this month from us. Call for our low, low price.

HARDWARE

MOD III Disk Drive Installation Kit

This is the complete kit to install your OWN Disk Drive into your own MOD III, with household tools. Complete installation manual.

The only thing you provide is your MOD III, your disk drives, and any MOD III operating system. About one hour installation time. **Shipping included.**

$442.00

A.M. ELECTRONICS, INC.

See last page of ad for ordering information.

NEW PRODUCTS

edited by Bert Latamore

Read the Stars By Computer

Three new programs for TRS-80s will read the future according to astrology.

The Astro-Scope gives individual astrological readings of 1,500 words or more for $30 for the screen version or $200 for a printout version which includes a license to reproduce textual material commercially.

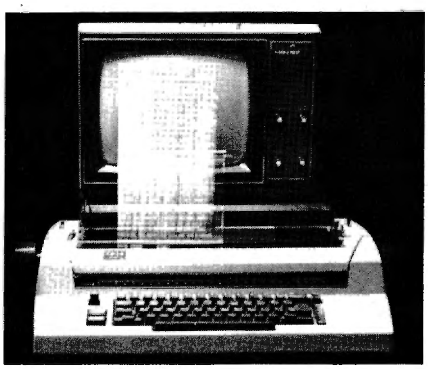
The Sex-O-Scope also does a full astrological readout, but it interprets it from a sexual viewpoint. Cost is $30 for screen or $200 for a printout version with commercial license.

The AGS-1 Natal Horoscope gives a complete calculation of a natal horoscope for use by professional astrologers. Priced at $125, it must be special ordered as the supplier tailors the program to the buyer's system.

Also available is the AGS Machine Language Sort which sorts 1,000 items in about four seconds at a cost of $50 for end users and $300 with commercial license.

All are available from AGS Software, Box 28, Orleans, MA 02653.

Reader Service ✓172



Mediamix Interface

Teach Your Micro to Type

The ETI² is a new, sophisticated interface that will connect any IBM Electronic Typewriter Model 50, 60 or 75 to any computer with a standard parallel printer port.

The Z-80 controlled interface has 2000 bytes of RAM and total access to all the typewriter's automatic functions. It allows you to adapt most word processing programs to the typewriter without modifying the program.

Information and prices are available from Mediamix, PO Box 67B57, Los Angeles, CA 90067.

Reader Service ✓342.

Education Catalog Printed

Queue Catalog IVB lists educational software from 70 publishers for the TRS-80 and Pet organized by computer, subject area and grade level. Catalog IVA will list the same information for Apple, Atari and Compucolor.

The catalogs are $8.95 each from Queue, Inc., 5 Chapel Hill Dr., Fairfield, CT 06432.

Reader Service ✓350

Guide Lists 100 Commands

A 10-page quick reference guide to the Oasis software system lists more than 100 commands derived from the 1000 page user's manual.

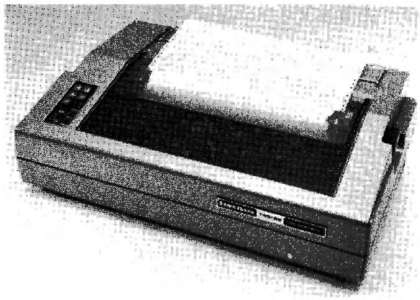
The guide lists major commands for single and multi-user systems, EXEC interactive job control language, a general purpose Text Editor and others. It is available for $1 from Phase One Systems, 7700 Edgewater Dr., Suite 830, Oakland, CA 94621.

Reader Service ✓346

New TRS-80 Line Printer Designed for Business Use

The new high speed TRS-80 Line Printer V from Radio Shack is designed for heavy business use.

It has a bidirectional, logic seeking, dot matrix head that prints 7×9 upper and lowercase characters 132 columns wide. It handles 5, 7.5, 10 or 15 characters per inch, 26 European characters and 30 graphics patterns. Print speed is 160 characters per second, 60 lines per minute.



TRS-80 Line Printer V

The printer is priced at $1,860 and is available exclusively at Radio Shack stores.

Reader Service ✓174

Program Teaches Fractions

Fractional Sound is a new program for TRS-80 Models I and III that teaches fractions to students in Grades 4 through 10.

It features seven parameters in either study or test mode, a self-relocating imbedded machine language sound routine, high speed graphics and the character Marley Mole. It requires 15K RAM bytes.

This and two other teaching programs are available on cassette for $14.95 each and on diskette for $18.95 for one, $24.95 for two and $29.95 for all three from Innovative Penquin, 2320 Hampton Dr., Harvey, LA 70058.

Reader Service ✓348

Cables, Programs Offered

E.B.G. & Assoc. have announced two new video connectors for the TRS-80 Model I and two new programs.

The Video Y Connector at $19.95 allows

one computer to run two videos, and the Video Extender Cable at $23.95 allows the TRS-80 video monitor to be moved to a remote position.

The Program of Lists at $19.95 stores up to 20 different lists, and the Pentad Disk Library at $19.95 maintains a library file of up to 100 diskettes.

Information is available from E.B.G. & Assoc., 203 N. Wabash, Suite 1510, Chicago, IL 60601.

Reader Service ✓168

Mod-II Demo Package Offered

The Small Busines Systems Group (SBSG) is offering a demonstration package of TRS-80 Model II software including a catalog, demo diskette and a manual including system reports for dealers and customers.

The demo diskette provides visual examples of various SBSG business programs with a variable speed capability to allow the salesperson time to discuss each example.

The package lists for $50 and the manual is available separately for $25 from Small Business Systems Group, Inc., 6 Carlisle Road, Westford, MA 01886.

Reader Service ✓330

Dirindex Indexes User Files

Programs Unlimited's Dirindex will index your programs, keep track of the size of your file in grans and show the space left on the diskette by side using a Model I disk system.

Dirindex does not rely on any particular DOS system and offers easy deletion of individual files or entire diskettes and full search capability. It will list files to screen or line printer organized in alphabetical or numerical order.

It is available for $19.95 from Programs Unlimited, Inc., 125 South Service Road, Jericho, NY 11753.

Reader Service ✓328

New Disk Drive Announced

A new disk drive for the TRS-80 Model III priced 29 percent below the Tandy-Radio Shack drive is being sold by VR Data.

The new drive plugs in and comes with a 90 day, 100 percent parts and labor guarantee and an extended warranty.

The unit, with one mini-disk drive, power supply, controller and mounting hardware, retails at $599 from VR Data Corp., 777 Henderson Blvd., Folcroft, PA 19032.

Get Data Input At Low Price

A new input interface allows a TRS-80 to read temperature light levels, pressure and voltages and generate a microMAC bus for low cost.

The module allows remote control of lamps, motors, pumps, heaters, appliances, communications devices, etc.

Offered by Connecticut Microcomputer Inc., 34 Del Mar Drive, Brookfield, CT 06804, the TRS-80 mod retails at $59.50 by itself or with a 16 channel AIM16 Analog Input Module, cables and power supply for $295.

Reader Service ✓327

Soundproof Quiets Printer

Dataroyal is offering optional soundproofing for its IPS-5000 intelligent matrix printers that, in an independent test, reduces the noise level of the machines to 63 decibels from the 66 decibels achieved with the standard package.

The option is available at $25 from Dataroyal Inc., 235 Main Dunstable Road, Nashua, NH 03060.

Reader Service ✓161

Medic Package Fills Forms

A new medical office program sorts and keeps patient and master files, maintains billing records, prints reports on Medicare and Blue Shield forms and office invoices, maintains a daily patient list, prints on Medicare and Blue Shield forms, prints a mailing list and even performs word processing.

Compatible with the TRS-80 Model I and II, the system also creates and maintains a literature reference system and keeps track of bank checks.

It is available under an introductory offer at $550 from LaSalle Computing Inc., 1547 DeKalb St., Norristown, PA 19401 until May 15. After that the price will go to $5,500.

Reader Service ✓333

Microconnections

Modems Give Phone Linkup

Microconnection modems will connect TRS-80 Model III and Color Computers with any time sharing system through the public telephone system, according to Microperipheral Corp.

The firm is also offering a new Autodial/Autoanswer Module for the Model III modem to allow unattended data transfer and remote access of the host computer for $79.95.

Both modems can transcribe and play back data communications on a cassette recorder at 300 baud.

The modems cost $199.50 each. Information is available from Micropheripheral Corp., 2643 151st Place N.E., Redmond, WA 98052.

Reader Service ✓325

MYCHESS Wins Tourney

Mychess is a new chess program for 32K TRS-80s which has won several computer chess tournaments.

Choose nine different levels with Mychess, and change levels or sides during the game. It will allow time limits for moves and save up to six games for later play or study, will predict the upcoming line of play after any move and take a turn or skip one.

It is available at $34.95 from Programma International Inc., 2908 N. Naomi St., Burbank, CA 91504.

Reader Service ✓160

Buffer Increases Efficiency

The new Microcompatible Print Buffer allows printers, plotters and other slow peripherals to work independently of your TRS-80, increasing efficiency by avoiding tying up the computer.

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM

FOR TRS-80*

Now the widely acclaimed JPC Cassette System is available for your TRS-80* computer. The price is only $90.00

[Reprint of June 1980 Review, *80 Microcomputing*]

TC-8 Cassette System
JPC Products
Albuquerque, NM
Kit: $90
Assembled: $120

by Carl A. Kollar

I guess I don't have to tell any TRS-80 owners how frustrating the cassette system that comes with the computer can be. Even with the factory mod that's available, the annoyance of loading and checking programs becomes just barely tolerable.

If you're like me, after you've just plunked down a chunk of money for a Level II 16K machine, "you ain't got nuttin left" for even one disk drive at 500 bucks apiece. So you suffer.

A reasonable alternative is the Exatron Stringy Floppy (ESF). This will cost you about 250 bucks and totally eliminates your loading and saving problems, automatically and fast. I've had one of these for about six months and love it!

But, if the price is still too steep, have I got a device for you!

The Device

The February 1980 issue of *Microcomputing* had an ad that intrigued the hell out of me. It was a high-speed cassette system by JPC Products acclaimed as a "poor man's floppy." It made all sorts of seemingly ridiculous claims such as "loads five times faster," "stores 50,000 bytes on a 10-minute cassette," "less than one bad load in a million bytes with the volume control anywhere between one and eight."

All this for a measly [90] bucks? How could this be? A call to Albuquerque answered a few questions: Yes, it had its own power supply, and, it stored programs five times faster because it utilized higher density data. The computer outputs the information at a higher rate out of the rear keyboard connector.

The ad had even claimed anyone could build it even if you have never soldered before. JPC would make it work, if you couldn't—for free. I was sold. I placed my order, and it arrived about two months later (parts shortage).

I work in electronics, so I found the unit exceptionally easy to build. It took about an hour. The manual is superb. (That's better than great.) It was clear, concise and exact with no ambiguities. Important parts placements are stressed (polarity markings on electrolytics, bands on diodes, etc.).

JPC was right! With these instructions, you couldn't go wrong. The board quality is excellent. It is double-sided and parts locations are clearly marked on the component side of the board. There are no jumper wires to install. JPC utilizes PC traces and plated-through holes for connections to traces on the other side of the board.

Also, there are absolutely no adjustments or settings to bother with.

The documentation is a sheaf of 8½ × 11 papers stapled together. It is written in the nicest format I've seen in a while. Each command and/or subjects is covered on its own sheet in large type. All explanations are in easy to read English—not computerese.

Commands and Features

SAVE"filename": Saves your BASIC program on cassette.
LOAD: Reads the next BASIC program from the cassette.
LOAD"filename": Searches for and loads the specified file from cassette.
LOAD? and LOAD?"filename": Reads file from cassette, and compares contents to memory.
LOADN: Prints a list of all the programs on a cassette, until interrupted by the "break" key.
LOADN"filename": Same as above except the tape will stop at the end of the program named.
KILL: Removes the file manager program from memory so that the extra memory can be used by large programs.
RSET: Allows the operator to rewind and position the tape on tape recorders that have these functions tied to the motor control jack.
RUN"filename": TC-8 searches for a specified program and runs it immediately.
PUT"filename": Same as SAVE "filename", except it is for use with system tapes.
GET: Same as LOAD, except it is for use with system tapes.
GET"filename": Same as LOAD "filename", except it is for use with system tapes.
GET? and GET?"filename": Same as LOAD? and LOAD?"filename", except it is for use with system tapes.
GETN and GETN"filename": Same as LOADN and LOADN"filename", except it is for use with system tapes.
OPEN: Required before cassette input or output of a data file can be attempted.
CLOSE: Required to end a cassette data file.
PRINT#: Allows numerical or string data to be output to a cassette file.
INPUT#: Allows numerical or string data to be input from a cassette file.

I haven't counted them, so I don't know about the "one load in a million bytes" claim, but my son, Anthony (age 11), loaded about 30 of his programs from his Radio Shack format tape to a new TC-8 format tape. He's run them all and found no bad loads.

Unlike the standard tape system, you can position your tape anywhere before the program you want and not have to look for a blank spot between programs. The TC-8 patiently waits for the program you want and then starts loading without getting confused by the portion of the previous program you just fed it.

Try that on your regular cassette system; you'll wear out the reset button. ■

ORDER NOW

To order your TC-8 kit, send your check or money order for $90.00 plus $3.50 postage and handling to JPC PRODUCTS CO., 12021 Paisano Ct., Albuquerque, NM 87112 (New Mexico residents add 4% sales tax). Credit card orders accepted by phone or mail. Personal checks will delay shipment. We will otherwise immediately ship you the TC-8 kit, the cabinet, the ribbon cable, the power adapter, an instruction manual, and a cassette containing the software.

✓190

JPC PRODUCTS CO.
Phone (505) 294-4623
12021 Paisano Ct.
Albuquerque, N.M. 87112

✓See List of Advertisers on page 306

NEW PRODUCTS

The buffer comes in 16K and 32K sizes with a variety of I/O ports and a printer AC power control option. It can handle up to two input and two output ports with baud rates set separately from 110 to 9600 baud.

The buffers start at $299 from Microcompatible Inc., PO Box 106, Scaly Mountain, NC 28775.

Reader Service ✓341.

Wye Connects Multi-Units

Lords Small Systems Design has produced a wye connector to allow two printers to access the same computer or two computers to share one printer.

Designed originally to allow connection of both a word processing and data printer to the same computer, the wye is U.L. approved and has a bat toggle switch to determine which printer is in use.

The connector is available at $99.95 for the TRS-80 Model I, $129.95 for Model II and $109.95 for the Model III from Lords Small Systems Design, PO Box 99, Port Angeles, WA 98362.

Reader Service ✓334

Color Computer Speaks

"The Speaker" is a totally programmable phonetic voice synthesizer for the TRS-80 Color Computer which plugs into the ROM Pack slot and talks through the monitor television.

The Speaker with demonstration and utility programs in both machine language and Color BASIC is $179.95 from Alford & Assoc., PO Box 6745, Richmond, VA 23230.

Reader Service ✓169

New Program Analyzes Complex Structural Models

The ST10M Finite Element (Space Frame) MC↑2, allows engineers and architects to analyze almost any structure from airplanes to bridges on any model of the TRS-80.

The program models beam, truss, membrane and plate elements and three-dimensional volumes and generates stresses on individual structural members and deflections of points throughout the structure.

Information on the new program, which fits into only 32K of RAM capacity, is available from McClintock Corp., 7000 SW 62nd Ave., Box 430980, Miami, FL 33143.

Reader Service ✓338

Software Security Enhanced

A new, reversible software enhancement package gives security against unauthorized program modification and plagiarism without changing the program in any way.

The Locker by The Management requires no additional RAM, disk space, assembly language coding or user calls, is permanent and installs easily. The only visible change is that LIST and LLIST will not work.

The Locker is available for $24.95 from The Management, PO Box T, Aledo, TX 76008.

Reader Service ✓331

Fortran Converter Available

The Management's Fortranslator, designed to help translate BASIC programs into FORTRAN, is now available for Model II and III TRS-80s. An enhanced version of their original Model I program is also available.

The program translates BASIC programs into FORTRAN template including subroutine calls and "goto" forms. It is designed to do 60 to 85 percent of the conversion work.

The Model I and III versions are priced at $29.95 and the Model II at $34.95 and are available from The Management, Box T, Aledo, TX 76008.

Reader Service ✓332

Mod III Terminal Gets Smart

A smart terminal program for the TRS-80 Model III is available on 1500 baud cassette and double density disk.

SMARTIII allows transfer of BASIC programs between computer and cassette or disk, off-line text preparation with Electric Pencil or Scripsit for on-line transmission and, with an additional program, file, permits generation and storage of text.

The program, with many other applications, is available for $99.50 from The MicroPeripheral Corp., 2643 151st Place N.E., Redmond, WA 98052.

Reader Service ✓336

RSM Patch Your Model III

A new patch program allows you to use Small Systems Software's RSM with the TRS-80 Model III to provide you a machine code monitor.

The patch operates at both 500 and 1500 baud rates.

The RSM Patch is available from Remarkable Software, PO Box 1192, Muskegon, MI 49443.

Reader Service ✓345

Analyze Investment Property

The Income Propery Analysis System allows analysis of before and after tax cash flows of income properties including rent rates, expenses, and rates of return.

Compatible with the TRS-80 Model I or Model III with at least one disk drive and 32K of memory, it is available from Advanced Business Microsystems, 5801 Marvin D. Love Freeway, Suite 103, Dallas, TX 75237.

Reader Service ✓164

Games Challenge Operator

Challengem is offering five games for the TRS-80 16K Level II or Model III. Subjects are Roman Conquest, Attack of the Yeti, Disaster at Sealab, Time for the Sheriff, and Fighter Run.

They are available for $9.95 each from Anthony Targonski, 106 Fifth St., Cambridge, MA 02141.

Reader Service ✓165

Radio Log, Billing Computed

Radio stations can now keep track of their logs and billing electronically using two new programs for the TRS-80 Model I.

The Electronic Log and The Electronic Bill are available from The Management, Box 111, Aledo, Texas 76008.

Reader Service ✓173

New Printer Costs Less

The Sidewinder is a new low-cost printer for all TRS-80 printer interfaces that prints 12 lines at once on 2-inch wide paper rolls.

Totally LPRINT and LLIST compatible, it prints 30 characters per second with a stationary thermal printing element.

The printer, software driver and a roll of paper sell for $199. An interface cable for the TRS-80 expansion interface is $19 and an interface cable for the TRS-80 keyboard is $49, all from Printel, Inc., Box 21094, Oklahoma City, OK 73156.

Reader Service ✓335

Double Your Memory Easily

Holmes Engineering's Internal Memory is designed to double the memory of the standard TRS-80 Model I to 32K bytes easily and inexpensively.

The device attaches to the RAM sockets inside the keyboard unit without modification or soldering of the unit, and the 10-page instruction booklet makes installation easy for those with little or no technical knowledge.

The unit is available at $47.50 without RAM from Holmes Engineering, 6246W. 3705S. SLC, UT 84120.

Reader Service ✓343

SGL Cuts Electric Pollution

SGL Waber Electric is bringing out a new line of power filters that protect semiconductors from electrical power surges, drops and other forms of damaging "electric pollution."

The new line of Power Master Line Monitor Power Conditioners range from plug-in units to a very sophisticated console or rack-mounted unit. All are designed to smooth out power fluctuations that can damage semiconductors or cause false data input and output.

Information is available from SGL Waber Electric, 300 Harvard Ave., Westville, NJ 08093.

Reader Service ✓337

Model III Gets The Business

The Small Business Systems Group (SBSG) is converting its business software, designed originally for the TRS-80 Model I and II, for use with the Model III.

Out presently are accounts receivable, accounts payable, general ledger and payroll programs and SBSG is working on conversion of their inventory and invoicing systems programs.

The programs are available for $195 each from Small Business Systems Group, 6 Carlisle Road, Westford, MA 01886.

Reader Service ✓329

Printer Driver Is Versatile

The RS-232 is a versatile and flexible new printer driver that offers many parameters and options and works with many printers, using the TRS-80 RS-232-C board.

It is available on cassette at $35 from Micro Systems Software, 3235 Kifer Road, Suite 32, Santa Clara, CA 95051.

Reader Service ✓162

V300-25 Printer

Printer Has Many Features

A new series of daisy wheel printers offers seven- and eight-bit character lengths, single- and double-stop bits, odd and even parity, and 300, 600, 1200 and 2400 baud transmission speeds. A programmable vertical format unit gives you a 66-line maximum form length with top-of-form and VT justification.

The V300-25 printer is priced at $1,895 and the V300-45 and $2,195 from Vista Computer Co., 1317 E. Edinger Ave., Santa Ana, CA 92705.

Reader Service ✓340

Battery Power for TRS-80s

The DC-80 direct connect power converter allows operation for a TRS-80 or expansion interface from a 12 volt battery.

The system gives an uninterruptable power source, 30 percent cooler computer operation and unlimited portability.

It is available for $49 Canadian from Digital Concepts Inc., 116 Moccasin Dr., Waterloo, Ontario, Canada N2L 4C3.

Reader Service ✓167

Computer Plays Blackjack

Casino Blackjack/Counter allows the player to simply play the game, try out playing strategies or learn and practice card counting with between one and four decks. The player has the center hand of five hands and sees all cards as they fall.

Compatible with TRS-80 Models I and III, it comes in cassette for $14.95 or disk for $19.95 from Manhattan Software, PO Box 32, Pacific Palisades, CA 90272.

Reader Service ✓166

EPROM Programs Model I

The EP-2A-79 EPROM Programmer will now interface with the TRS-80 Model I with a TR-24 module. Eleven EPROMs are offered on disk or cassette, all entirely self-prompting. The user can create and edit a program in memory, read an object code from disk or tape, or read and duplicate an existing programmed memory.

The Programmer is available from $169 and the TR-24 interface for $85 from Optimal Technology Inc., Blue Wood 127, Earlysville, VA 22936.

Reader Service ✓347

Reader Interfaces Scanner

Reader interfaces all model TRS-80s with Scan-Tron Model 2012 allowing the computer to read pencil-marked forms for test scoring and analysis, student registration, surveys, inventory and labor accounting and many other applications.

Using less than 1K of RAM, Reader is available on diskette for $175 from Desert Sound, Inc., 16268 Main St., Hesperia, CA 92345.

Reader Service ✓349

Program Tabulates Polls

A new program for tabulating political, public opinion, sociological or market research polls has grown out of the latest U.S. Senate race.

The new program was originally developed for a successful senate campaign and is extremely flexible, even tabulating open-ended questions. Compatible with any model TRS-80 with 32K and one disk drive, it costs $99.95 from M, M, & S Software, 3 Seabrook Court, Stony Brook, NY 11790.

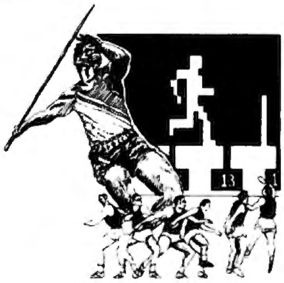
Reader Service ✓170

Get the most from your micro with software and accessories from one of the world's largest selections.

The Program Store

CALL TOLL FREE 800 424-2738

OLYMPIC



DECATHLON

By Timothy Smith from Microsoft
The graphics capabilities you were promised when you bought your computer are finally utilized in this marvelous series of programs. Just like the real Decathlon, you compete in 10 demanding games that encompass different forms of running, jumping and throwing.

Play alone or with as many as eight competitors, the gold medal will always go to the skillful -- never the luckiest -- because your score depends entirely on skillful manipulation of the keyboard.

One of our more expensive game collections -- and worth every penny! You MUST see this system in action. Otherwise, you simply won't believe the combination of truly outstanding graphics, fast-paced action, nail-biting intensity, and even a touch of comedy you'll experience with Olympic Decathlon!

16K Tape...$24.95 32K Disk...$24.95

COSMIC FIGHTER

By B. Hogue & J. Konyu from Big Five
Terrific sound, graphics and unique challenges mark this new space game a winner! While fighting off the alien convoys, each more skillful than the last, you must keep track of your rocket fuel or risk explosion as you maneuver toward your space station. Can you dock immediately, or is the station overrun by aliens? Find out by ordering Cosmic Fighter today.

16K Tape...$14.95
32K Disk version...$17.95

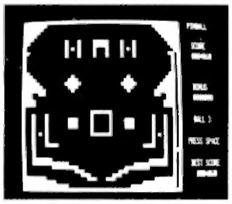
PROJECT OMEGA

By Bob Nicholas -- Adventure International
In probably the most accurate simulation ever produced for a microcomputer, you are responsible for the production, finance, health and well-being of Project Omega, the Earth's first deep space colony.

Painstakingly researched, Project Omega will provide much enjoyment and satisfaction as you overcome the frustrations and obstacles of taming an uncharted environment. The tape version is for one player; the disk version supports one or more, plus a special tournament option.

16K Tape...$14.95 32K Disk...$24.95

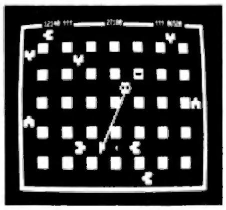
PINBALL



By John Allen from Acorn
Get your flipper fingers ready for action in this real-time, machine language game.

Lots of sound and flashing graphics make this fast action game so much like the real thing that you'll have to remind yourself not to shake your TRS-80. Choose from five playing speeds to match your skill. Can you beat your friends' scores? Will you avoid the dreaded "Bermuda Square?" Get PINBALL today and find out.

Protected tape...$14.95
Protected disk...$20.95



ATTACK FORCE!

By B. Hogue & J. Konyu from Big Five
Unlike the usual space "shoot-em-ups," your ship is not tied to the bottom of the screen. In Attack Force, you use the arrow keys to control both speed and direction as you maneuver all over the screen in search of the alien Ramships and Flagships. A realtime, machine language game with amazing graphics and sound.

You have to be quick to avoid the enemy ships that warp down on you, and the Flagships' lasers can fire in all directions -- even diagonally! And don't look away for an instant, because one of the alien spacecraft might be transformed into a mirror-image of your own!

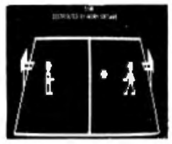
16K Tape...$14.95
32K Disk Version...$17.95

ZORK

By Infocom from Personal Software
In Zork, the Great Underground Empire, unearthly creatures guard 20 treasures. Bring all the treasures back to the trophy case and you can leave alive! You must pick your way through intricate mazes, collecting objects that may help or hinder you in your quest. But keep your wits about you, because in Zork, they take no prisoners!

TRS 80 or Apple II, 32K Disk...$39.95

BASKET BALL



By John Allen from Acorn
You have to be fast to keep up with the action as you try to outscore your opponent in five minutes of one-on-one basketball. Compete against a friend or your computer.

Steal the ball, duck around your opponent and slant toward the basket for a lay up! The graphics are based on a 3-dimensional depiction of a basketball court, and ball dribbling sounds add to the realism.

Protected Tape...$14.95
Protected Disk...$20.95

LDOS

JUST ANOTHER PRETTY DOS?

If new LDOS were just another disk operating system (DOS), we would not recommend it to you. However, two differences make this system unique and important: customer support and user benefits.

When you buy any DOS, you need service for programming assistance, updates and tips on how best to use its features. This is part of what you buy with LDOS. First, you will be supported by a toll-free phone line listed on your registration card. Second, you will be promptly notified of all updates and may send your original diskette to a service center for updating. You pay only the cost of return mail -- you can do it every week if you like. Third, a regular newsletter will inform you of any updates and provide tips on using some of LDOS's many special features.

With LDOS you get a well documented, thoroughly tested, and powerful DOS. The publisher is committed to a professionally written and detailed users' manual. Besides contracting with some of the best microcomputer systems houses for technical and customer support, a highly regarded technical writing firm is doing the manual.

The power of LDOS is its ease of operation, its independence of hardware configuration, and its device independence. You can make selected backups, chain together a series of programs and operations, and operate several different types of drives from the computer. LDOS has all the features of VTOS 4.0, fully implemented and working. There are hundreds of features which we do not have room to mention, but the best features are its ease of use for the new disk owner combined with its sheer power potential for the expert. We will gladly send you more information on LDOS if you just call our toll free number.

LDOS with Manual...$139

THE PROGRAM STORE · Dept K11 Box 9609 · 4200 Wisconsin Ave. NW · Washington, D.C. 20016

Item ____ Price ____ Postage $1.00 Total ____

☐ CHECK ☐ VISA ☐ MASTERCARD MC Bank # ____ Card # ____ Exp ____

name ____
addr ____
city ____ state ____ zip ____

See List of Advertisers on page 306

Let's assemble and install that interface.

High Density Graphic Interface—Part Two

Dennis Murray
CompuTech
1005 Chestnut Dr.
Christiansburg, VA 24073

Paul Fowler, Jr.
Enable Electronics
2103 Charlton Lane
Radford, VA 24141

Last month we discussed the theory and operation of the TRS-80 video section and the high-resolution graphics interface. This month we'll explain how to assemble and install the high-resolution interface. The authors felt that the cleanest approach would be a step-by-step procedure with boxes to check off as each step is completed.

Finally there will be a discussion and examples of a software driver for the graphics.

Assembly Procedure

Note: Installing this board or opening your TRS-80 in any way voids the manufacturer's warranty. We suggest that you wait until the warranty period is over before adding this modification.

You will be working with components that are sensitive to static electric discharges. Therefore, standard CMOS handling procedures should be followed.

☐ Check the circuit board (HR1) for solder bridges. Remove any that you find.

☐ Orient HR1 so that the side that says "video out" (on the right hand side of board) faces you. (See Fig. 1.)

☐ In descending order, solder memory chips IC9 thru IC4 in place. Be sure to start with IC9 in the upper right-hand corner. This ensures that all memory chips find their correct mounting holes. Note the location of Pin 1 on each IC, which is designated by a dot or is to the left of the notched edge of the chip as viewed from the top (Fig. 1).

☐ Solder the remaining twelve chips in place.

☐ Double-check orientation of Pin 1 on each IC according to the diagram in Fig. 1.

☐ Solder Dip Switch DP1 in place.

☐ Solder the eleven 10k resistors (R1 thru R11) to the circuit board.

☐ Solder the nineteen .01 uF capacitors as shown.

At this point, construction of the high-density interface board

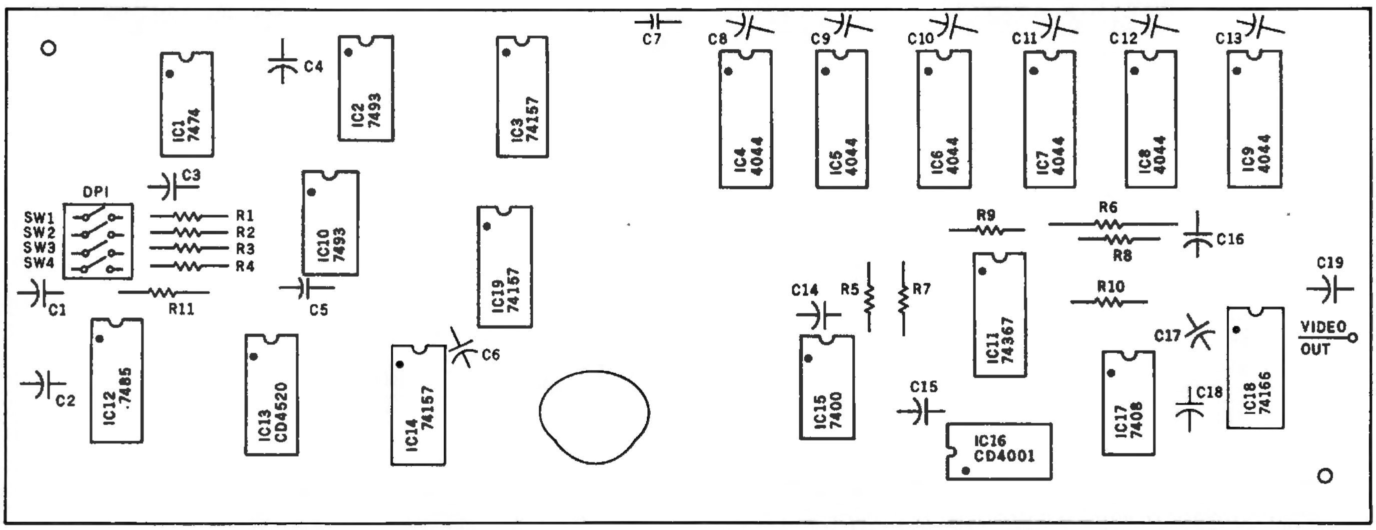


Fig. 1. All Series 74 chips are 74LS (Low Power Schottky).

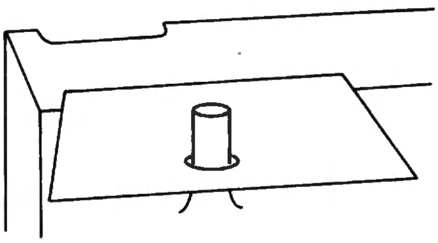


Fig. 2.

is complete. Next we must wire the board in to the TRS-80.

Disassembly Of the TRS-80 Keyboard

□Remove the power, video, cassette, and any other cables connected to the CPU.

□Turn the keyboard face down on a soft surface (towel, rug, etc.) to keep from scratching the finish and remove the six screws. Note the position of the different size screws; the longer ones go toward the rear of the unit.

□Grasp the CPU by both ends, flip it right side up, and lift off the top to expose the keyboard.

□Lift the keyboard from the back and pull it toward you, taking care not to stress the interconnecting cable going to the CPU board. Remove the five plastic spacers from the keyboard support posts and lay the keyboard back down.

□Holding the keyboard and CPU board in place, turn the entire unit upside down once more.

□Lift the bottom off of the case.

□Set the key/CPU boards aside and lay the case bottom in front of you so that the support posts face you with the longest posts toward the rear.

□Locate the left rear post. With component side up, place the large 9/16" hole of the hi-res circuit board over the post. (See Fig. 2).

□Using the 1/8" holes in the upper left and lower right hand corners of the hi-res board as a guide, mark the case bottom for drilling.

□Drill two 1/8" holes in the case bottom for mounting the hi-res board, but *do not* mount the board at this time.

□Select a convenient location on the case bottom (or case top) and mount SW5. This switch is for selection of the normal TRS-80 graphics *or* the high-resolution graphics.

□If you are going to use the memory disable option, locate a convenient location and mount SW6 at this time (see below).

□Set the case aside.

Getting Wired

Place the hi-res circuit board in front of you and examine the various inputs. Note: SHIFT = SH and SHIFT/LOAD = SH/LD.

□Referring to Table 1, cut the wires to the length indicated. Trim, tin, and solder them to the hi-res circuit board.

□On shift and video out use twisted pair with at least three turns per inch. An easy way to make twisted pair is to chuck two lengths of wire in a hand drill, anchor, or have someone hold the other ends of the wires and turn the drill.

Use two different colors of wire to make it easy to determine which wire is grounded and which is signal. Be sure to

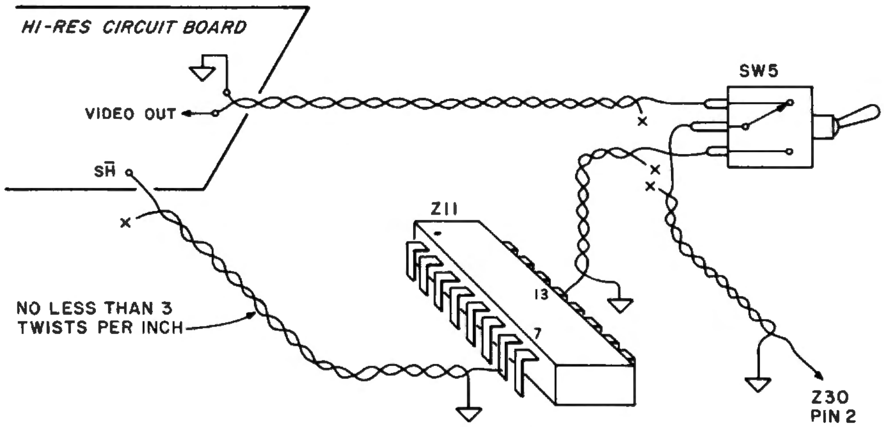


Fig. 3.

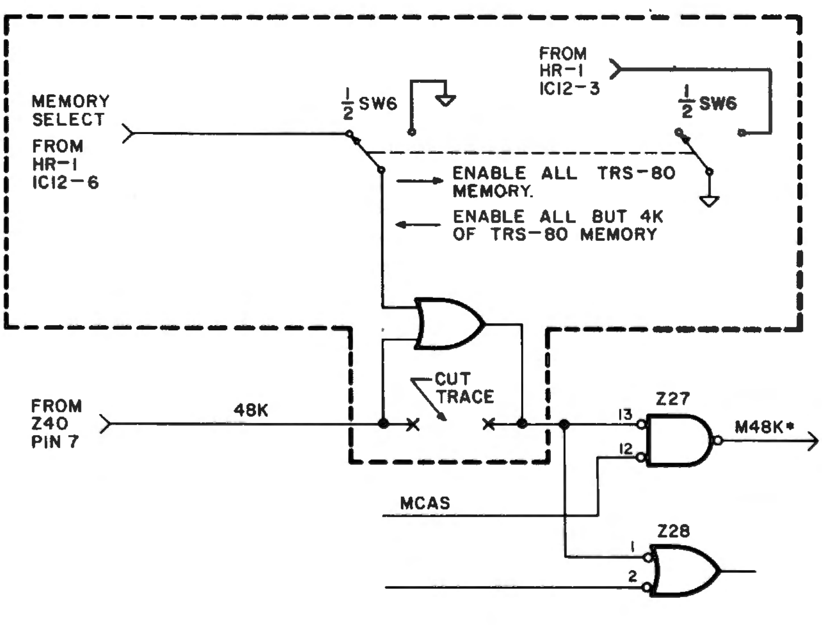


Fig. 5.

HRI MOUNTING PAD I.D.	TRS-80 IC#	TRS-80 IC PIN#	WIRE LENGTH
A3	Z22	13	4½"
A2	Z22	11	4½"
$\overline{WR}$	Z22	5	4½"
A9	Z39	13	4"
A7	Z39	11	4"
A5	Z39	9	4"
A8	Z39	3	4"
A6	Z39	5	4"
A4	Z39	7	4"
A11	Z38	13	4"
A14	Z38	11	5"
A15	Z38	9	5"
A10	Z38	3	4"
A12	Z38	5	5"
A13	Z38	7	5"
A1	Z55	13	4½"
A0	Z55	11	4½"
D1	Z76	13	6"
D0	Z76	11	6"
D2	Z75	13	6"
D5	Z75	3	6"
D3	Z75	5	6"
D4	Z75	7	6"
SHIFT	Z11	7	5"*
SHIFT/LOAD	Z11	15	5"
$\overline{VERT}$	Z32	11	8½"
$\overline{BLANK}$	Z30	10	9½"
HORZ	Z30	8	9½"
+5	Z55	16	5½"
GND	WIDE TRACE	(See Fig. 4)	4"
LATCH	Z9	3	10½"
DOT2	Z24	2	9½"
VIDEO OUT TO	SW5	(See Text)	*

*ASTERISK DENOTES TWISTED PAIR.

Table 1. Connect all wires in the order shown (start at top).

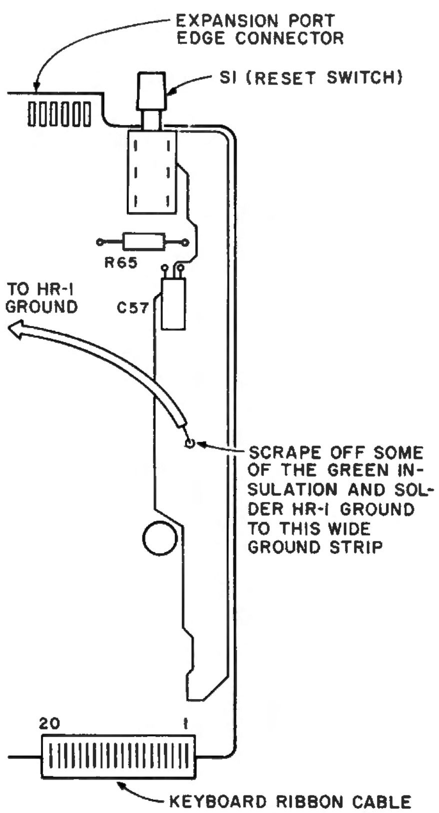


Fig. 4. The Ground Connection

Starting Addr. In Decimal	Hex Value	S4	S3	S2	S1
61440	F000	OFF	OFF	OFF	OFF
57344	E000	OFF	OFF	OFF	ON
53248	D000	OFF	OFF	ON	OFF
49152	C000	OFF	OFF	ON	ON
45056	B000	OFF	ON	OFF	OFF
40960	A000	OFF	ON	OFF	ON
36864	9000	OFF	ON	ON	OFF
32768	8000	OFF	ON	ON	ON
28672	7000	ON	OFF	OFF	OFF
24576	6000	ON	OFF	OFF	ON
20480	5000	ON	OFF	ON	OFF
16384	4000	ON	OFF	ON	ON
12288	3000	ON	ON	OFF	OFF
8192	2000	ON	ON	OFF	ON
4096	1000	ON	ON	ON	OFF
0	0000	ON	ON	ON	ON

Table 2.

ground only one end of the grounded pair. (See Fig. 3.) Make the video out twisted pair long enough to reach one pole of SW5.

☐ Place the key/CPU board in front of you and cut the trace from Z11 Pin 13 to Z30 Pin 2. This trace is easy to get to from the unpopulated side of the board.

☐ Place the keyboard upside down (component side up) in front of you.

☐ Turn HR1 component side down and place it on top of the CPU board, being careful to align the large hole.

☐ Starting at the top of Table 1, solder all wires from the originating HR1 mounting pad to the appropriate TRS-80 IC pin number.

☐ Solder GND to the wide trace next to SW1 on the CPU board. (See Fig. 4.)

☐ Referring to Fig. 3, solder one side of video out to one pole of SW5. Use twisted pair.

☐ Solder the other pole of SW5 to Z11 Pin 13 on the CPU board. Use twisted pair.

☐ Solder the throw (center lug) of SW5 to Z30 Pin 2. Use twisted pair.

Hi-res Memory Disable Option

The hi-res graphics interface is a memory-mapped device. Therefore, it is compatible with only the 4K, 16K, and 32K units. If the hi-res memory is to use the same address locations as an existing block of RAM, the memories will fight each other resulting in erratic performance.

If you have a system with less than 48K, set switches SW1 to SW4 to reference a range of ad-

Part	Qty.	Part #	Description
IC4, 5, 6 7, 8, 9	6	2147 or 5257 or 4044	4K × 1 Static Ram
IC18	1	74166	8-Bit Shift Register
IC11	1	74367	Hex Buffer Driver
IC15	1	7400	Quad Dual Input NAND Gate
IC17	1	7408	Quad Dual Input AND Gate
IC16	1	CD4001	Quad Dual Input NOR Gate (CMOS)
IC12	1	7485	4-Bit Magnitude Comparitor
IC1	1	7474	Dual D-Type Flip-Flop
IC13	1	CD4520	Dual 4-Bit Binary Counter
IC2, 10	2	7493	4-Bit Binary Counter
IC3, 14, 19	3	74157	Quad 2-Input Multiplexer
DP-1	1		Quad (SPST) Dip Switch
R1-R11	11		10k Resistor 20% 1/8 or 1/4 Watt
SW5	1		SPDT Miniature Switch
SW6	1		SPST Miniature Switch (optional)
HR1	1		Hi-Resolution Video Circuit Board
			Braided Hook-up Wire
C1-C20	20	.01 uF	Ceramic Capacitors

Note: All TTL Logic Chips May Be of the Low Power (LS) Series.

Table 3. Parts List

Orange Micro

296

"THE COMPUTER PRINTER SPECIALISTS"

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

CENTRONICS 737 (RADIO SHACK LINE PRINTER IV)

Word Processing Print Quality

• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 50/80 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns

CENTRONICS 737-1 (Parallel) ....(List $995) $765
CENTRONICS 737-3 (Serial).................(List $1045) $815

VISTA — C. ITOH

Daisy Wheel Letter Quality

• 25 CPS (Optional 45 CPS) • Typewriter quality • Centronics parallel • RS 232 Serial (Optional) • Proportional spacing • Bidirectional • Programmable VFU • Self test • Diablo compatible • Friction feed (Optional tractors) • 136 printable columns. • Manufactured by C. ITOH.

VISTA V300 (C. ITOH).......................(List $1895) $ Call

EPSON MX80/MX70

OUR BEST BUY!

Low-Priced Professional Print Quality

• 9 x 9 dot matrix • Lower case descenders • 80 CPS • Bidirectional, Logic seeking • 40, 66, 80, 132 columns per line • 64 special graphic characters: TRS-80 Compatible • Forms handling • Multi-pass printing • Adjustable tractors

EPSON MX80..............................(List $645) $Call
EPSON MX 70 Dot graphics, 5 x 7 matrix......(List $450) $Call

ANACOM

Low Cost, High Speed, Wide Carriage

• 9 x 9 dot matrix • Lower case descenders • Wide carriage • Adjustable tractors to 16" • 150 CPS, Bidirectional, Logic Seeking

ANACOM 150 ....(List $1350) $ Call

ANADEX

Dot Graphics, Wide Carriage

• 11 x 9 dot matrix; lower case descenders • Dot resolution graphics • Bi-directional, logic seeking • Up to 200 CPS • RS 232 Serial & Parallel • Forms control • X-ON/X-OFF • Up to 6 part copy.

ANADEX 9501............................(List $1650) $1350

OKIDATA MICROLINE SERIES

TRS-80 Graphics Compatibility, Friction Feed

• 9 x 7 dot matrix • 80 CPS • 80, 132 columns — 64 shapes for charts, graphs & diagrams • Double wide characters • 6/8 lines per inch • Up to 3 part copy • Friction & pin feed • 200 M character head warranty

OKIDATA MICROLINE 80 .... (List $800) $520
OKIDATA M82 Bidirectional, Forms handling .... (List $960) $750
OKIDATA M83 Wide carriage, 9 x 9 dot matrix ..... (List $1260) $1050

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing.

5510-5 RO, Serial, w/tractors.................(List $2995) $2625
5530-5 RO, Parallel, w/tractors................(List $2970) $2599

TELEVIDEO CRT'S

AT DISCOUNT PRICES!

TVI 912C
TVI 920C
TVI 950

Please Call Toll Free
Prices are too low to advertise

QUANTITY PRICING AVAILABLE

IDS PAPER TIGERS

Dot Resolution Graphics, quality print, speed

• 7 wire printhead (445); 9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking (460) • 8 character sizes; 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification (460).

IDS 445G 7 wire printhead, graphics ....(List $895) $ 750
IDS 460G 9 wire printhead, graphics ....(List $1394) $1150
IDS 560G 9 wire, wide carriage, graphics ......(List $1794) $1590

PRINTERS

MALIBU 165 wide carriage, graphics, letter quality .. (List $2495) $ 1975
QUME 5/45 typewriter quality ....(List $2905) $ 2559

INTERFACE EQUIPMENT

CCS APPLE PARALLEL Interface & cable .... $ 150
APPLE II - EPSON MX80 parallel interface board & cable .... $ 100
SSM AIO BOARD Apple Serial/parallel interface ...... (List $225) $ 175
MICROTRONICS Atari parallel interface..................... $ 69
ATARI 850 Interface module, serial/parallel .... $ 199
TRS-80 CABLES to keyboard or Exp. interface .... $ Call
NOVATION D-CAT direct connect modem.................. $ Call

CALL FOR FREE CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

At Orange Micro, we try to fit the right printer to your application. Call our printer specialists for free consultation.

TELEPHONE ORDERS: Mon.-Fri. 8:30 - 5:00
The Orange Micro Printer Store (Retail):
Mon.-Fri. 10:00 - 6:00, Sat. til 4:00

Phone orders WELCOME; same day shipment. Free use of VISA & MASTERCARD. Personal checks require 2 weeks to clear. Manufacturer's warranty included on all equipment. Prices subject to revision.

Orange Micro, Inc.
3150 E. La Palma, Suite I
Anaheim, CA 92806

```
6000 'THIS SECTION WILL "SET" SCREEN FOR GRAPHICS
6010 FOR W=15360 TO 16383
6020 POKE W,128
6030 NEXT W:RETURN
6040 '
6050 '
6100 ' TO "CLEAR" HI-RES SCREEN
6110 Z=0
6120 FOR W=-32768 TO -29696
6130 POKE W,Z
6140 NEXT W:RETURN
6150 '
6160 '
6200 'TO "SET" HI-RES SCREEN
6220 Z=191:GOSUB 6120
6230 RETURN
6240 '
6250 '
6300 ' THIS SECTION "PUTS" X1,Y1 TO SCREEN
6310 IF X1<0 THEN STOP
6320 IF X1>191 THEN STOP
6330 IF Y1<0 THEN STOP
6340 IF Y1>95 THEN STOP
6350 I1=INT(X1/6) 'I1=HORZ. DISPLACEMENT IN BYTES
6360 I2=Y1*32+I1 ' I2=DISPLACEMENT FROM FIRST LOCATION
6370 I3=2[(5-X1+I1*6)  ' I3 IS THE BIT MASK
6380 I4=I2-1*(65536-32768)  'I4=THE ABSOLUTE MEMORY LOC
     ATION
6390 I5=PEEK(I4) ' I5= PRESENT CONTENTS OF MEMORY
6400 IF X0=0 THEN 6500 'IF X0=0 WE WANT A BLACK DOT; IF

     X=ANYTHING ELSE WE WANT A WHITE DOT
6410 I5=I5 OR I3 'ANY BITS ON LEAVE ON; OTHERWISE, LEAV
     E OFF
6500 POKE I4,I5
6510 RETURN
6600 I3=255-I3
6610 I5=I5 AND I3
6620 GOTO 6500 'GO PUT THIS BYTE TO SCREEN
```

Program Listing 1. High Resolution Drive

dresses not used by your system. (See Table 2.) This range will become the memory addresses used by the high-resolution graphics interface.

If you have a 48K system, then you must disable a portion of computer RAM and enable the hi-res RAM when this mode is desired. This may be accomplished with a DPDT switch as shown in Fig. 5. Note: The authors have not implemented the circuit in Fig. 5. It is presented only as a suggestion.

Learning How to Blink

Now that the serious part is over and we have built this graphics wonder, we need some software to make it work. This is accomplished quite easily with the BASIC subroutines in Program Listing 1. Recalling that the hi-res interface generates 192 dots horizontally and 96 dots vertically, let X1 = the horizontal dots and Y1 = the vertical dots. Therefore, X1, Y1 will define a single point on the monitor screen. Let X0 be a flag that defines the color of the dot: If X0 = 0 the dot will be black; if X0 = 1 the dot will be white. Assume that the screen is ready to accept graphics and that it is totally blank. Also, assume that X1 = 100, Y1 = 50, X0 = 1, and that we have just "GOne SUB" to line 6300. Since line 6300 is a remark we will do nothing.

Lines 6310-6340 check to see that X1 and Y1 fall within the limits of the video graphics. Line 6350 defines the horizontal displacement in six bit bytes (remember, there are only six bits/word in this interface). This value is called I1 and is INT (100/6) = 16 in this case. Therefore, this point 100,50 will be located 16 bytes from the left side of the screen. Line 6360 defines I2 which is the displacement in 6 bit bytes from location 0,0. In this case I2 = 50 lines down × 32 bytes/line + 16 bytes from the first location on the screen (or 1616 bytes from the first location).

The next line (6370) tells us which of the 6 bits in byte #1616 will be on. As you can see $I3 = 2^{(5-100+16\times 6)} = 2^1 = 2$. This means that the second bit from the right in byte #1616 is to be turned on. Why the second bit? Because two decimal equals 000010 binary. If X1 = 99 you would find that I3 would equal 2^2 or 4 and that is the 3rd bit over, or 000100 binary.

The absolute memory location is given in line 6380 by I4. The first screen memory location in this example is at 8000 hex or 32768 decimal. Therefore, the absolute memory location to poke is 32768 plus 1616, right? Wrong! Since the TRS-80 uses ones-complement to PEEK and POKE addresses above 32768 we must obtain the ones-complement of 32768 + 1616. Fortunately, a handy little formula is defined for us in the Level II Manual under the section pertaining to PEEKs and POKEs. Thus, the absolute ones-compliment of the address is—ta-da—1616 − 1 × (65536 − 32768) or −31152 decimal. (32768 is the starting location of the first screen address. If the starting location were C000 hex then this value would be 49152 decimal.) Clear as mud.

Let's review. We know that we want to access the 1616th byte from the starting location of the screen, that the absolute location is −31152 in memory, and that we want to load that location with 000010 binary. So let's go ahead and POKE = −31152 (2), right? *Wrong!*

What about the other five bits in byte −31152? What if they're all on? Then we just POKED a black line into our screen. How to get around this? Simple, PEEK location −31152 and return the value as I5 in line 6390. Look at it and leave ON bits on, and if bit 100,50 is off, turn it on. This is what happens in lines 6410 and 6500.

What? What about line 6400, you say? Boy, some people worry about everything! Remember X0? X0 gives us the option to turn 13 = 000010 off. If X0 = 0 then we would have bypassed 6410 and obtained the compliment of 000010 or 111101 in line 6600. ANDing this with whatever I5 equals would leave location −31152 the same as it was except the 2's bit would be off. Regardless of whether X0 = 0 or 1 we end up POKEing I4, I5 in line 6500 and returning to our main program.

Now, this is all well and good to go POKEing around in the raster, but if you don't heed this one little rule, you won't get what you're after!

The TRS-80 must have its screen set for graphics. This is accomplished by setting the most significant bit of memory locations 3C00H to 3FFFH to a true condition (1). Use the little subroutine from lines 6000 to 6030 for this. It simply POKEs the 1024 memory locations allocated to the TRS-80 normal graphics with 10000000 binary (128 decimal). Remember in last month's discussion that these bits had to be 1s in order for the TRS-80 graphics to be accessed. And so it must be for the hi-resolution graphics. This method allows the normal alphanumerics to be used with the hi-res graphics.

```
10 GOSUB 6000: GOSUB 6100
20 Z=45: X0=1
30 D=1
40 V=-1
50 FOR I=1 TO 2
60 FOR D=D TO D+179
70 A=D*.0174
80 X1=INT(V*COS(A)*45)+Z
90 Y1=INT(-SIN(A)*45)+45
100 GOSUB 6300
110 NEXT D
120 Z=Z+90
130 V=-V
140 NEXT I
150 FOR X1=0 TO 191: Y1=45: GOSUB 6300: NEXT X1
160 FOR Y1=0 TO 95: X1=96: GOSUB 6300: NEXT Y1
170 INPUT Q
```

Program Listing 2. Sine Wave Plot in BASIC

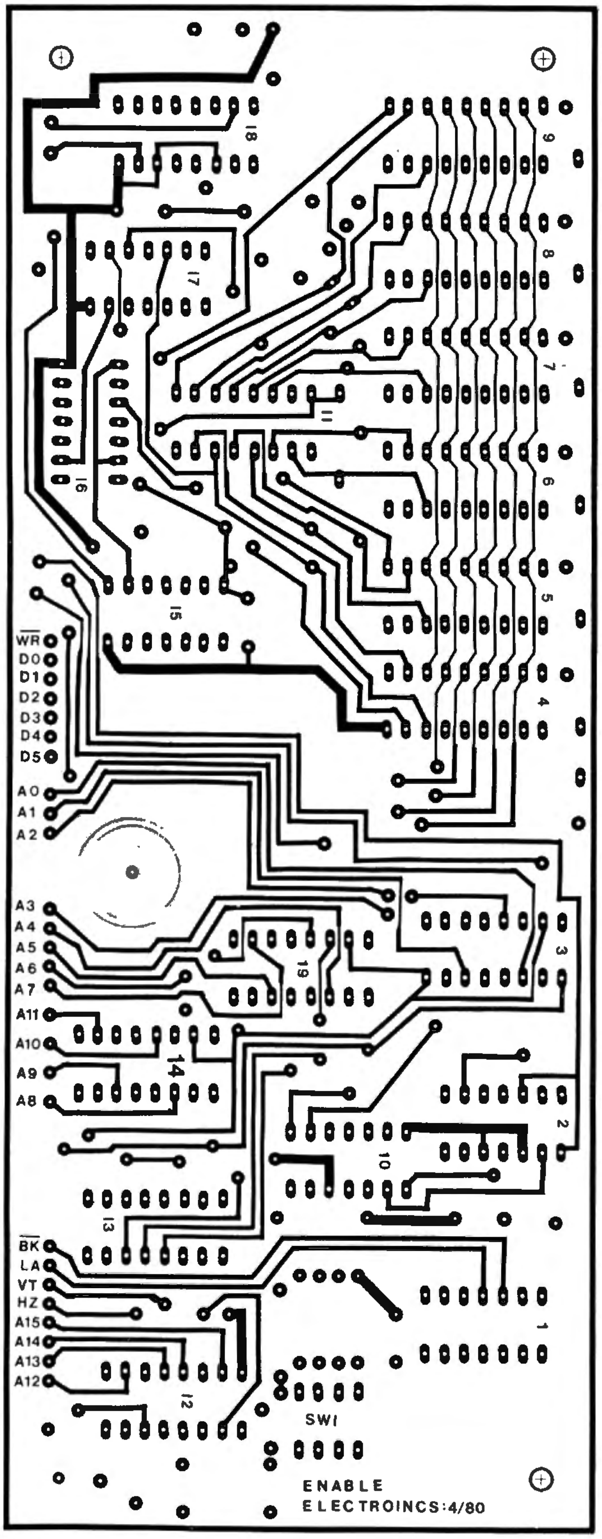
Two more little subroutines have been added for convenience. The Clear subroutine from lines 6100 to 6140 will set all values in the hi-res memory to zero giving a black screen. The Set routine (6200-6730) will set all hi-res values to one, giving a white screen.

Now that Program Listing 1 has been explained, let's say a few words about Program Listing 2. It's a sine wave. Run it.

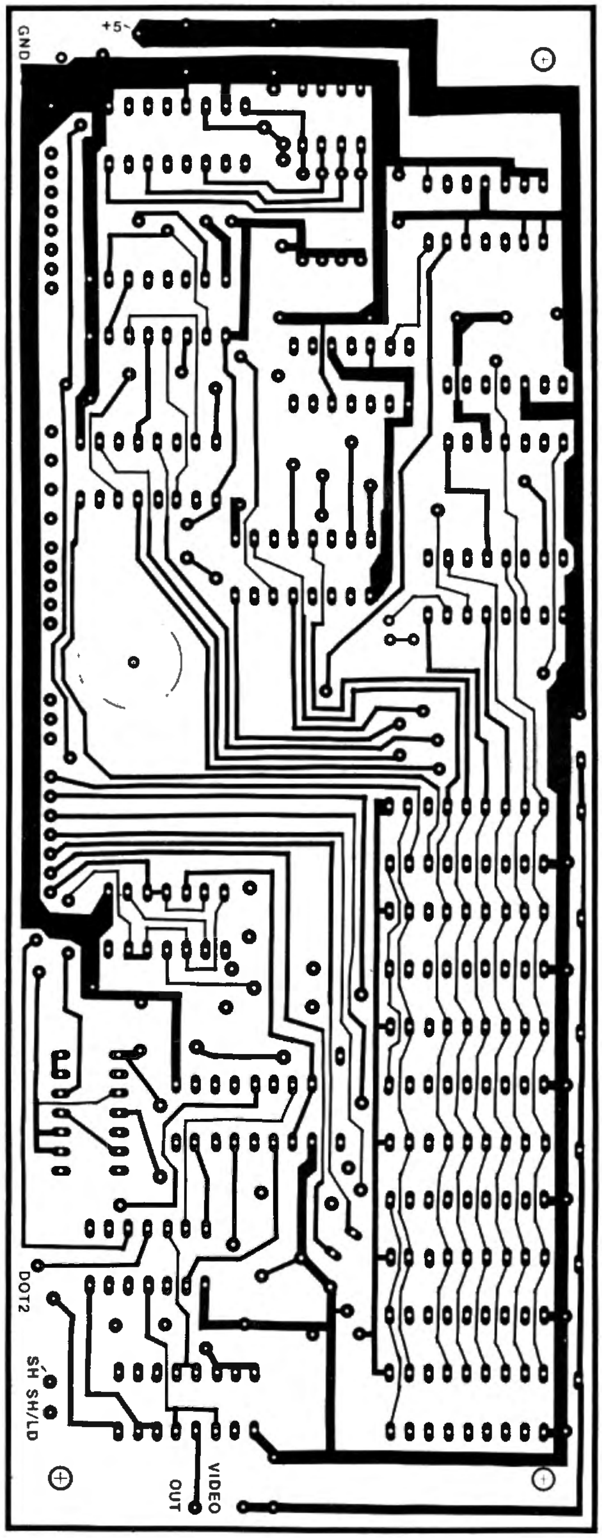
Now that you have High-resolution capability, new horizons will open to you; great challenges, empires will rise and fall! . . . .

Maybe you'll even draw a few graphs. ■

A set of high-speed plotting subroutines written in assembly language will be available in June. These routines are easily accessible from BASIC and will draw a straight line between any two points or plot individual dots. Anyone interested in this package should contact the authors for pricing and recording media available.



Bottom view



Top View

✓222 for catalog ✓224 for dealer information ✓213 for author information

ADVENTURE INTERNATIONAL BOX 3435 LONGWOOD, FL 32750 (305) 862-6917

This car buying advisor for confused consumers is a lesson in modular programming.

The Auto Mentor

Leslie E. Sparks
1014 Evergreen Drive
Durham, NC 27712

Should I replace my old car? Should I fix my old car that burns leaded gas or get a new one that burns unleaded? Is it worth $1000 to get an extra five miles per gallon?

Interesting questions, aren't they? Especially when you consider that the difference between a right and a wrong answer can be several hundreds of dollars. A properly programmed computer can help you find the right answer. That's right, your computer can start earning its keep.

The hitch, of course, is in finding a properly programmed computer. Mine didn't come out of the box programmed to do anything except ask MEMORY SIZE? But, for me, that's the fun of owning a computer.

All the programs I've developed follow the same general steps.

1) Defining the problem.
2) Developing the necessary relationships—generally mathematical equations.
3) Developing the basic structure for program.
4) Coding the problem in computer language.
5) Typing the program.
6) Debugging.
7) Writing documentation.

Problem Definition

Your problem must be defined in a way that the computer can be programmed to answer the question. At the same time, you must develop the criteria for making a decision.

The question, "Should I get a new car?" is not specific enough for the computer.

What is it that we really want to know?

The question really is: What is the optimum way to spend my car money? The criteria are economic and this optimum is defined in economic terms.

The problem is still a long way from well specified. What is this economic optimum? How is it calculated?

"The concept of present value is used in most investment analyses and is a way of comparing cash flows with different timing on a common basis."

Our original question is actually an investment problem. According to one theory of investment analysis, the optimum is that month which maximizes the net present value of keeping the old car and replacing it at the end of the month, compared with the costs of replacing the old car now. Two more questions: What is the value of keeping the old car? What is it's present value?

Take the easy one first. The value of keeping the old car, V, is the total cost of owning and operating the new car, TC(New), minus the total cost of owning and operating the old car, TC(Old). Or mathematically:

$$V = TC(New) - TC(Old)$$

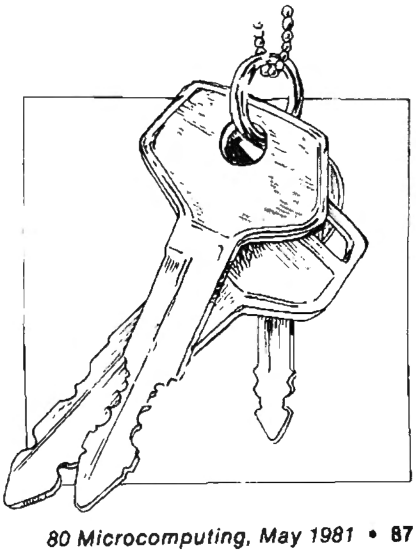
If the value of keeping the old car is negative, it's obviously time to replace it. But the optimum time may occur while the value of keeping the old car is still positive.

The total cost of owning and operating a car, old or new, can be broken into separate items.

- The cost of fuel, F.
- The cost of normal maintenance, M.
- The cost of insurance, IN.
- The cost of special repairs, S.
- The cost of monthly car payment, IP.
- The cost of the down payment, Down.

The benefit of getting a new fuel efficient car is that fuel costs will be less for the new car; maintenance costs may also be lower. Balanced against the fuel cost savings are the down payment, the monthly car payment and, probably, the cost of insurance.

How are the future benefits and costs compared so that a decision can be made?



Timing Important

It is incorrect to compare all these costs, or cash flows, unless their timing is also considered. The comparison is made by adjusting all the individual costs to a common time—the present. The present value of keeping the old car is found by taking all the current and future cash flows for both the old and new car and adjusting them to the present.

Future cash flows have to be adjusted because a dollar is worth more now than a dollar received next year. The current dollar can earn interest, for example, and will be worth a dollar plus the interest in the future. This adjustment is called discounting, and may be looked at as the opposite of compounding interest. All these discounted cash flows are then totaled to give the present value of keeping the old car.

The discount factors are:

- df = 1/(1 + r)↑N for a one-time cash flow at period N.
- df = (1 − (1/(1 + r)↑N))/r for a series of equal cash flows with r<>0.
- df = N for a series of equal cash flows with r = 0.

N is the number of periods and r is the discount rate.

One additional complication is that costs escalate with time. This escalation of costs can be balanced by replacing the discount rate with an effective discount rate (the discount rate minus the escalation rate).

What is the discount rate? A good value for the discount rate is the risk-free interest that you can earn on your money.

It is now possible to write a complete set of equations for keeping the old car N more months and then replacing it. Or purchasing a new fuel efficient car immediately. These equations are given in Table 1. These equations include the effects of taxes, the trade-in value of the old car and the salvage value of the new car. We are interested in the incremental costs. If, for example, the escalation rate of new car prices is the same as the discount rate, then no costs are incurred as a result of lost interest by purchasing the new car now.

Running the Program

Let's begin translating the equations in Table 1 into a computer program.

The program must do the following tasks:

- Get data from the user.
- Calculate the present value of keeping the old car.
- Display the results on the CRT.
- Print the results on a line printer, if the user wants.

Each of these tasks should be handled in a subroutine or program module. This means, for example, that data input doesn't

Fig. 1. Sample Problem Results

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
DETAILED RESULTS OF THE CALCULATIONS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
MONTH     VALUE      TC(OLD)          TC(NEW)        FUEL(OLD)-FUEL(NEW)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
 1       5.82956     167.801         173.63             94.4546
 2       11.2439     334.658         345.901            188.573
 3       16.2542     500.574         516.828            282.352
 4       20.8596     665.562         686.421            375.797
 5       25.0681     829.621         854.689            468.906
 6       28.8856     992.755         1021.64            561.677
 7       32.3131     1154.98         1187.29            654.118
 8       35.3567     1316.29         1351.65            746.224
 9       38.0243     1476.69         1514.72            837.995
 10      40.3204     1636.2          1676.52            929.431
 11      42.2443     1794.81         1837.05            1020.54
 12      43.8068     1952.53         1996.34            1111.31
 13      45.0054     2109.37         2154.38            1201.76
 14      45.853      2265.33         2311.18            1291.87
 15      46.3442     2420.42         2466.76            1381.66
 16      46.4934     2574.63         2621.13            1471.12
 17      46.3001     2727.99         2774.29            1560.25
 18      45.7693     2880.48         2926.25            1649.05
 19      44.9036     3032.13         3077.03            1737.52
 20      43.7097     3182.92         3226.63            1825.67
 21      42.1899     3332.88         3375.07            1913.5
 22      40.3508     3481.99         3522.34            2000.99
 23      38.1958     3630.27         3668.46            2088.17
 24      35.729      3777.72         3813.45            2175.02
 25      32.9519     3924.35         3957.3             2261.55
 26      29.8699     4070.16         4100.03            2347.76
 27      26.4893     4215.16         4241.65            2433.64
 28      22.8115     4359.34         4382.16            2519.21
 29      18.8413     4502.73         4521.57            2604.45
 30      14.5806     4645.31         4659.89            2689.38
 31      10.0381     4787.1          4797.14            2773.99
 32      5.21436     4928.1          4933.31            2858.28
```

Figure continues

"The main program module is a series of GOSUB statements that transfer control to the other modules as needed."

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
DETAILED RESULTS OF THE CALCULATIONS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
MONTH      VALUE         TC(OLD)             TC(NEW)           FUEL(OLD)-FUEL(NEW)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

   33     .11084          5068.31       5068.42                2942.25
   34     -5.26465        5207.75       5202.48                3025.91
   35     -10.9077        5346.4        5335.49                3109.25
   36     -16.8213        5484.29       5467.46                3192.28
   37     -22.9946        5621.4        5598.41                3274.99
   38     -29.4287        5757.76       5728.33                3357.39
   39     -36.1167        5893.35       5857.24                3439.40
   40     -43.0562        6028.19       5985.14                3521.26
   41     -50.2471        6162.29       6112.04                3602.72
   42     -57.6792        6295.63       6237.95                3683.87
   43     -65.3555        6428.24       6362.88                3764.72
   44     -73.272         6560.11       6486.84                3845.26
   45     -81.4209        6691.24       6609.82                3925.49
   46     -89.8027        6821.66       6731.85                4005.41
   47     -98.4126        6951.34       6852.93                4085.02
   48     -107.249        7080.31       6973.06                4164.33


$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
                         CAR ANALYSIS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
                         INPUT DATA
MILES DRIVEN PER YEAR                                      15000
CURRENT PRICE OF FUEL FOR OLD CAR $/GAL                    1.3
CURRENT PRICE OF FUEL FOR NEW CAR $/GAL                    1.3
MILES PER GALLON OLD CAR                                   12
MILES PER GALLON NEW CAR                                   40
INSURANCE COST OLD CAR $/YR                                150
INSURANCE COST NEW CAR $/YR                                350
MAINTANENCE COST OF OLD CAR PER YEAR                       250
MAINTANENCE COST OF NEW CAR PER YEAR                       100
PRESENT TRADE IN VALUE OF OLD CAR                          500
SALVAGE VALUE OF OLD CAR                                   500
SALVAGE VALUE OF NEW CAR                                   0
ESTIMATED REMAINING LIFE OF OLD CAR YEARS                  4
ESTIMATED LIFE OF NEW CAR YEARS                            10
PURCHASE PRICE OF NEW CAR $                                9000
% NEW CAR PRICE REQUIRED DOWN PAYMENT                      10
LIFE OF LOAN, YEARS                                        3
NUMBER OF LOAN PAYMENTS PER YEAR                           12
YEARLY INTEREST RATE %                                     14
RISK FREE RETURN ON BEST INVESTMENT % YR                   10
ESCLATION RATE FOR FUEL PRICE FOR OLD CAR %/YR             8
ESCLATION RATE FOR FUEL PRICE FOR NEW CAR %/YR             8
ESCLATION RATE FOR MAINTANENCE AND INSURANCE %/YR          4
ESCLATION RATE FOR NEW CAR PRICE % PER YEAR                5
yOUR MARGINAL INCOME TAX RATE                              0


$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THE TIME GO GET A NEW CAR IS  16  MONTHS

ON THE NEXT PAGE VALUE IS THE NET PRESENT VALUE OF REPLACING THE
OLD CAR J MONTHS FROM NOW INSTEAD OF NOW
TC(OLD) IS THE PRESENT VALUE OF THE COST OF THE OLD CAR FOR J MONTHS
TC(NEW) IS THE SAME FOR THE NEW CAR AND FUEL(OLD)-FUEL(NEW) IS
THE SAVING IN FUEL THAT RESULT FROM REPLACING THE OLD CAR NOW.
```

```
F(Old) =    D·P(Old)/MPG(Old)·(1 − (1/(1 + r − f)↑N))/(r − f)
F(New) =    D·P(New)/MPG(Old)·(1 − (1/(1 + 2 − f)↑N))/(r − f)
M(Old) =    R(Old)·(1 − (1/(1 + r − e)↑N))/(r − e)
M(New) =    R(New)·(1 − (1/(1 + r − e)↑N))/(r − e)
IN(Old) =   I(Old)·(1 − (1/(1 + r − e). N))/(r − e)
IN(New) =   I(New)·(1 − (1/(1 + r − e)↑N))/(r − e)
S(Old) =    S/(1 + r)↑N
C(New) =    CP·(1 − (1/(1 + r − c)↑N))·(1 − (1 − sv)/((1 + c)↑12·L) − 1)) − SV·(1 − (1/(1 + r − x)↑N))
Down =      d·C(New)
IP(New) =   (1 − d)·C(new)·i/(1 − (1/(1 + 1)↑M))·(1 − (1/(1 + r)↑M))/r
TD(New) =   T·(IP(New) − (1 − d)·C(New)·(df1 − I)/((1 + 1·df2))
df1 =       i/(1 − (1/(1 + i)↑M))
df2 =       (r − i)/(1 − (1/(1 + r − i)↑M))
V(Old) =    (new) − F(Old) + IN(New) − IN(Old) + R(New) − R(Old) + Down + IP − S(Old) − TD(New)
```

Where D = distance driven per month, P = price of fuel, R = monthly cost of repairs, I = monthly cost of insurance, S = cost of one time repairs to old car, CP = purchase price of new car, r = discount rate, f = escalation rate of fuel, e = escalation rate for repairs and insurance, c = escalation rate for new car price, i = interest rate for loan, M = number of loan payments, L = life of new car, SV = trade in value of old car, s = escalation rate of trade in value of old car (may be negative), d = fraction of purchase price required for down payment, sv = salvage value of new car.

Table 1. List of Equations

affect anything else in the program and vice versa. We'll also need a main program to keep track of things and allow the various subroutines to communicate with each other. And finally we need a module that lists all of the variables and tells a little bit about what the program is all about. This final module—the documentation module—is *not* a substitute for written documentation.

The program structure is now fixed. The structure is shown in Table 3.

The documentation module is a series of remark statements. As a minimum this module should contain a list of all the variables used in the program and what they are used for.

The main program module is a series of GOSUB statements that transfer control to the other modules as needed.

The first module that requires discussion is the data input module. The data we need are:

- Average miles driven per month.
- Miles per gallon for old car.
- Miles per gallon for new car.
- Price of fuel for old car.
- Price of fuel for new car.
- Escalation rate for fuel, old car.
- Escalation rate for fuel, new car.
- Insurance premium for old car.
- Insurance premium for new car.
- Escalation rate for insurance.
- Average monthly cost of normal maintenance for old car.
- Average monthly cost of maintenance for new car.
- Escalation rate for repairs.
- Estimated cost of any special repairs for old car.
- Purchase price of new car.
- Percent of purchase price required for down payment.
- Yearly interest rate for new car loans.
- Escalation rate for new car prices.
- Number of years of loan.
- Number of loan payments per year.
- Discount rate.

That's a lot of data.

Because the data input section is the main interface between the user and the computer, we need to make sure that you can get your data into the computer and correct any errors. The program will ask for each piece of information needed by name and also indicate what the units are. The easiest way to do this is to set up a string array which contains the names of the data items. We will also set up a numeric array to hold the data.

Now Your Child Can Learn More . . . At a Faster Pace

NEW!

GEOGRAPHY EXPLORER: U.S.A.

This program allows your child to travel the country and learn vital facts about each of our 50 states. Geography Explorer offers the most fascinating way of learning yet. Learn each state's name, capitol, largest city, nickname, etc. As a bonus, this package offers the capability of light pen use. **Order No. 0071RD $49.95**

NEW!

MATH MASTER

MATH MASTER is an educational package designed to help your child understand the mathematical concepts of addition, subtraction, multiplication and division through problem solving. This package also instructs the student in fraction problem solving. MATH MASTER will "walk" your child through the developmental concepts of mathematics with never-ending patience. **Order No. 0257R $9.95**

WORDWATCH

Wordwatch consists of 4 different programs, each designed to enhance your child's understanding of word relationships—this results in an improved vocabulary. In WORDRACE, the student must choose the proper definition for a given word. HIDE 'N SPELL asks the student to find misspellings. In SPELLING BEE, the student takes a pre-recorded quiz in which words are played aloud! Completing the package is SPELLING TUTOR, where words are jumbled, reversed or otherwise altered, and it's up to the student to straighten them out! Together, these four programs will help decrease dictionary dependency! **Order No. 0111R $7.95**

NEW!

OMNI-CALCULATOR

This package is a brilliant solution to the problems involving the complex world of different units of measure. It provides rapid means of conversion from one unit of measure to another in any of 10 categories: length, volume, mass, velocity, area density, power, energy, pressure/stress and temperature. Omni Calculator will give your child a great learning advantage, a head start in understanding the complex relationships between different units of measure. **Order No. 0211RD $19.95**

NEW!

THE ELEMENTS

This program can be used to introduce students to the periodic table of the elements, or for review by students and adults who want to refresh their memory in chemistry. It includes each of the elements' name, atomic number, weight and symbol, as well as its acid/base and normal physical state. Enrich your child's knowledge of science with the help of THE ELEMENTS. **Order No. 0216R $9.95**

LIFE

Even if you've only been involved with computers for a short while, you're sure to have heard of LIFE. This program is the computerized simulation of the life cycle of a colony of bacteria. It is based on a few simple concepts, but the results are captivating, animated graphics and an enlightening introduction into the world of biology and genetics. The LIFE program may be just the motivation your child needs to pursue an interest in the sciences. **Order No. 0078R $9.95**

TO ORDER: For the TRS-80*

See your local Instant Software dealer. If these packages are unavailable, order direct or

CALL TOLL FREE 1-800-258-5473

(We ship within 48 hours)

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

"The microcomputer will be a powerful tool in education. One of its greatest advantages is its interactive nature—the student becomes an active participant in learning, and the computer can give immediate feedback. The student does not have to wait until the lesson is complete or the teacher is free to check his work; the computer can evaluate the work immediately, and reward him if it is correct or explain why if the work is incorrect. The computer can then determine, on the basis of his work, whether the student is ready to go on to the next lesson or needs to review the present one.

"Educational computer programs will be available for students at all levels—from pre-school to post-college, from remedial to gifted—and in any subject. These programs can give a child a headstart in school; they can reinforce and provide practice in concepts from hand-eye coordination to Math to English or even Health. A computer can individualize a program to a child's level and pace. Programs will challenge the gifted child, allowing him to advance at his own pace. Best of all, educational computer programs can make learning fun."

Mary Shooshan, Educational Software Editor and former teacher

*TRS-80 is a trademark of the TANDY CORPORATION.

Instant Software™

All packages are on cassette except for Geography Explorer: USA, Omni-Calculator and Teacher's Aide which are on disk.

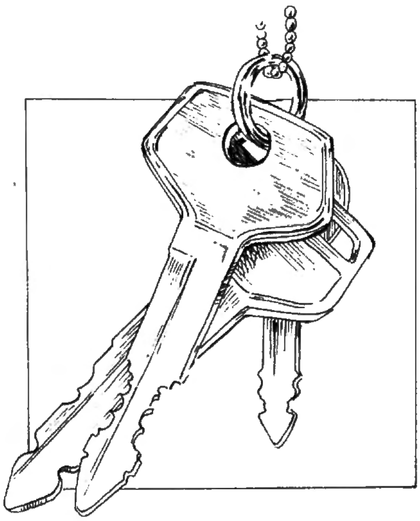
PETERBOROUGH, N.H. 03458
603-924-7296

See List of Advertisers on page 306

"What is the optimum way to spend my car money? The criteria are economic and this optimum is defined in economic terms."

The program will go through a loop that will call up the name of the item that the computer needs and allow the user to input the value. It is also a good idea to print the current value of the variable when we ask for it. This means that both the PRINT and INPUT statements must be used. The PRINT statement asks the question and displays the current value, and the INPUT statement gets the data.

What about errors? After we have gone through the data input loop and obtained all the data, let's clear the CRT and display it all. Now the program asks "Are these data



correct?" If not, then you can re-enter any corrections. If all data are correct control returns to the main program. Calculations cannot begin until you have answered "ARE THESE DATA CORRECT?" with "YES."

The program uses INKEY$ to get your answer. You need only answer Y or N. Pressing any other key repeats the question: please answer Y or N. Since there will be several times when we want a Y or N response. I built a small subroutine to get the Y or N response. This is a good subroutine to have around for other programs.

The module that calculates our equations is straightforward. Notice that the discount factor must be calculated for all of the cash flows. One way to calculate this is be means of the user-defined function in Level III or Disk BASIC. In this case we would write:

```
100 DEF FNDF(R,N)=(1-(1/(1+R)↑N))/R
```

Note that both R and N are dummy variables.

There are two disadvantages to using the user-defined function. First, we limit the transportability of the program to Level III and Disk BASIC users. Secondly, the case of R = 0 cannot be handled.

On the other hand a simple BASIC subroutine can handle the problem R = 0 and still be transportable to other computers. The subroutine would look like this:

```
1000 REM DISCOUNT FACTOR SUBROUTINE
1010 IF R=0 THEN DF=N ELSE DF=(1-(1/(1+R)↑N))/R
1020 RETURN: REM END OF SUBROUTINE
```

To use this approach, R and N must be set by the calling program before the GOSUB instruction.

To display the results, first clear the CRT. Then, print the number of months to keep the old car. Next, the program asks the user if he wants to see detailed results. If so, the program displays a year at a time the value of keeping the old car and the various cost items for each month.

If you need a printed version, the CRT displays the input data, then prints it.

After everything is printed, or, even if you decline the hard copy, the program asks if you want to run another problem. If so control transfers back to the data input module. Otherwise, the programs ends.

Debugging

Program debugging is straightforward. Debug each module as it is written. Compare the calculated discount factors with published tables. When the entire program is written (and after each module has been debugged) test the extremes of data input. If a problem comes up, look for such things as a variable with a three-character name, lowercase in the string statements, bad logic in IF statements, bad input data or failure to convert percent. Most of the debugging should be simple because of the modular construction of the program.

We're done, right? Wrong! We need to document the program.

Program documentation should consist of two parts. A part for the user and a part for the programmer. They need different information about the program and it's up to us to provide it.

The user doesn't care how the program works. He needs to know how to run the program, what data is required, what the results mean and what other applications he has for the program—*without* modifying it.

The programmer needs to know how the program works so he can patch it, if necessary, or so he can customize it. He must know what special tricks you used, so he can translate the program to another computer that doesn't have the same features as yours.

Documentation

Program title: *Car Analysis* by L. E. Sparks

Abstract: This program calculates the optimum time to keep an old car before replacing it with a new, more fuel efficient car. The optimum time is defined as that time which maximizes the present value of keeping the old car. The program can also be used to analyze the economics of purchasing other fuel or energy efficient appliances.

Language: TRS-80 Level II or Disk BASIC.

Hardware: TRS-80 Level II Microcomputer with 16K RAM is minimum. Printer is op-

Name	Description
F VARIABLES	Strings used as labels
X(1)	Miles driven per year
X(2)&X(3)	Fuel price old and new
X(4)&X(5)	MPG old and new
X(6)&X(7)	Insur. old and new
X(8)&X(9)	Maint. old and new
X(10)	Trade in value old
X(11)&X(12)	Salvage value old and new
X(13)&X(14)	Life old and new
X(15)	Price new car
X(16)	Percent down required
X(17)	No. yrs of loan
X(18)	No. paymets per yr
X(19)&X(20)	Int. and discount rates
X(21)&X(22)	Escalation fuel old and new
X(23)	Escalation insur. and maint.
X(24)	Escalation new car price
X(25)	Income tax rate
X1–X4	Effective disc. rate monthly
DR	Monthly discount rate
DX	Fract. down
YM	Max. life old car
JM	Optimum life old car
VM	Max. present value old
M	No. car payments
L	Monthly loan rate
V(J)	NPV of old car for J mo.
TL(J)	Total cost of old for J mo.
TH(J)	Total cost of new for J mo.
XL(J)	Total fuel old for J mo.
XH(J)	Total fuel new for J mo.
N&R	Dummy variables
IL&IH	Monthly insur. costs
RL&RH	Monthly repair costs
Functions Used	
INKEY$	Get single character
LPRINT	Line printer command
CHR$(29)	Print normal letters
CHR$(28)	Print large letters
CHR$(140)	Top of form

Table 2. Symbol Table

tional.

User instructions: The program will analyze the economics of investing in a fuel efficient car as opposed to keeping your old car. The program calculates the present value of keeping the old car at the end for N months, where N varies from one to the remaining life of the old car. The time to replace the old car is at the end of the month with the highest present value. Note that if the present value is always negative, the time to replace the car is now.

- Documentation module
- Main program module (used to transfer control)
- Initialization module (used to initialize everything)
- Data input module (used to get data from keyboard)
- Calculation module
- Display module (used to display result on CRT)
- Line printer module (used to print results on printer)

Table 3. Program Structure Outline

To use the program load it into the computer, type RUN and press ENTER. In a few seconds the message CAR ANALYSIS BY LESLIE E. SPARKS VERSION 1.0 PLEASE ENTER THE DATA ASKED FOR. IF YOU MAKE A MISTAKE DON'T WORRY. YOU WILL HAVE A CHANCE TO CORRECT IT BEFORE CALCULATIONS START. will be displayed on the CRT. Enter the requested data and press ENTER after each item. If you make a mistake and find it before you have pushed ENTER, use the back arrow to correct it. Otherwise, keep going; you will have a chance to correct the error later.

After all of the data have been entered, they will be displayed in two blocks. You will be asked if the data are correct. If they are not, press N, and the program will allow you to re-enter any incorrect data. When all of the input data are correct, press Y. You cannot get into the calculation portion of the program until you have answered the question ARE THESE DATA CORRECT? by pressing Y.

As soon as you press Y, the computer starts calculating. The calculations take

Program Listing. Should I Buy a New Car?

```
10 REM SHOULD I BUY A NEW CAR? TRS-80 LEVEL II BY L.E.S
     PARKS
20 REM DISK FILE NAME CAR VERSION 1.0 3 MAY 1980
30 REM ALL F VARIABLES ARE STRINGS AND ARE USED FOR LAB
     ELS
40 REM X(1)=MILES DRIVEN PER YEAR,X(2)=PRICE OF FUEL OL
     D CAR
50 REM X(3)=PRICE OF FUEL NEW CAR,X(4)=MPG OLD,X(5)=MPG
      NEW
X(6)=INSURANCE OLD,X(7)=INSURANCE NEW,X(8)=MA
     INT OLD,
X(9)=MAINT NEW,X(10)=TRADE IN VALUE OLD,X
     (11)=SALVAGE VALUE OLD
X(12)=SALVAGE VALUE NEW,X(1
     3)=REMAINING LIFE OLD,X(14)=LIFE NEW
60 REM X(15)=PRICE NEW CAR, X(16)=%DOWN PAYMENT, X(17)=
      LIFE
OF LOAN, X(18)=PAYMENT PER YEAR,X(19)=INTER
     EST RATE,X(20)
DISCOUNT RATE, X(21)&X(22)=ESCLATIO
     N RATE FUEL PRICES OLD AND NEW
X(23)=ESCLATION INS
     UR&MAINT,X(24)=ESCLATION NEW CAR
70 REM X(25)=MARGINAL INCOME TAX RATE, X1-X4=EFFECTIVE
     MONTHLY
ESCLATION RATE IN FRACTION,DR=MONTHLY DISC
     OUNT RATE FRACTION,
DX=DOWNPAYMENT AS FRACTION,YM=
     MAX LIFE OLD CAR IN MONTHS, VM=
MAX NET PRESENT VA
     LUE OLD CAR
80 REM JM=OPTIMUM NUMBER OF MONTHS TO KEEP OLD CAR,M=NU
     MBER
OF CAR PAYMENTS, L=MONTHLY INTEREST RATE,TL(J
     )=TOTAL PRESENT
VALUE OF COST OF OLD CAR FOR J MON
     THS, TH(J) SAME FOR NEW CAR
90 REM XL(J)&XH(J) TOTAL PRESENT VALUE OF FUEL COSTS FO
     R OLD AND
```

Program continues

"A reasonable discount rate is the maximum interest or return that you can earn on a risk free investment."

several seconds.

As soon as the calculations are complete, the optimum time to keep your old car is displayed on the CRT along with the question DO YOU WANT DETAILS? Answer the question by pressing Y or N. You do not need to press ENTER. If you want the details, the results of the calculations for each month will be displayed, one year at a time. To go on to the next year press any key.

Regardless of how you answer the question about details, you will be asked DO YOU WANT HARD COPY? Again answer Y or N. If you want hard copy, the input data and all of the information that was displayed on the CRT will be printed.

The last question you will be asked is DO YOU WANT TO RUN ANOTHER PROBLEM? Answer Y or N as before. If you answer Y, you will find yourself back in the data input portion of the program. All of the data you used in the previous calculation is still in the computer. When you get in the data input loop, you will find these data in parenthesis. If you want to keep the value displayed, press ENTER; otherwise enter the new data. If you answer N, the program run will be terminated and control returned to BASIC as indicated by READY.

Although the program is specifically written to provide an answer to the new car question, it can analyze the economics of other investments involving a choice of keeping an old, less efficient device and replacing it with a new, energy efficient device. The input data have to be redefined but no programming changes are required.

Interpretation of results: The program uses present value as the criterion for determining the economic advisability of replacing the old car. The concept of present value is used in most investment analyses and is a way of comparing cash flows with different timing on a common basis. The sooner a dollar is received, the higher its present value is. The best investment is the one that maximizes the present value. Thus the optimum time to get rid of the old car is at the end of the month that has the highest present value. Note that if the present value of keeping the old car is always negative, the time to replace it is now.

The present value of a cash flow is a function of the discount rate used to calculate the present value. A reasonable discount rate is the maximum interest or return that you can earn on a risk free investment. The program uses a default of 10 percent per

```
NEW CARS FOR J MONTHS, IL&IH MONTHLY INS
     URANCE COSTS FOR OLD&NEW
100 REM RL&RH=MONTHLY REPAIR COSTS FOR OLD AND NEW CARS
110 REM DP(J)=PRESENT VALUE OF DOWNPAYMENT COST AT END
     OF J MO.
120 REM IP(J)=PRESENT VALUE OF CAR PAYMENTS FOR J MONTH
     S
130 REM TD TAX DEDUCTION FOR INTEREST
140 CLEAR 500:REM CLEAR STRING SPACE
150 GOTO 190  :REM TRANSFER TO MAIN PROGRAM
160 REM SUBROUTINE TO CALCULATE DISCOUNT FACTOR
170 IF R = 0 THEN DF=N ELSE DF = (1-(1/(1+R)[N))/R
180 RETURN
190 REM MAIN PROGRAM
200 GOSUB 350  :REM INITIALIZE
210 GOSUB 900  : REM GET INPUT DATA
220 GOSUB 1190 :REM DO CALCULATIONS
230 GOSUB 1790 : REM DISPLAY RESULTS
240 PRINT"DO YOU WANT HARD COPY?"
250 GOSUB 290  : REM GET INPUT FROM KEYBOARD
260 IF Y$ ="Y" THEN GOSUB 2050 :REM PRINT IT ON LINEPRI
     NTER
270 PRINT" DO YOU WANT ANOTHER CASE?"
280 GOSUB 290  : IF Y$="Y" THEN 210  ELSE END
290 REM SUBROUTINE TO GET Y OR N FROM KEYBOARD
300 Y$=INKEY$:IF Y$=""THEN 300
310 IF Y$ ="Y" OR Y$ ="N" THEN 340
320 PRINT"PLEASE ANSWER Y OR N"
330 GOTO 300
340 RETURN
350 REM INITIALIZE EVERYTHING
360 DEFSTR F
370 DEFINT J
380 DR=0:X1=0:X2=0:X3=0:X4=0:YM=0:VM=0:M=0:L=0:D=0:IL=0
     :J=0:IH=0
390 RL=0:RH=0:C=0:N=0:R=0:DF=0:IP=0:IK=0:REM INITIALIZE
     VARIABLES
USED LATER FOR SPEED
400 DIM X(25),F(25),V(120),TH(120),TL(120),IL(120),IH(1
     20),RL(120),RH(120),IP(120)
410 DIM XL(120),XH(120),DP(120)
420 F(1)="MILES DRIVEN PER YEAR"
430 X(1)=15000:X(2)=1.3:X(3)=1.3:REM SET UP DEFAULTS
440 F(2)="CURRENT PRICE OF FUEL FOR OLD CAR $/GAL"
450 F(3)="CURRENT PRICE OF FUEL FOR NEW CAR $/GAL"
460 F(4)="MILES PER GALLON OLD CAR"
470 X(4)=12
480 F(5)="MILES PER GALLON NEW CAR"
490 X(5)=40
500 F(6)="INSURANCE COST OLD CAR $/YR"
510 X(6)=150
520 F(7)="INSURANCE COST NEW CAR $/YR"
530 X(7)=350
540 F(8)="MAINTANENCE COST OF OLD CAR PER YEAR"
550 X(8)=250
560 F(9)="MAINTANENCE COST OF NEW CAR PER YEAR"
570 X(9)=100
580 F(10)="PRESENT TRADE IN VALUE OF OLD CAR"
590 X(10)=500
600 F(11)="SALVAGE VALUE OF OLD CAR"
610 F(12)="SALVAGE VALUE OF NEW CAR"
620 X(11)=500
630 F(13)="ESTIMATED REMAINING LIFE OF OLD CAR YEARS"
640 X(13)=4
650 F(14)="ESTIMATED LIFE OF NEW CAR YEARS"
660 X(14)=10
670 F(15)="PURCHASE PRICE OF NEW CAR $"
680 X(15)=9000
690 F(16)="% NEW CAR PRICE REQUIRED DOWN PAYMENT"
700 X(16)=10
710 F(17)="LIFE OF LOAN, YEARS"
720 X(17)=3
730 F(18)="NUMBER OF LOAN PAYMENTS PER YEAR "
740 F(19)="YEARLY INTEREST RATE %"
750 X(18)=12:X(19)=14:X(20)=10
760 F(20)="RISK FREE RETURN ON BEST INVESTMENT % YR"
770 F(21)="ESCLATION RATE FOR FUEL PRICE FOR OLD CAR %/
     YR"
780 F(22)="ESCLATION RATE FOR FUEL PRICE FOR NEW CAR %/
     YR"
790 F(23)="ESCLATION RATE FOR MAINTANENCE AND INSURANCE
     %/YR"
```

Program continues

When DecisionMaster™ speaks everybody listens.

Let's face it. We all have to make decisions. Decisions that can change our lives. Decisions that can make us happy or unhappy. Decisions that could win us fame or fortune. Now, DecisionMaster can help you make the best decisions of your life.

Use Bayesian theory to peer into the future...even if you've never heard of the Bayes' Rule. Do a complete weighted factor analysis...without knowing what one is. Use discounted cash flow to compare investment alternatives without bothering with present value tables. These and other sophisticated theories that were once the exclusive domain of professors and top business executives are *built into* DecisionMaster's algorithms... so you can use them at the touch of a key!

DecisionMaster is easy to use. It features:

- A fully documented manual developed by an authority in the field.
- A unique program-controlled cross reference system.
- A powerful formatted-screen data entry system.

You'll use DecisionMaster in hundreds of routine decisions, as well as more important ones such as • Buying a house • Changing jobs • Selecting investment • Evaluating insurance policies • Expanding product lines • Leasing vs. purchasing.

If you buy only one computer program this year, make it DecisionMaster. And when it speaks, *listen.*

DECISION?MASTER™

To order, see your software dealer or return this form with your check to: DecisionMaster, Dept. 80, 10428 Westpark, Houston, Texas 77042 Add $1.50 for shipping and handling

Diskette for: ☐ Apple II* (32K) $29.95 ☐ TRS-80** (Level II-32K) $29.95 ☐ TRS-80** (Model II) $39.95

Name

Address

City State Zip

CHARGE by phone toll-free: 1-800-231-5768 Ext. 306 (In Texas: 1-800-392-2348) or return this form:

☐ VISA ☐ MasterCard Bank Code

Account No.

Signature

Expiration Date

*Apple II is a registered trademark of Apple Computers, Inc.
**TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

✓302

```
800 F(24)="ESCLATION RATE FOR NEW CAR PRICE % PER YEAR"
810 F(25)="yOUR MARGINAL INCOME TAX RATE"
820 X(21)=8:X(22)=8:X(23)=4:X(25)=00:X(24)=5
830 REM NOTE THAT THE DEFAULT VALUES ARE THE SAME AS TH
     E EXAMPLE
840 REM NOTE THAT THE EXAMPLE IGNORES INCOME TAXES IF Y
     OU RUN
    THE EXAMPLE WITH A 50% TAX RATE THE TIM
     E TO REPLACE THE CAR IS NOW
850 F1="THE FIRST 12"
860 F2="tHE FINAL GROUP CHECK CAREFULLY"
870 J1=1
880 J2 = 12
890 RETURN
900 REM INPUT DATA
910 CLS:PRINT"CAR ANALYSIS BY LESLIE E. SPARKS VERSION
     1.0"
920 PRINT"PLEASE ENTER THE DATA ASKED FOR"
930 PRINT"DO NOT WORRY IF YOU MAKE A MISTAKE"
940 PRINT"YOU WILL HAVE A CHANCE TO CORRECT THE MISTAKE
      LATER."
950 PRINT"THE CURRENT VALUE OF THE VARIABLE IS SHOWN IN
      ()"
960 FOR J= 1 TO 25
970    PRINT F(J);"(";X(J);")";
980    INPUT X(J)
990 NEXT J
1000 J1=1:J2=12:F=F1
1010 CLS
1020 PRINT F
1030 FOR J=J1 TO J2
1040    PRINT J;" ";F(J);X(J)
1050 NEXT J
1060 PRINT"ARE THESE CORRECT"
1070 GOSUB 290
1080 IF Y$="Y" THEN 1140 ELSE INPUT"WHAT ITEM NUMBER DO
      YOU WANT TO CHANGE ";JC
1090 IF JC < J1 OR JC>J2 PRINT" i CAN'T CORRECT THAT ON
     E" ELSE 1120
1100 FOR JJ= 1 TO 800:NEXTJJ
1110 GOTO1010
1120 INPUT"ENTER CORRECT VALUE ";X(JC)
1130 GOTO1010
1140 IF J1>1 THEN RETURN ELSEF=F2
1150 J1 = J2+1
1160 J2=25
1170 PRINT F
1180 GOTO1010
1190 REM SUBROUTINE TO CALCUALTE PRESENT VALUE OF KEEPI
     NG OLD CAR V(J)
1200 REM SET UP EFFECTIVE DISCOUNT RATES
1210 DR=X(19)/12/100:REM MONTHLY DISCOUNT RATE DECIMAL
     FRACTION
1220 X1=DR-(X(22)/12/100):REM EFFECTIVE RATE FOR OLD CA
     R FUEL
1230 X2=DR-(X(23)/12/100):REM EFFECTIVE RATE FOR NEW CA
     R FUEL
1240 X3=DR-(X(23)/12/100):REM EFFCTIVE RATE FOR INSUR A
     ND MAINT
1250 X4=DR-X(24)/12/100:REM EFFECTIVE RATE FOR NEW CAR
1260 DX=X(16)/100
1270 YM=12*X(13):REM MAX LIFE OF OLD CAR IN MONTHS
1280 VM=0:REM INITIALIZE MAX VALUE
1290 M=X(17)*X(18):REM NUMBER OF LOAN PAYMENTS
1300 L=X(16)/12/100:REM MONTHLY INTEREST RATE FOR LOAN
1310 D=X(1)/12:REM MONTHLY MILES DRIVEN
1320 IL=X(6)/12:REM MONTHLY INSURANCE FOR OLD CAR
1330 IH=X(7)/12:REM MONTHLY INSRUANCE FOR NEW CAR
1340 RL=X(8)/12:REM MONTHLY MAINT. FOR OLD CAR
1350 RH=X(9)/12:REM MONTHLY MAINT. FOR NEW CAR
1360 C=X(15):REM COST OF NEW CAR
1370 SH=1-(X(15)-X(12))/X(15):REM SALVAGE VALUE AS FRAC
     TION OF PRICE
1380 SL=(X(11)-X(10))/X(10)/(X(13)*12):SL=S-SL
1390 REM NOW START CALCULATIONS
1400 CLS:PRINT "STARTING CALCULATIONS NOW "
1410 FOR J=1 TO YM+1
1420       REM CALCULATE ALL COSTS FOR OLD CAR
1430       N=J:R=X1
1440       GOSUB160
1450       XL(J)=D*X(2)/X(4)*DF
```

Program continues

"The program is broken into modules—documentation, initialization, data input, calculation, CRT display, and hard copy."

year.

Sample problem: The input data are as follows.

miles driven = 15000 miles per year.
miles per gallon old car = 12
miles per gallon new car = 40
price of fuel for old car = $1.30 per gallon
price of fuel for new car = $1.30 per gallon
escalation rate for fuel = 8 percent per year for both old and new cars
normal maintenance for old car = $350 per year
normal maintenance for new car = $250 per year
insurance for old car = $150 per year
insurance for new car = $350 per year
escalation rate for maintanence and insurance = 4.0 percent per year
price of new car = $9000
down payment = 10 percent of purchase price
escalation rate for new car price = 5.0 percent per year
interest rate = 14 percent per year
number of years of loan = 3
number of payments per year = 12
discount rate = 10 percent per year
trade-in value of old car = $500
salvage value of old car = $0
salvage value of new car = $0

See Fig. 1 for the results of the example.

The program: The program is broken into modules—documentation, initialization, data input, calculation, CRT display, and hard copy. Communication between modules is handled by a small main program.

The documentation module consists of remark statements only. The purpose of this module is to provide minimum documentation of the program. It contains the program title, version number, author, and a variable list. All of these lines except the credit line can be removed from a working program.

The initialization module clears string space, sets up the arrays, and provides labels for data input and out strings. If investments other than the car example are to be analyzed, the labels can be changed to reflect the new data requirements. In general, no other changes will be required in the program.

Data are entered from the keyboard using INPUT. The program goes through a loop where the data are asked for by item name. The current value of the variable is shown in parenthesis.

The calculation module is also straightforward. If your BASIC allows user-defined functions, they can be used to calculate the discount factor. The discount factor for a series of equal payments is calculated in a subroutine. The equations which are solved by the program are shown in Table 1.

The CRT display module is straightforward and requires no comment.

```
1460       R=X3
1470     GOSUB160
1480      IL(J)=DF*IL
1490      RL(J)=DF*RL
1500      TL(J)=XL(J)+IL(J)+RL(J)
1510     R=X2
1520     GOSUB160
1530     XH(J)=D*X(3)/X(5)*DF
1540     R=X3
1550      IH(J)=IH*DF
1560      RH(J)=RH*DF
1570    N=J:R=X4
1580    GOSUB160
1590    CC=C*X4*DF*(1+(1-SH)/((1+X4)[(12*X(14))-1))
1600    CC=CC-X(10)*(1-(1+SL)[-N)
1610      DP(J)=CC*DX
1620    R=L:N=M:GOSUB160
1650    IP=CC*(1-DX)/DF
1660      R=DR
1680      GOSUB160
1690    IP=IP*DF
1700      IP(J)=IP
1710 N=M:R=L:GOSUB160  :D1=1/DF:R=DR-L:GOSUB160  :D2=1/
     DF
1720    TD=X(25)/100*(IP-(1-DX)*CC*(D1-L)/((1+L)*D2))
1730      TH(J)=IP(J)+XH(J)+IH(J)+RH(J)-TD
1740      V(J)=TH(J)-TL(J)
1750      IF V(J)>VM THEN VM=V(J) ELSE 1770
1760      JM=J
1770 NEXTJ
1780 RETURN
1790 REM SUBROUTINE TO DISPLAY RESULTS ON THE CRT
1800 CLS
1810 PRINT"   CAR ANALYSIS BY LESLIE E. SPARKS VERSION
     1.0"
1820 PRINT
1830 PRINT@240," "
1840 IF VM<=0 THEN PRINT TAB(20)"THE TIME TO GET A NEW"
     :PRINT TAB(20)"CAR IS NOW!!!" ELSE 1860
1850 GOTO 1880
1860 PRINT TAB(20)"THE TIME TO GET A NEW CAR IS IN ";JM
     ;" MONTHS"
1870 PRINT
1880 PRINT:PRINT:PRINT:PRINT"DO YOU WANT TO SEE DETAILS
     ?"
1890 GOSUB290
1900 IF Y$="Y" THEN 1910 ELSE RETURN
1910 CLS
1920 PRINT "DETAILS OF CALCULATIONS "
1930 J=0
1940 FOR JJ=1 TO YM/12
1950 CLS
1960 PRINT"MONTH    VALUE";TAB(19);"TL";TAB(28);"TH";TA
     B(37);"FUEL SAVINGS"
1970     FOR JK = 1 TO 12
1980 J=J+1
1990     PRINT J;TAB(9)V(J);TAB(17);TL(J);TAB(28);TH(J);
     TAB(38);XH(J)-XL(J)
2000   NEXT JK
2010 PRINT"PRESS ANY KEY FOR NEXT YEAR"
2020 Y$=INKEY$:IF Y$=""THEN 2020
2030 NEXT JJ
2040 RETURN
2050 REM LINE PRINTER SUBROUTINE
2060 LPRINT:LPRINT:LPRINT:LPRINT:LPRINT:REM IF YOU HAVE
      RADIO SHACK PRINTER CHANGE TO LPRINT" ":LPRINT" "
      :LPRINT" "
2070 LPRINT TAB(10) STRING$(68,"$")
2080 REM FIRST PRINT OUT INPUT DATA
2090 LPRINT TAB(30)CHR$(28);"CAR ANALYSIS";CHR$(29)
2100 LPRINT TAB(10)STRING$(68,"$")
2110 LPRINT TAB(30) CHR$(28);"INPUT DATA";CHR$(29): REM
      LARGE PRINT THEN SMALL PRINT
2120 FOR J=1TO25
2130     LPRINT TAB(10)  F$(J);TAB(60)"        ";X(J)
2140 NEXT J
2150 LPRINT:LPRINT:LPRINT
2160 LPRINT TAB(10)STRING$(68,"$")
2170 IF VM<=0 THEN LPRINT TAB(10) CHR$(28);"THE TIME TO
      GET A NEW CAR IS NOW!";CHR$(29):GOTO2230
2180 LPRINT TAB(10)"THE TIME GO GET A NEW CAR IS ";JM;"
      MONTHS"
```

Program continues

"INKEY$ allows the user to enter a single character from the keyboard without pressing enter."

The line printer module uses two commands peculiar to the Microtek MT-80P printer—LPRINT CHR$(28); (large characters) and LPRINT CHR$(29); (normal characters) to control the character size. If your printer allows software control of character size, replace these two commands with the appropriate commands for your printer. If not, delete these two commands.

There are several times when the user is asked to answer a question either Y or N. The user's answer to these questions is handled with INKEY$. INKEY$ allows the user to enter a single character from the keyboard without pressing enter. If your BASIC does not support this type of command, replace it with a normal INPUT command.

A variable list and function table are given in Table 2.

Note that in the program listing the bracket appears. This is the character that the printer uses for the up arrow.■

```
2190 LPRINT
2200 LPRINT TAB(10)"ON THE NEXT PAGE VALUE IS THE NET P
     RESENT VALUE OF REPLACING THE
2210 LPRINT TAB(10)"OLD CAR J MONTHS FROM NOW INSTEAD O
     F NOW"
2220 LPRINT TAB(10)"TC(OLD) IS THE PRESENT VALUE OF THE
      COST OF THE OLD CAR FOR J MONTHS
";TAB(10);"TC(NE
     W) IS THE SAME FOR THE NEW CAR AND FUEL(OLD)-FUEL(
     NEW) IS
";TAB(10);"THE SAVING IN FUEL THAT RESULT
     FROM REPLACING THE OLD CAR NOW."
2230 LPRINTCHR$(140):REM GET NEW PAGE
2240 REM NOTE YOUR PRINT MAY USE A DIFFERENT COMMAND FO
     R TOP OF FORM.
2250 LPRINT
2260 LPRINT TAB(5)STRING$(68,"$")
2270 LPRINTTAB(5) CHR$(28);"DETAILED RESULTS OF THE CAL
     CULATIONS":REM LARGE PRINT
2280 LPRINTCHR$(29);TAB(5) STRING$(68,"$")
2290 LPRINTCHR$(29);TAB(5)"MONTH";TAB(15);"VALUE";TAB(2
     6);"TC(OLD)";TAB(43);"TC(NEW)" ;TAB(58);"FUEL(OLD)
     -FUEL(NEW)
2300 LPRINT TAB(5)STRING$(68,"$")
2310 FOR J= 1 TO YM
2320    LPRINT TAB(5) J;TAB(12)V(J);TAB(26);TL(J);TAB(4
     0)TH(J);TAB(60);XL(J)-XH(J)
2330 NEXT
2340 RETURN
```

A calculating utility for those trying to put a roof over their heads.

The Home Buyer's Helper

Clarence Stinson
17315 Ruette Abeto
San Diego, CA 92127

With mortgage interest rates fluctuating as they are, what will it cost you to buy a home? Your TRS-80 can become your home buyer's calculator. With this program you can estimate the monthly mortgage payments and closing costs at interest rates varying from nine to eighteen percent.

It is surprising the difference interest rates make in a mortgage payment; you would do well to run estimates at home before confronting a realtor. Find the price range of houses that appeal to you, the range of interest rates, required down payments, loan lengths, property taxes, etc. Punch these figures into the TRS-80, as shown in Example 1, and see the results, as in Example 2.

Don't like what you see? Try different inputs: a lower interest rate; higher down payment; extended loan. You probably won't find a painless solution, but you'll go house hunting with a better knowledge of the options.

IRS to the Rescue

The Internal Revenue Service is usually cast as the villain in the drama of the average guy's economic life. The Home Buyer's Calculator can recast it in the role of hero when you are considering a home of your own. By itemizing your deductions you can receive a rebate each year based on your annual mortgage interest, property taxes and your current tax bracket. The higher each of these items is, the more the rebate! The Home Buyer's Calculator estimates the total rebate you can expect. It also figures the rebate as a deduction from your monthly mortgage payment to help you see what you will actually be paying.

Fill in the Blanks

Make a trial run through the program. Example 1 illustrates the first screen display, with questions to be answered. Interest rate must represent a fraction of a percent in decimal form. The program includes (lines 650–710) tables of monthly payments per $1000 for percentages ranging from nine to eighteen percent. The tables come from the standard realtors *Blue Book* and cover 20, 25 and 30 year loans. The program interpolates fractional percentages.

Estimating real estate taxes will vary with your locale. Closing costs are also variable and include such things as escrow fee, credit report, property inspection, etc. A ballpark estimate of one to 1.5 percent is reasonable, unless the buyer is required to pay points to obtain a loan. In that case, add the points (one point = one percent) to the normal closing cost.

After entering the last item in Example 1, there is a pause, then the screen shown in

Calculate the monthly mortgage payments and closing costs.
Enter the values requested.
On repeat runs, if the value in unchanged, simply press 'ENTER'.
Sale Price? <u>60000</u> *
Down Payment? <u>6000</u>
Interest Rate (9–18)? <u>11.25</u>
Number of Years for Loan (20, 25, 30)? <u>25</u>
Annual Real Estate Taxes? <u>600</u>
Closing Cost as % of Sale Price? <u>1</u>

**User inputs are underlined.*

Example 1. Menu for Home Buyer's Calculator

Calculate the monthly mortgage payments and closing costs.

Sale Price	60,000.00
Loan Amount	54,000.00
Interest Rate	11.25
Principle & Interest	539.59
Taxes	50.00
Hazard Insurance	10.95
* Total Monthly Payment *	600.54
Down Payment	6000.00
Tax Impound (6 months)	300.00
Insurance Impounds (12 months)	131.40
Closing Costs	540.00
* Total Cash Required *	6,971.40

Example 2. Calculation of Monthly Payment and Cash Required

Estimated IRS Rebate Based on the Following:
—Current Property Tax
—Average Mortgage Interest for First 5 Years
—Your Current Tax Bracket
Enter following values from IRS Form 1040:
Adjusted Gross Income (line 31)? <u>21000</u>
Itemized Deductions (line 33) or 0? <u>3850</u>
Total Income Tax (line 37)? <u>2460</u>

Example 3. Input for Income Tax Rebate

Estimated Annual IRS Rebate	938.28
Monthly Mortgage Payment	600.54
Estimated Monthly IRS Rebate	78.19
Net Monthly Mortgage Payment	522.35

Example 4. Calculation of IRS Rebate

```
10 REM HOME BUYER'S CALCULATOR
20 REM AUTHOR: C.J.STINSON, MAY 1980
30 CLEAR:E$="###,###.##"
40 DEFINT H,J,K
50 CLS:PRINT TAB(15)"*** HOME BUYER'S CALCULATOR ***":PRINT
60 PRINT"CALCULATE THE MONTHLY MORTGAGE PAYMENTS AND CLOSING COST."
70 PRINT"ENTER THE VALUE REQUESTED."
80 PRINT"ON REPEAT RUNS,IF THE VALUE IS UNCHANGED, SIMPLY PRESS 'ENTER'."
90 PRINT:INPUT"SALE PRICE";S
100 INPUT"DOWN PAYMENT";D
110 L=S-D
120 INPUT"INTEREST RATE (9-18)";I
130 IF I<9 GOTO 150
140 IF I<=18 GOTO 160
150 PRINT"INVALID":GOTO 120
160 INPUT"NUMBER OF YEARS FOR LOAN (20,25,30)";Y
170 IF Y=20 THEN Z=0:GOTO 210
180 IF Y=25 THEN Z=10:GOTO 210
190 IF Y=30 THEN Z=20:GOTO 210
200 PRINT"INVALID":GOTO 160
210 GOSUB 730
220 INPUT"ANNUAL REAL ESTATE TAXES";T
230 INPUT"CLOSING COST AS % OF SALE PRICE";C
240 CLS:PRINT"SALE PRICE";TAB(30)USINGE$;S
250 PRINT"LOAN AMOUNT";TAB(30)USINGE$;L
260 PRINT"INTEREST RATE";TAB(30)USINGE$;I
270 PRINT"PRINCIPLE & INTEREST";TAB(30)USINGE$;M
280 TX=T/12:PRINT"TAXES";TAB(30)USINGE$;TX
290 N=.00219*S/12
300 PRINT"HAZARD INSURANCE";TAB(30)USINGE$;N
310 TT=M+TX+N:PRINT"* TOTAL MONTHLY PAYMENT *";TAB(30)USINGE$;TT
320 PRINT:PRINT"DOWN PAYMENT";TAB(30)USINGE$;D
330 PRINT"TAX IMPOUND (6 MONTHS)";TAB(30)USINGE$;TX*6
340 PRINT"INSURANCE IMPOUND (12 MONTHS)";TAB(30)USINGE$;N*12
350 CC=C/100*L:PRINT"CLOSING COST";TAB(30)USINGE$;CC
360 TM=N*12+TX*6+CC+D
370 PRINT"* TOTAL CASH REQUIRED *";TAB(30)USINGE$;TM
380 INPUT"PRESS 'ENTER' TO CONTINUE.";A$
390 PRINT"TO REPEAT THE HOME-BUYER'S CALCULATOR ENTER (H)."
400 INPUT"TO ESTIMATE INCOME TAX REBATE ENTER (R).";A$
410 IF A$="H" GOTO 50
420 IF A$<>"R":PRINT"INVALID":GOTO 390
430 CLS:PRINT TAB(15)"*** INCOME TAX REBATE ***":PRINT
440 PRINT"ESTIMATED IRS REBATE BASED ON THE FOLLOWING:"
450 PRINT"    -CURRENT PROPERTY TAX
460 PRINT"    -AVERAGE MORTGAGE INTEREST FOR FIRST 5 YEARS"
470 PRINT"    -YOUR CURRENT TAX BRACKET.":PRINT
480 PRINT"ENTER FOLLOWING VALUES FROM IRS FORM 1040:"
490 INPUT"ADJUSTED GROSS INCOME (LINE 31)";B
500 INPUT"ITEMIZED DEDUCTIONS (LINE 33) OR 0";DE
510 INPUT"TOTAL INCOME TAX (LINE 37)";IT
520 BD=B-DE:PC=IT/BD:LL=L:PP=0:AA=0
530 FOR H=1 TO 60
540 AI=LL*I/1200:AA=AA+AI
550 PP=M-AI:LL=LL-PP:NEXT H
560 AA=AA/5+T 'AVERAGE ANUALNUAL INTEREST & TAX
570 AR=AA*PC 'ANNUAL REBATE
580 CLS:PRINT"ESTIMATED ANNUAL IRS REBATE";TAB(30)USINGE$;AR
590 PRINT:PRINT"MONTHLY MORTGAGE PAYMENT";TAB(30)USINGE$;TT
600 PRINT"ESTIMATED MONTHLY IRS REBATE";TAB(30)USINGE$;AR/12
610 PRINT TAB(30)"  -------"
620 PRINT"NET MONTHLY MORTGAGE PAYMENT";TAB(30)USINGE$;TT-AR/12
630 PRINT:INPUT"PRESS 'ENTER' TO CONTINUE";A$
640 GOTO 50
650 REM MONTHLY PAYMENTS PER $1000 FOR 9% TO 18%
660 DATA 9.00,9.66,10.33,11.02,11.72
670 DATA 12.44,13.17,13.92,14.59,15.44
680 DATA 8.40,9.09,9.81,10.54,11.28
690 DATA 12.04,12.81,13.59,14.38,15.18
700 DATA 8.05,8.78,9.53,10.29,11.07
710 DATA 11.85,12.65,13.45,14.26,15.08
720 REM CALCULATE MONTHLY PAYMENT
730 RESTORE:J=I:K=J-8+Z
740 FOR H=1 TO K:READ M:NEXT H
750 ID=I-J:IF ID=0 GOTO 770
760 READ N:M=(N-M)*ID+M
770 M=L/1000*M:RETURN
```

Program Listing

Example 2 appears. It presents the estimated monthly mortgage payment and the total cash required to purchase the house. The hazard insurance (or home-owner's insurance) was an average for my area. If it isn't appropriate for your area, change the constant in line 290.

If you have a printer and want to list the information instead of displaying it on the screen, change all the PRINT commands to LPRINT in lines 240–370. After printing or copying the totals for reference, you can repeat the first part of the program with different values or you can calculate your income tax rebate. On repeat runs, if the value is unchanged, simply press ENTER.

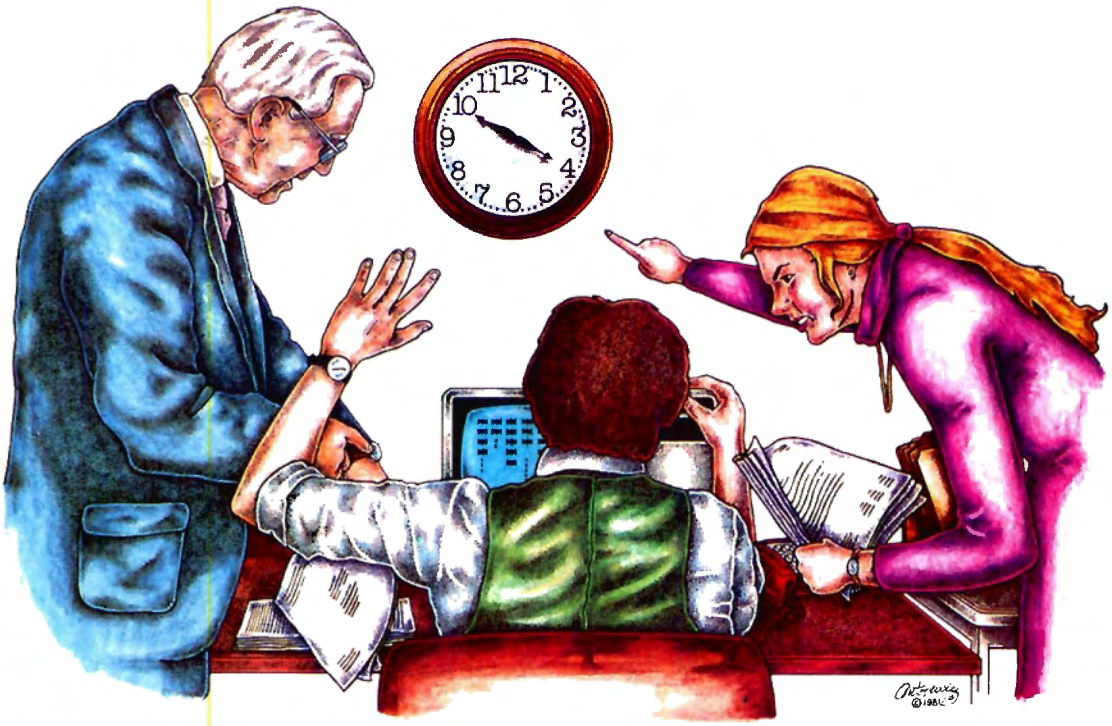
Form 1040

In the last part of the program, the savings you can expect from the IRS are calculated on the average mortgage interest and taxes fo the first five years of home ownership, multiplied by the percent of your income you paid for income tax last year.

Three items are required from your latest IRS Form 1040: adjusted gross income, itemized deductions (if taken) and total income tax paid. The line numbers refer to Form 1040. If you used the short form (1040A), use the values from the corresponding lines, and of course there are no itemized deductions.

Calculating the income tax rebate can only be an estimate because of all the variables. This exercise is particularly revealing to those who are not yet home owners and therefore probably do not itemize their deductions. If you already use the homeowner's deduction, you are of course aware of the savings, but it is interesting to see the effect it has on your monthly mortgage payments. ■

Four New Programs for May To Keep You Busy Till June!



"Deadlines, deadlines! Don't they realize I'm on a collision course with what must be the biggest ASTEROID in the galaxy?! 'Statistics,' they say. That's what we'll all BE if I can't fend off all these SPACE INVADERS! They're clamoring for reports when it's up to me to intercept enemy Gnat fighters, who are determined to strangle our deep space trade routes! With BALL TURRET GUNNER, any moment could be my last! And if that's not enough, I'm surrounded by Quelon Fighter escorts in COSMIC PATROL . . . and the Quelons never miss! Reports! Statistics! Deadlines! I guess some people just have no sense of priority."

Cosmic Patrol

The Cosmic Patrol program puts you in command of a small interstellar patrol craft. Your mission is to defend Terran space and prey on the Quelon supply ships. Beware of the I-Fighter Escorts—they're armed, fast, and manned by intelligent robots linked to battle computers. they NEVER miss! With its fast, real-time action, impressive sound option and superb graphics, COSMIC PATROL is the best program of its kind.

Order No. 0223R tape $14.95 Requires Level II 16K
0224RD disk $19.95 Requires Level II 16K

Asteroid

It's up to you to destroy any asteroids in your sector and to prevent alien spacecraft from infiltrating the Terran Defense Network. Your ship is equipped with an anti-matter cannon. You can shoot large asteroids, but this will shatter them into smaller pieces, each capable of destroying your ship. In addition, alien ships can make instantaneous hyperspace jumps into your area to fire on your ship. You'll need nerves of steel to survive ASTEROID we have no need for non-survivors.

Order No. 0237R tape $14.95 Requires Level II 16K
Order No. 0247RD disk $19.95 Requires Level II 16K

Invaders

The INVADERS are coming! Earth's defenses are all dead except for your laser base. Your assignment is to destroy the approaching invaders. Before Earth's sensors failed, they detected 550 armed invaders in space, speeding toward us in 10 attack formations of 55 each. The sensors detected four different types of attack crafts—large, medium, small and a short profile craft which is the most difficult to destroy If you can't stop the attacks, Earth is doomed!!

Order No. 0240R tape $9.95 Requires Level II 4K

Ball Turret Gunner

Imagine yourself at the control console of the LW 1417 Stratoblazer, ready for the onslaught of enemy GNAT fighter planes. Your Hindsight Director informs you that a GNAT fighter is closing fast. With your finger tensed over the firing key, you'll have only seconds to react Suddenly, you see the fire command. With one beam, the Gnat fighter is reduced to an expanding ball of ionized gas. Mission accomplished. BALL TURRET GUNNER with your choice of multiple level of difficulty, optional sound effects and superb graphics, is more than a game—it's an adventure. Experience it!

Order no. 0051R tape $9.95 Requires Level II 16K.

✓2

Instant Software™

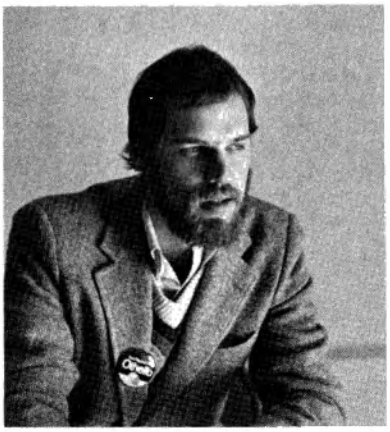
PETERBOROUGH, N.H. 03458

To Order: See your local Instant Software dealer. If these programs are unavailable, order directly.

Call Toll-Free
1-800-258-5473

✓See List of Advertisers on page 306

Success Talks.



"I particularly liked the help I received in rounding out the edges of my program . . . made it more marketable."

Charlie Heath
Author of the top microcomputer version of Othello, which recently beat all other microcomputer opponents in the Santa Cruz Open.

"They (Instant Software) have the largest marketing organization, other than Tandy. This gives my program more exposure . . . and more sales."

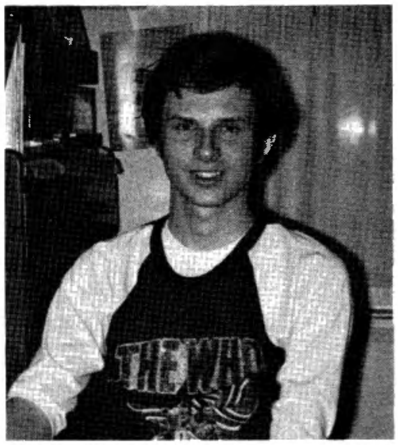
Terry Kepner
Author of ONE "D" MAIL-LIST, and numerous articles in the Industry's top magazines.

"As a European programmer, I can safely say that Instant Software, with its vast marketing and distribution network, has proven to me to be the best organization in the world to submit my TRS-80 utility programs to."

Jake Commander
Author of over 10 top selling utility programs, and many articles on advanced computer programming techniques.

"One thing I've been impressed with is the scale of your Marketing capabilities. So many dealers . . . so many advertisements."

Mike Wall
Author of the best selling ASTEROIDS and COSMIC PATROL . Soon to be a Computer Science major at M.I.T.



These four Instant Software authors have all written programs proven successful in the marketplace. Now it's your turn.

If you're a programmer and are writing what you feel to be an exceptional software program, it's time to think about bringing it to market.

We believe it takes time, talent, training and experience to write good software. It also takes time, talent, training, experience and money to establish an effective marketing system. YOU have the first set of qualifications; WE have the second. Together, we can team up successfully to bring your program to market.

So, what's in this for you? Excellent royalty percentage, paid monthly. 26 sales reps covering the United States with only one objective—to sell your programs. 22 distributors located worldwide with the same sole objective. Over 450 Instant Software dealers in the US. Hundreds of thousands of catalogs, mailed directly to computer users throughout the U.S.A. and the world every three months, to continue your exposure to the buying public. Total support. Any questions?

Success speaks for itself. Let's talk. For a submissions package, write to:

Submissions Department
Instant Software, Inc.
Rt. 101 and Elm Street
Peterborough, N.H. 03458

or call 1-603-924-7296

Instant Software™

PETERBOROUGH
NEW HAMPSHIRE 03458

✓2

An in-depth look at wrap-around mortgages.

How to Buy and Sell Houses

Dale A. Whitman
Professor of Law
University of Washington
18011 60th Avenue, N.E.
Seattle, WA 98155

Financing the sale of your house isn't as easy as it once was. Interest rates on new home mortgages have recently reached high levels which were previously undreamed of, and in too many cases money isn't available from lending institutions at any interest rate. Even if a loan can be obtained, many prospective buyers prefer not to lock themselves into high interest rates, and unless an alternative method of financing the sale is available, they may simply decide to wait.

If your house is on the market and you need to sell quickly, you can be devastated by such buyer reactions.

The alternative which many people have settled on in recent months is financing by the seller himself. If the seller is willing to defer receiving some or all of his equity, and if he will settle for a lower interest rate on the deferred amount than institutional lenders are demanding, he may be able to arrange a much quicker sale.

If there is an existing mortgage loan on the property, it usually has a considerably lower interest rate than current rates, and therefore it is desirable to keep it in effect when the house is transferred. But its outstanding principal balance at the time of the sale is often too low for the buyer to take it on with a cash down payment. For example, suppose the balance on the existing loan is $48,000 and the selling price of the house is $100,000. (The original balance when the existing loan was first made would have been somewhat higher than $48,000, and would have been reduced by the portion of each monthly payment that is applied to amortization. For example, if the original balance were $50,000, the interest rate eight percent and the stated term of the loan 30 years, and if four years and two months had elapsed at the time of the resale of the house, the outstanding balance on the original loan would be just slightly over $48,000.) These figures are a bit higher than average, but not at all unusual.

Unless the seller provides some financing, the buyer will need to come up with the $52,000 difference in cash (which many buyers would find financially impossible) or obtain a second mortgage loan from a bank or finance company (which might be extremely difficult and might involve even higher interest rates than prevail on first mortgages).

Here seller financing can come to the rescue. One method involves the seller "taking back" a second mortgage for part of the price. In the example above, the seller might accept the buyer's promissory note and mortgage for $25,000, leaving only $27,000 in cash to be paid by the buyer at the time of the transfer. In a typical case, the buyer might have ten to twenty years to pay off the second mortgage in monthly installments of principal and interest.

If this method is used, the buyer and seller agree on an interest rate which is to accrue on the unpaid balance of the second mortgage and note. For instance, if they agree on 12 percent interest and a 20-year term, the monthly payments on the second mortgage will be $275.27.

At the same time, the buyer assumes and begins to make the full monthly payments to the lending institution which holds the first mortgage loan originally placed on the property by the seller when he bought the house. If the interest rate on that loan is eight percent and it had an original term of 30 years and an original balance of $50,000, its monthly payments will be $366.88. Thus, the buyer makes two payments each month —one to the bank (or other first mortgage lender) and the other to the seller. Their total will be $642.15.

If this is the approach on which the parties agree, it is easy to calculate the yield (that is, the effective interest rate) to the seller; it is the same as the agreed interest rate on the second mortgage note.

There's another way for the parties to structure the transaction—a way that can give the seller a much higher effective yield on his investment, and yet which may still be very acceptable to the buyer. It's called the wrap-around mortgage. (Californians, who seem to like being different, usually call it the "all-inclusive deed of trust". The meaning is exactly the same as the wrap-around mortgage.)

Here's the way it works. Based on the facts of our example above, the seller agrees to provide all necessary financing, $73,000, himself. He agrees to make this financing available to the buyer for a 25-year term at 12 percent interest. As in the illustration above, the buyer needs to put down only $27,000 in cash.

Under this approach, the original loan to the bank will not be paid off at the time of the sale. Instead, each month the *seller* receives the monthly payment on the $73,000 "loan" from the buyer, and will make the monthly payment to the bank on the original loan. Technically, the $73,000 "loan" is secured by a second mortgage, since the first mortgage to the bank is still in existence.

But from the buyer's point of view, the bank is out of the picture; he is paying on only one loan, and the payments are made to the seller.

The significant advantage of this method from the seller's perspective is that it earns him a much higher effective yield than the stated 12 percent interest. This is because part of the money the seller is "loaning" is in turn being lent to him by the bank at (by current standards) an exceptionally low rate of interest. In our example, only $25,000 of the $73,000 "loan" made by the seller is really being made with the seller's capital. The rest is, in reality, the bank's money, lent in an earlier era at only eight percent. Yet the seller is receiving 12 percent interest on the whole $73,000.

If the fact of the seller's yield being higher than 12 percent is not immediately obvious to you, imagine a situation in which you could borrow money from the bank at eight percent, relend it to another borrower at 12 percent, and use none of your own money at all. Here your effective yield (if we ignore your overhead costs) would be infinite! The seller who makes a wrap-around loan is not in quite this attractive a position, but he comes close if his own capital investment is small compared to the balance on the existing loan.

Enter the Computer

If a seller is considering making a wrap-around loan to finance his sale, he would like to know just what the effective yield will be. Program Listing 1 will answer that question quickly and precisely. It will also allow easy variation in the parameters of the transaction, so that the seller can ask "How much would my yield be affected if the wrap-around loan is for 20 years rather than 25?" and other similar questions. The program also lets the user determine the yield if the wrap-around loan is paid off prior to maturity.

Program Description

Here's an analysis of the program. Lines 100 through 150 input the original data on the existing loan. Line 120 converts the interest rate into a monthly decimal factor with the expression I1 = I0/1200. Similarly, line 130 converts the term of the loan into months by the expression N1 = N0*12. Line 150 computes a composite date by multiplying the year in which the loan was made by 12, and adding the month.

Lines 160 and 170 plug the relevant values into the subroutine at line 1000, which is a standard formula for computing the monthly payments on a fully amortizing loan. This monthly payment amount is then assigned the variable M1. Lines 200 to 260 implement a similar procedure by inputting relevant values for the wrap-around loan. Monthly payments are then computed in line 270 and the subroutine at line 1000.

Line 270 also computes R1, the number of months remaining on the original bank loan. Line 280, along with the subroutine at line 2000, then computes the future value of the original loan amount as of the time the wrap loan is made. Line 290 (together with the subroutine at line 3000) computes the future value of the payments which have been made on the original loan up to the time the wrap loan is made. The difference between these two values is, by definition, the outstanding balance on the original loan as of the date of the wrap loan closing. This is a standard method for calculating the outstanding balance on a mortgage loan as of a given date. That amount is assigned to variable B1 in program line 290.

The program section running from line 400 to line 465 displays all of the data which has been input on both of the loans, as well as the monthly payments on both. If you have made an error in inputting some data item, you can see the mistake on the screen. Hit BREAK, and GOTO 100. This will correct the mistaken item without disturbing the other data.

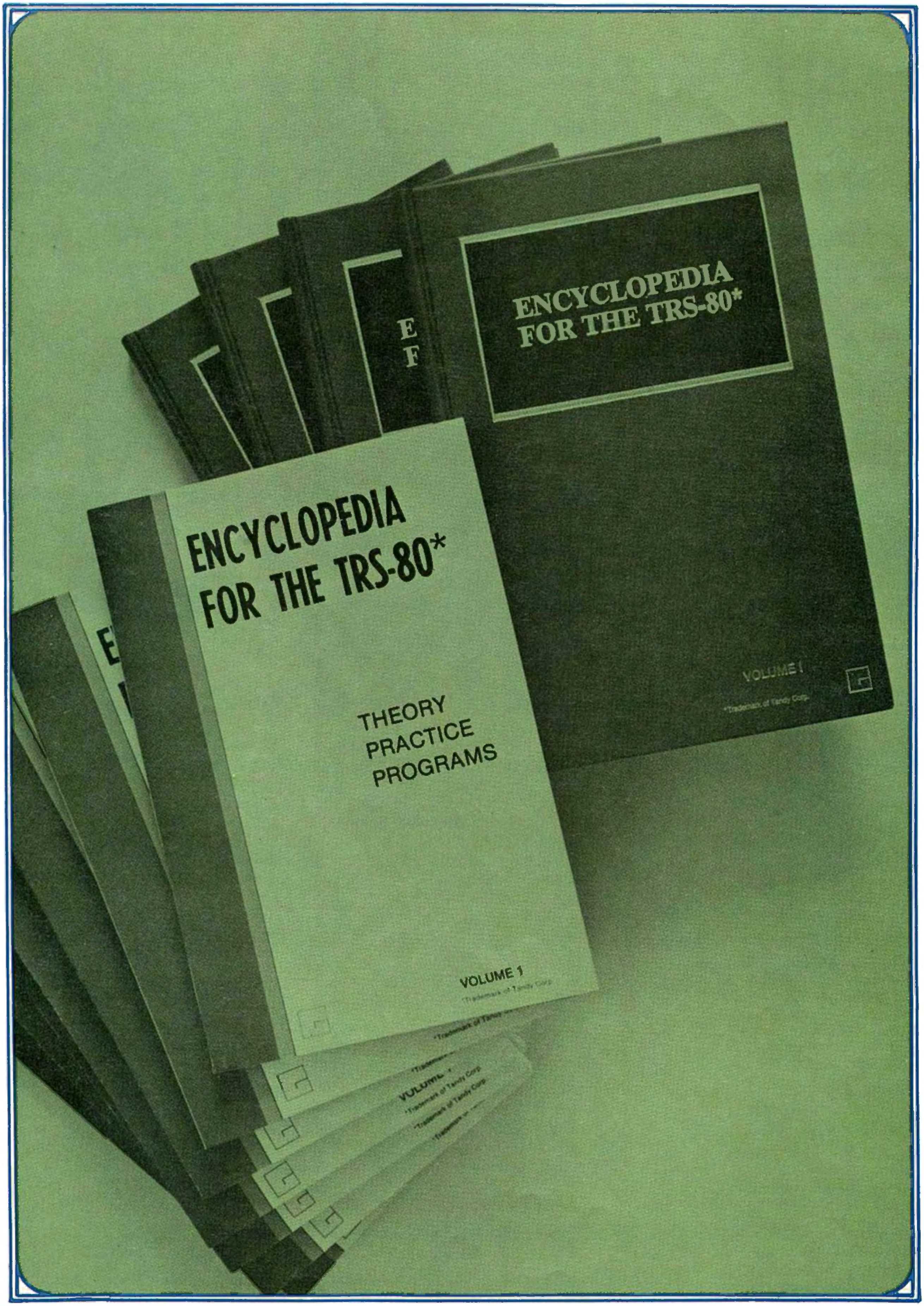
Line 470 displays the previously calculated balance on the original loan at the time the wrap loan is made. Lines 502 through 509 calculate and display the wrap lender's investment in the wrap loan—that is, the amount of the wrap loan minus the balance owing on the original loan. If there were no wrap loan, but instead a sale (by the seller) for cash, he would receive this amount after paying off the original loan. Hence, it represents his capital investment in the transaction.

Incidentally, the program refers to the "wrap lender" here, but that person, in our illustration, is simply the seller of the property. Sometimes institutional investors, especially life insurance companies, make wrap-around loans on properties which are not being sold by the lender. The program uses "wrap lender" for the sake of generality.

The section starting with line 510 deals with the possibility that the wrap loan will be paid off prior to its maturity in a lump sum. Many house sellers put clauses in the documents requiring this; they may put the wrap loan on a schedule of payments which would require 20 years to pay off, but require a cash payoff after, say, three years. This will give the buyer time to refinance the house with an institutional lender, perhaps at a lower interest rate (if rates fall, which the parties hope, but which is by no means certain.) Even if prepayment is not obligatory, the buyer may desire to prepay the loan anyway because lower rates have in fact become available.

If the user indicates at lines 510 and 512 that a prepayment will occur, line 515 then computes the balance which the wrap loan will have on the payoff date, and lines 540 through 560 do the same with the balance on the original loan as of the wrap loan's assumed payoff date. Both of these

✓See List of Advertisers on page 306


ENCYCLOPEDIA
FOR THE TRS-80*
VOLUME I
ENCYCLOPEDIA
FOR THE TRS-80*
THEORY
PRACTICE
PROGRAMS
VOLUME 1

The Newest Peripheral for your Microcomputer

The Encyclopedia for the TRS-80*

What's the key to getting the most from your TRS-80? No, it isn't disk drives or printers or joysticks. It's **information.** Without a continual supply of information and ideas, you can't realize the full potential of the TRS-80.

Now, you'd think that the 150 pages of articles published in **80 Microcomputing** each month would satisfy even the most voracious of TRS-80 users. But no! You've asked for even more information!

Our response to the clamor for additional information is the **Encyclopedia for the TRS-80,** a ten-volume reference work that is absolutely packed with programs and articles carefully selected to help you make the most of your microcomputer. You can consider the volumes of the **Encyclopedia** to be an extension of the documentation that came with your TRS-80. The articles and programs will be similar to those found in **80 Microcomputing.** Each book is full of material on programming techniques, business, games, tutorials, education, utilities, interfacing . . . you name it.

Unlike conventional encyclopedias, the **Encyclopedia for the TRS-80** will never become stale or out of date. That's because the volumes of the **Encyclopedia** are being issued one-at-a-time, over a period of months. This means that each new volume will reflect the latest developments and discoveries, making this a living encyclopedia for TRS-80 users.

Volume 1 will be out in June. It's more than 270 pages in length . . . that's 270 pages of solid information . . . no advertising. Here's a sample of what's included in Volume 1:

Four Graphics Methods—Improved techniques for dealing with the TRS-80's low resolution graphics.
TTY Interface—Build an interface hookup a Teletype to a TRS-80
The President Decides—A thrilling simulation that lets you make decisions as if you were President of the United States
TRS-80: Calling FORTRAN from BASIC—Call Microsoft FORTRAN subroutines from BASIC on the TRS-80
Measure Instructional Effectiveness—Many programs teach or tutor; this one helps determine the effectiveness of the lessons.
The Invoice—A program for small businesses which produces custom invoices.
Punch Out Your Disks—Double your disk storage in seven easy steps
LOAD and Go—Adds a new and useful command to Level II BASIC.

The **Encyclopedia for the TRS-80** is available in two editions. The deluxe COLLECTOR'S EDITION is a handsome green and black hardcover volume, with the title stamped in gold and a colorful protective dust jacket provided. These durable volumes are perfect for school and library use and make a handsome addition to your personal library. They are available for $19.95 per volume. Also available is the more economical softcover edition. This contains all the information in the hardcover edition, but at a savings of over 45%. It is ruggedly bound for heavy use and has an attractive green, ivory, and gold cover. Available for $10.95 per volume. Both editions are in an easy-to-read 6" × 9" format which is perfect bound to remain flat when opened.

80 MICROCOMPUTING SPECIAL INTRODUCTORY OFFER

Order the entire set (the first ten volumes) of the **Encyclopedia for the TRS-80** and receive the tenth volume FREE!
PAY ONLY **$164.00** for the first ten volumes of the deluxe hardcover COLLECTOR'S EDITION—a regular **$199.50** value
PAY ONLY **$83.00** for the first ten volumes of the softcover edition—a regular **$109.50** value
PLUS—This special offer includes UPS shipping and handling charges (an additional savings of **$15.00**)
Beat inflation by buying the whole set now . . . we cannot guarantee these same single volume prices for future volumes.
You cannot lose! Preview the first volume of the set and judge for yourself. If the **Encyclopedia for the TRS-80** does not meet your needs or expectations, just return the first volume within ten days and we will refund the entire amount of your purchase. If at any time you are dissatisfied, you may cancel the remaining volumes and receive a refund for the full value of those volumes. The books will be automatically shipped to you every 6 weeks via UPS. Take advantage of this offer now! You can't afford to miss out on the newest peripheral—the **Encyclopedia for the TRS-80.** Call your order in today TOLL FREE at 1-800-258-5473, or use the coupon below or the postage-paid postcard included in this magazine.

THE Encyclopedia for the TRS-80*

YES, please reserve for me a complete set of the first ten volumes of the **Encyclopedia for the TRS-80.**

___ I want to go first class so send me the deluxe hardbound COLLECTOR'S EDITION—a $199.50 value for **$164.00 postpaid**. A total savings of $34.95 over the single volume price! (Order #EN8100)

___ Send me the "blue collar" softcover edition—the same information at a substantially lower price, a $109.50 value for **$83.00 postpaid**! A total savings of $25.95 over the single volume price! (Order #EN8080)

YES, I want to start my Encyclopedia collection due out in June.

___ Volume 1 of the deluxe hardcover COLLECTOR'S EDITON @$19.95 (Order #EN8101)**

___ Volume 1 of the Softcover edition @ $10.95 (Order #EN8081)**

***(Please enclose $1.50 per volume for shipping and handling. All volumes will be shipped UPS if the complete street address is provided; otherwise shipment is by 4th class book rate.) Allow 8-10 weeks for delivery.*

___Payment enclosed   Please charge to my ___VISA ___Mastercard ___AmEx

Card#___________ MC Interbank#________ Expires________

Name______________________

Address______________________

City____________ State______ Zip________

Signature______________________

80 Microcomputing, Pine St., Peterborough, NH 03458

*TRS-80 is a trademark of Tandy Corp.   8005

". . . every time you RUN the program while debugging it, your data will be in place automatically. . . ."

routines use the same procedure discussed above in connection with lines 280 and 290. Incidentally, the original loan *must* be paid off by the seller at the same time the wrap loan is paid off by the buyer (if it has not already been fully paid by the monthly payments); otherwise the buyer would have fully paid for the property, but would still not have cleared his title of the original mortgage.

Using this information on the balances due on the two loans at the time of payoff (TW and TE, respectively), lines 580 through 590 display the financial consequences of the payoff: how much money is received by the wrap lender, how much of that must be paid over to the original lender, and how much the wrap lender retains.

Computing the Yield

Now we come to the part of the program for which all the rest is merely prologue—the computation of the yield (i.e., the effective interest rate to the wrap lender) on the wrap loan. Two separate routines deal with this matter, depending on whether an early payoff is expected or not. Variable Y is the flag which indicates whether early payoff will occur. In line 620 it sends execution to line 650 thorugh 660 if there is no early payoff, while if there is, lines 625 through 645 control instead.

The yield computation itself is a recursive process, and works like this in principle: Some interest rates are simply assumed to be the correct yield. (In this program, the initial trial assumption is 18 percent per annum, or .015 per month, set in line 610.) Then the present value (as of the date the wrap loan is made) of all future cash flows to the wrap lender is computed, using the assumed interest rate as a discount factor. The actual formulas which do this discounting are in lines 4000 and 5000. The discounting present values are then summed against one another; some of them are positive (e.g., the monthly payments received by the wrap lender) and some are negative (e.g., the monthly payments made by the wrap lender to the original lender). Their sum is variable SM in line 640 or 660.

This present value is then netted against the wrap lender's original investment, previously derived as P2-B1. If the net present value is exactly equal to P2-B1, so that the sum is zero, we have happened to pick exactly the correct yield or discount rate—18 percent! Of course, it would be blind luck if this happened, and normally the sum (NV in line 670) is either positive or negative, and not zero. Lines 680 and 690 use this information (the sign of NV, represented by BASIC keyword SGN) to adjust the assumed discount rate and execution is sent back to line 625 or 650, the beginning of the FOR-NEXT loop which calculates the present values, for a new trial.

Lines 680 and 690 do what is termed a binary search. For example, if the assumed discount rate of 18 percent is too high, it will produce a net present value, SM in line 670, which is lower than the wrap lender's original investment (P2-B1). NV will then have a negative sign, and line 680 adjusts the assumed interest rate IT downward by one-half of the difference between IT's previous value and its value in the trial before that. Thus, in successive trials, IT might assume values of 18 percent, nine percent, 13.5 percent, etc. as it closes in on the correct value. When the correct value is finally reached, so that SGN of the difference between the last two interest rate trials is zero, control passes to line 700 for a print of the final value.

The approach of successive trials is the only feasible way of computing the effective yield produced by a series of unequal future cash flows, and it takes time. The loop used in this program has 99 steps, but in fact I have never had a computation require more than 25, and they are often shorter. About ten seconds or so are required for 25 steps.

Lines 695 and 696 display the trial number, the net value, and the assumed interest rate for each trial, so you can see the progress the computer is making, and so you will know that the program has not locked up or bombed.

One might worry about precision in this program, but such concerns are really irrelevant. The six digits of single-precision arithmetic are more than adequate for purposes of comparing yields on alternative investments with the wrap-around loan. I tried changing to double-precision, but succeeded only in slowing down execution very noticably without producing more useful results.

Programming Tips

I like to write programs which deal with financial and investment decisions, and I'd like to pass on a few ideas I've discovered which make writing and running such programs much easier and more enjoyable.

The first idea stems from the fact that any editing of a program line wipes out all stored variable values. In a program which is being debugged, and which requires a large number of variables to be defined (as by INPUT statements), this is a pain in the neck. You can easily spend half of your program writing time inputting the same old data again and again.

To solve this problem, use a DATA statement like that in line 10000, along with a READ statement like line 95. Thus, every time you RUN the program while debugging it, your data will be in place automatically and without delay or pain. The READ statement should be placed before the point in the program to which return is made after a full run of the program—line 100 in this case.

The DATA and READ statements have another advantage as well, and I always leave them in even after the program is fully debugged for this reason. They make demonstrations of the program very easy, since if spectators have no particular data they wish to see run, your pre-written DATA statement will show them very quickly how the program operates.

A second idea which is very helpful in programs of this kind grows out of Level II BASIC's characteristic of simply leaving the former value of a variable in place if an INPUT statement is encountered and the user merely presses ENTER without any numbers. This means that you can write INPUT statements which display the former value, then ask the user to input a new value. If the old value is satisfactory, the user can simply press ENTER without taking the trouble to insert the same value over again. This program uses this feature in lines 100 to 150 and 200 through 250. When combined with the READ and DATA statements mentioned above, they permit an extremely quick demonstration of the program; the user merely types RUN and then presses ENTER to show, in sequence, each item of data on each of the two loans. The pre-existing values are shown in brackets, and can readily be changed by typing new values if desired.

Using this technique requires a little care in writing the program, as it is important that further manipulations on a variable's value not be done once it is input. For example, in the wrap-around mortgage program (and many other financial programs) it is customary to input interest rates on an annual basis (e.g., 12 percent). But the rate must be converted to a monthly decimal for computation purposes. One often sees programs which redefine the original variable

". . . the use of the STRING$ command is an excellent way to divide up the screen. . . ."

to accomplish this. For example:

120 INPUT "ANNUAL INTEREST RATE";I1:I1 = I1/1200

There's nothing wrong with this method if the program will be run only once. But if you plan to repeat it, using most of the same input data again without a new RUN (as by the GOTO 100 in line 730), you discover that in the second run I1 is all wrong. Thus, if the value originally input for I1 in the first run was 12, line 120 above will convert it to .01. The next time line 120 is executed, if no new I1 is input, the program will divide the .01 value by 1200, giving you an I1 of .000083, which is obviously wrong.

Fortunately, this problem is easy to avoid. Simply use a new variable for the manipulation of the input data, rather than redefining the old variable. In line 120 of the wrap-around mortgage program, I0 is the value to be input; I1 is then defined as I0/1200, leaving I0 at its original value. It will still be there, with a value of 12, the next time line 120 is executed. The same principle may be seen in line 130, which manipulates the term of the loan, converting it from years to months.

This program is written in 32-character-per-line format because it was originally prepared for classroom demonstration purposes in my private land development course. I use 19-inch Sony TV monitors with direct video input; they are designed for use in a closed-circuit TV system, and with a suitable adapter (containing merely a single feedthrough capacitor), perform very nicely when driven by the TRS-80. However, their resolution is not as good as the TRS-80's 12-inch monitor, and the bigger print is therefore quite helpful, especially for students who may be 20 feet or more from the monitor. In general, the 32-character format is often preferable even for home use, especially when only a small amount of information needs to be shown on the screen at one time.

Finally, the use of the STRING$ command is an excellent way to divide up the screen into several blocks showing different sets of information. STRING$ is rapid in execution and easy to program; I think it is one of the unsung heros of Microsoft's Level II. You'll see it in lines 105, 205, 405, 415, etc., in this program.

Legal Issues

While a wrap-around mortgage is often an attractive way to sell a house, it can raise some tricky legal problems. There isn't space in this article to discuss them in any real depth, but let me mention two of the more important ones.

Many mortgage loans made by institutional lenders in the past few years contain "due-on-sale" clauses. Such a clause says, in substance, that if you sell your house without the bank's permission, it has the right to insist that the loan be paid off immediately. Often the bank will give the necessary permission, but only on condition that (1) the new owner has satisfactory

```
1Ø '*** WRAPAROUND MORTGAGE YIELD ***
2Ø '                DALE A. WHITMAN
3Ø '18Ø11 6ØTH AVE. N.E.
4Ø 'SEATTLE, WA. 98155
45 DIMIT(99)
5Ø CLS:PRINTCHR$(23):PRINT"THIS PROGRAM COMPUTES THE":PRINT"EFFECT
     IVE YIELD ON A":PRINT"WRAP-AROUND MORTGAGE.
6Ø FORN=1TO1ØØØ:NEXTN
8Ø READP1,IØ,NØ,Y1,X1,P2,I9,N9,Y2,X2
9Ø DEFSTRF:F="###,###"
1ØØ CLS:PRINTCHR$(23):PRINT"ENTER DATA ON EXISTING LOAN:"
1Ø5 PRINTSTRING$(31,61)
11Ø PRINT"ORIGINAL PRINCIPAL AMOUNT:":PRINT"  <";USINGF;P1;:PRINT"
     >";TAB(2Ø)"";:INPUTP1
12Ø PRINT"ANNUAL INTEREST RATE:":PRINT"  <";IØ;"% >";TAB(2Ø)"";:IN
     PUTIØ:I1=IØ/12ØØ
13Ø PRINT"ORIGINAL LOAN TERM (YEARS):":PRINT"  <";NØ;" YRS >";TAB(
     2Ø)"";:INPUTNØ:N1=NØ*12
14Ø PRINT"DATE LOAN ORIGINALLY MADE:":PRINT"  YEAR: <";Y1;">";TAB(
     2Ø)"";:INPUTY1
15Ø PRINT"  MONTH: <";X1;">";TAB(2Ø)"";:INPUTX1:D1=12*Y1+X1
16Ø P=P1:I=I1:N=N1:GOSUB1ØØØ
17Ø M1=M
2ØØ CLS:PRINTCHR$(23):PRINT"ENTER DATA ON NEW WRAP LOAN:"
2Ø5 PRINTSTRING$(31,61)
21Ø PRINT"INITIAL PRINCIPAL AMOUNT:":PRINT"  <";USINGF;P2;:PRINT">
     ";TAB(2Ø)"";:INPUTP2
22Ø PRINT"ANNUAL INTEREST RATE:":PRINT"  <";I9;"% >";TAB(2Ø)"";:IN
     PUTI9:I2=I9/12ØØ
23Ø PRINT"INITIAL LOAN TERM (YEARS):":PRINT"  <";N9;" YRS >";TAB(2
     Ø)"";:INPUTN9:N2=N9*12
24Ø PRINT"CLOSING DATE FOR WRAP LOAN:":PRINT"  YEAR: <";Y2;">";TAB
     (2Ø)"";:INPUTY2
25Ø PRINT"  MONTH: <";X2;">";TAB(2Ø)"";:INPUTX2
26Ø D2=12*Y2+X2
27Ø I=I2:N=N2:P=P2:GOSUB1ØØØ:M2=M:R1=N1-D2+D1
28Ø P=P1:I=I1:N=N1-R1:GOSUB2ØØØ:TA=T
29Ø P=M1:I=I1:N=N1-R1:GOSUB3ØØØ:TB=T:B1=TA-TB
4ØØ CLS:PRINTCHR$(23);:PRINT"SUMMARY OF DATA INPUT:"
4Ø5 PRINTSTRING$(31,61)
41Ø PRINTTAB(1Ø)"ORIG.LOAN";TAB(22)"WRAP LOAN"
415 PRINTSTRING$(31,61)
42Ø PRINT"ORIG.AMT.";TAB(12)USINGF;P1;:PRINTTAB(24)USINGF;P2
43Ø PRINT"INT. RATE";TAB(14)IØ;"%";TAB(26)I9;"%"
44Ø PRINT"ORIG.TERM";TAB(16)NØ;TAB(28)N9;
45Ø PRINT"MO. PMT";TAB(12)USINGF;M1;:PRINTTAB(24)USINGF;M2
46Ø PRINT"CLOSING";TAB(1Ø)X1;"/";Y1;TAB(22)X2;"/";Y2
465 PRINTSTRING$(31,61)
47Ø PRINT"WHEN WRAP LOAN IS CLOSED, ORIG.":PRINT"LOAN BALANCE WILL
     BE $";:PRINTUSINGF;B1:PRINT"WITH";R1;"MONTHS REMAINING."
475 PRINTSTRING$(31,61)
48Ø INPUT"PRESS ENTER TO CONTINUE";Z2
485 CLS:PRINTCHR$(23):PRINT"HIT 1 TO RE-ENTER ORIGINAL LOAN":PRINT
     :PRINT"HIT 2 TO RE-ENTER WRAP LOAN":PRINT:PRINT"HIT ANY OTHER
     KEY TO CONTINUE."
49Ø X$=INKEY$:IFX$=""GOTO49Ø
492 X=VAL(X$):ONXGOTO1ØØ,2ØØ
495 PRINT:PRINTSTRING$(31,61):PRINT
496 Y=Ø:PRINT"HIT 1 IF WRAP LOAN WILL BE":PRINT"PAID OFF PRIOR TO
```

Program continues

"You should discuss the wrap-around loan with the original lender before proceeding. . . ."

credit and income, and (2) the interest rate on the loan will be substantially increased.

From the viewpoint of the buyer and seller, the income and credit standards are usually no problem, but an increase in the interest rate may make the whole wrap-around mortgage idea far less atttractive, since it can drastically reduce the seller's yield on the wrap loan. Some states don't permit lenders to insist on an interest rate increase, but others do. What's more, federal savings and loan associations can demand interest rate increases in all states, since they are governed by federal law.

You should discuss the wrap-around loan with the original lender before proceeding with the transaction. If the lender objects or demands that the interest rate be increased, you will probably need to consult a reputable lawyer with real estate experience to determine whether the lender is on sound legal ground. By all means, stand up for your rights; lenders will often back down when confronted. But don't try to go ahead with the wrap-around transaction without telling the lender. The chances are very good that your buyer will be found out before long, and he may be forced to pay off the loan or switch to a higher interest rate under very inconvenient circumstances. In fact, if you are the seller under such conditions, you might end up having to pay a higher interest rate on the original loan than the buyer is paying you on the wrap loan!

A second caution—this time to buyers. You need to be sure that, when you pay your monthly installment to the seller, he in turn will pay the bank. It's a good idea to run your payments through a neutral third party, such as a lawyer or an escrow company, with instructions to split your payments into two parts, one to the seller and one to the bank. This way, the seller won't be tempted to keep all of your money and let the bank loan go into default.

There are plenty of other legal considerations which go into the structuring of a wrap-around loan, so you should work with an experienced real estate lawyer. Real estate brokers and sales people often try to set up wrap-around deals without a lawyer's involvement, but too frequently they don't understand all of the ramifications of their own actions; indeed, the same is true of many lawyers who do not specialize in real estate work. Don't be afraid to spend a couple of hundred dollars on good advice and document preparation; it could save you thousands in higher interest or legal fees in the long run. ■

```
     MATURITY.":PRINT:PRINT"HIT ANY OTHER KEY TO CONTINUE."
497 X$=INKEY$:IFX$=""GOTO497
498 Y=VAL(X$):IFY=1THENGOTO499ELSEGOTO5Ø2
499 PRINT:INPUT"ENTER NUMBER OF YEARS TO PAYOFF";N8:N3=N8*12
5Ø2 CLS:PRINTCHR$(23);:PRINT"AT TIME OF WRAP LOAN CLOSING:":PRINT"
475 PRINTSTRING$(31,61)
48Ø INPUT"PRESS ENTER TO CONTINUE";Z2
485 CLS:PRINTCHR$(23):PRINT"HIT 1 TO RE-ENTER ORIGINAL LOAN":PRINT
     :PRINT"HIT 2 TO RE-ENTER WRAP LOAN":PRINT:PRINT"HIT ANY OTHER
     KEY TO CONTINUE."
49Ø X$=INKEY$:IFX$=""GOTO49Ø
492 X=VAL(X$):ONXGOTO1ØØ,2ØØ
495 PRINT:PRINTSTRING$(31,61):PRINT
496 Y=Ø:PRINT"HIT 1 IF WRAP LOAN WILL BE":PRINT"PAID OFF PRIOR TO
     MATURITY.":PRINT:PRINT"HIT ANY OTHER KEY TO CONTINUE."
497 X$=INKEY$:IFX$=""GOTO497
498 Y=VAL(X$):IFY=1THENGOTO499ELSEGOTO5Ø2
499 PRINT:INPUT"ENTER NUMBER OF YEARS TO PAYOFF";N8:N3=N8*12
5Ø2 CLS:PRINTCHR$(23);:PRINT"AT TIME OF WRAP LOAN CLOSING:":PRINT"
     INVESTMENT OF WRAP LENDER (WRAP":PRINT"LOAN MINUS BALANCE ON
     ORIGINAL"
5Ø5 PRINT"LOAN = $";:PRINTUSINGF;P2-B1
5Ø8 PRINTSTRING$(31,61)
5Ø9 IFY=ØPRINT:INPUT"PRESS ENTER TO CONTINUE";Z2:GOTO6ØØ
51Ø PRINT"AT TIME OF PAYOFF (";N8;"YEARS":PRINT"FROM WRAP LOAN CLO
     SING ):"
52Ø P=P2:N=N3:I=I2:GOSUB2ØØØ:TA=T
525 P=M2:N=N3:I=I2:GOSUB3ØØØ:TB=T
53Ø TW=TA-TB 'TW IS WRAP BAL AT PAYOFF
54Ø P=P1:N=N1-R1+N3:I=I1:GOSUB2ØØØ:TA=T
55Ø P=M1:N=N1-R1+N3:I=I1:GOSUB3ØØØ:TB=T
56Ø TE=TA-TB 'TE IS ORIG LOAN BAL AT PAYOFF
57Ø PRINT:PRINT"WRAP LOAN BALANCE:";TAB(2Ø)USINGF;TW
58Ø IFTE<ØPRINT:PRINT"ALL RETAINED BY WRAP LENDER.":PRINT:PRINT"OR
     IGINAL LOAN WILL BE FULLY PAID":PRINT:INPUT"PRESS ENTER TO CO
     NTINUE";Z2:GOTO6ØØ
585 PRINT:PRINT"ORIG.LOAN BALANCE:";TAB(2Ø)USINGF;TE
59Ø PRINT:PRINT"NET CASH TO";TAB(2Ø)"-------":PRINT"WRAP LENDER:";
     TAB(2Ø)USINGF;TW-TE:PRINT:INPUT"PRESS ENTER TO CONTINUE";Z2:G
     OTO6ØØ
6ØØ CLS:PRINTCHR$(23):PRINT"NOW COMPUTING YIELD."
61Ø IT(Ø)=Ø:IT(1)=.Ø15
62Ø IFY=ØGOTO65Ø
625 FORJ=1TO99:I=IT(J):P=M2:N=N3:GOSUB4ØØØ:SM=T
628 IFTE>ØP=TW-TEELSEP=TW
63Ø N=N3:I=IT(J):GOSUB5ØØØ:SM=SM+T
635 P=M1:N=N3:IFN1-D2+D1<N3THENN=N1-D2+D1
64Ø GOSUB4ØØØ:SM=SM-T
645 GOTO67Ø
65Ø FORJ=1TO99:I=IT(J):P=M2:N=N2:GOSUB4ØØØ:SM=T
66Ø I=IT(J):P=M1:N=N1-D2+D1:GOSUB4ØØØ:SM=SM-T
67Ø NV=SM-P2+B1
68Ø IT(J+1)=IT(J)+SGN(NV)*(1/2)*ABS(IT(J)-IT(J-1))
69Ø IFSGN(IT(J)-IT(J-1))=ØGOTO7ØØ
695 PRINT"TRIAL";J;" NET VAL DIFF.=";NV
696 PRINT"INTEREST =";12ØØ*IT(J);"%":PRINT
697 NEXTJ
7ØØ PRINT"FINAL YIELD =";12ØØ*IT(J);"%"
71Ø PRINT:PRINT"PRESS ENTER FOR NEW RUN."
72Ø INPUTZ3
73Ø GOTO1ØØ
999 END
1ØØØ V=1/(1+I)[N:M=P*I/(1-V):RETURN:'MONTHLY PAYMENT
2ØØØ T=P*(1+I)[N:RETURN:'FUTURE VAL OF LUMP SUM
3ØØØ T=P*(((1+I)[N)-1)/I:RETURN:'FUTURE VAL OF PERIODIC PMTS
4ØØØ T=P*(1-(1/(1+I))[N)/I:RETURN 'PRES. VAL OF STREAM OF FUTURE P
      MTS
5ØØØ T=P/(1+I)[N:RETURN 'PRESENT VAL OF SINGLE FUTURE PMT
1ØØØØ DATA4ØØØØ,8,3Ø,1975,1,8ØØØØ,12,25,198Ø,1
```

Program Listing 1. Wrap-around Loan Program

✓See List of Advertisers on page 306

Evaluate the financing plan for your next new-used or used-new car with this program.

A Dollar Down...

Jack Martin
773 Radar Squadron
Montauk, NY 11954

When spring rolls around, you'll be looking at the rust spots, faded paint and scratches on the old family car, and getting the itch to trade her in for a shiny new one.

You'll visit the car lot, kick a few tires, and when you have that new car with all the options picked out, the sales representative will say, 'We'll take your old car and $5000. You can drive it home today for $175.78 per month, for the next 36 months. Can I write up the order?"

You'll study the figures, look at the bright paint job, look again at the figures, then you'll say, "Yeh, we'll take it!"

What you should do is take the figures home first, and run them through this program. It will provide answers to cost and interest quickly and without the sales-pitch pressure.

The Program

Although written for the Radio Shack Level II TRS-80, the program should readily adapt to other BASICs. There are no unique statements used, except INKEY$ to reduce response time and save wear on the ENTER key.

Lines 60 through 130 graphically introduce the program. Lines 150 to 190 ask the user to input information concerning the cost of the car, monthly payments, and the term of the loan. The program will handle periods of 18, 24, 30, 36, 42, 48, 54, or 60 months. These are common financing terms and should fit most circumstances.

Lines 350 through 400 and the data statements store specific values in the S(N,R) array. The value stored represents the Present Value of $1 Received Monthly or PVIF. It is based on the formula:

$$A_{n,r} = \$1 \left[\frac{1-(1-r)^{-n}}{r} \right] = \$1\, P_{n,r}$$

The value is used to determine the interest rate and becomes a necessary ingredient in other calculations performed by the program.

After the math functions are completed, the program displays information concerning the car purchase (lines 520 to 630). Then the user may alter the interest rate or the period of repayment to see what difference the change makes. The new values are compared to the original deal. Sometimes these comparisons can be real eye-openers! If a comparison is not requested, the user may run a complete new cost analysis, or exit the program.

The percentage rate change and comparison is accomplished in lines 840 to 1340. The program will handle percentage inputs between nine and 24 percent. The time period is done in lines 1360 through 1830.

Buying the Car

Getting back to our car buying example, you would actually pay 16 percent interest on the loan, and the $5000 car would cost you $6328.08.

Could you get the loan cheaper, for less interest? Probably, but it would require shopping around for the loan just as carefully as you did for the car.

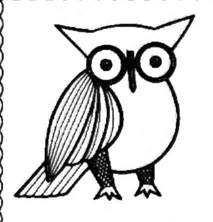
Let's say the local bank or credit union provided the $5000 loan at 12 percent interest over the same 36-month payment period. Using the comparison feature of the program, you discover you would only have to pay back $5978.48, a savings of $349.60. Look closer at the two comparisons. At 16 percent, your payments were $175.78, while at 12 percent interest they are $166.06. That leaves you $9.71 each month to spend.

Cars cost quite a bit these days, but don't pay more than you have to. I hope this program will help prevent that. ■

```
10 REM     *** CAR DEAL ***
20 REM     JACK MARTIN      MAY 1980
30 CLEAR500
40 A$="$$##,####.##"
50 DIM S(8,16)
60 CLS
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINTSTRING$(55,"$");
90 PRINT
100 PRINT@450,"             C A R   D E A L"
110 PRINT
120 PRINTSTRING$(55,"$");
130 FORX=1TO600:NEXTX
140 REM   INPUT DATA
150 CLS:INPUT"PRICE OF CAR --- $ ";C
160 PRINT
170 INPUT"AMOUNT OF MONTHLY PAYMENT --- $ ";M1
180 PRINT
190 INPUT"HOW MANY MONTHS TO PAY LOAN --- ";N1
200 IFN1=18THENB=1:GOTO320
210 IFN1=24THENB=2:GOTO320
```

```
220 IFN1=3ØTHENB=3:GOTO32Ø
230 IFN1=36THENB=4:GOTO32Ø
240 IFN1=42THENB=5:GOTO32Ø
250 IFN1=48THENB=6:GOTO32Ø
260 IFN1=54THENB=7:GOTO32Ø
270 IFN1=6ØTHENB=8:GOTO32Ø
280 IFN1<>18ORN1<>24ORN1<>3ØORN1<>36ORN1<>42ORN1<>48ORN1<>54ORN1<>
    6ØGOTO29Ø
290 PRINT"SORRY - I CAN ONLY HANDLE FINANCING FOR 18, 24, 3Ø, 36,"
300 PRINT"42, 48, 54 OR 6Ø MONTHS."
310 GOTO19Ø
320 PRINT@916,"COMPUTING...."
330 LETX1=C/M1
340 REM    SET UP ARRAY
350 RESTORE
360 FOR N=1TO8
370 FORR=1TO16
380 READS(N,R)
390 NEXTR
400 NEXT N
410 REM    FIND 'X' VALUE
420 FORR=16TOØSTEP-1
430 IFR=ØTHENPRINT:PRINT"I DO NOT FINANCE BELOW 9%":GOTO71Ø
440 IFX1=S(B,R)THEN49Ø
450 IFX1>S(B,R)THEN48Ø
460 IFR=16ANDX1<S(B,R)THENPRINT:PRINT"THAT LOAN IS ABOVE 24%, MY L
    IMIT":GOTO71Ø
470 LETP1=R+9:GOTO51Ø
480 NEXTR
490 LETP1=R+8
500 REM    DISPLAY DATA
510 T1=M1*N1:I1=T1-C
520 CLS:PRINT"   C O S T    A N A L Y S I S"
530 PRINTSTRING$(55,"=");
540 PRINT:PRINT"COST OF CAR:";"";:PRINTTAB(45)USINGA$;C
550 PRINT"MONTHLY PAYMENT:";"";:PRINTTAB(45)USINGA$;M1
560 PRINT"PERIOD OF LOAN (MONTHS):";"";:PRINTTAB(48);N1
570 PRINT"INTEREST RATE OF LOAN (%):";"";:PRINTTAB(48);P1
580 PRINT"TOTAL COST OF LOAN:";"";:PRINTTAB(45)USINGA$;T1
590 PRINT
600 PRINT"AS YOU CAN SEE, IT IS A";P1;"PERCENT LOAN."
610 PRINT"THAT $";C;"CAR WILL ACTUALLY COST YOU $";T1
620 PRINT"OVER THE";N1;"MONTH PERIOD. YOU WILL PAY"
630 PRINT"$";I1;"IN INTEREST."
640 PRINT
650 PRINT"A DIFFERENT INTEREST RATE OR PERIOD OF LOAN CAN"
660 PRINT"MAKE A BIG DIFFERENCE. WANT A COMPARISON (Y/N)?"
670 GOSUB197Ø
680 IFKB=89GOTO77Ø
690 IFKB<>78GOTO67Ø
700 IFKB=78GOTO71Ø
710 PRINT:PRINT"WANT ANOTHER COMPLETE COST RUN?"
720 GOSUB197Ø
730 IFKB=89GOTO3Ø
740 IFKB<>78GOTO72Ø
750 IFKB=78GOTO185Ø
760 REM    GET PARAMETER CHANGE
770 PRINT:PRINT"DO YOU WISH TO CHANGE <P>ERCENT OR"
780 PRINT"<M>ONTHS OF REPAYMENT?"
790 GOSUB197Ø
800 IFKB=8ØGOTO84Ø
810 IFKB<>77GOTO79Ø
820 IFKB=77GOTO136Ø
830 REM    PERCENT CHANGE ROUTINE
840 CLS:INPUT"WHAT IS NEW PERCENTAGE RATE (BETWEEN 9% & 24%)";P2
850 IFP2<9ORP2>24THENPRINT"THAT RATE IS OUT OF RANGE. ENTER AGAIN"
    :INPUTP2:IFP2<9ORP2>24THENGOTO85Ø
860 PRINT@916,"STANDBY...."
870 IFP2=9THEND=1
880 IFP2=1ØTHEND=2
890 IFP2=11THEND=3
900 IFP2=12THEND=4
910 IFP2=13THEND=5
920 IFP2=14THEND=6
930 IFP2=15THEND=7
940 IFP2=16THEND=8
950 IFP2=17THEND=9
960 IFP2=18THEND=1Ø
970 IFP2=19THEND=11
980 IFP2=2ØTHEND=12
990 IF P=21 THEN D=13
1000 IFP2=22THEND=14
1010 IFP2=23THEND=15
1020 IFP2=24THEND=16
1030 LETX2=S(B,D)
1040 LETM2=C/X2:T2=M2*N1:I2=T2-C
1050 CLS:PRINT"NOW THAT $";C;"CAR WILL COST YOU"
1060 PRINT"$";T2;"OVER THE";N1;"MONTH PERIOD."
1070 IFT2>T1GOTO122Ø
1080 PRINT"THAT MAKES THE";P2;"PERCENT LOAN "
1090 PRINT@188,USINGA$;T1-T2;:PRINT" CHEAPER THAN THE";P1;"PERCENT
     LOAN."
1100 FORZ=1TO5ØØ:NEXTZ
1110 PRINTSTRING$(55,"=");
1120 PRINT
1130 PRINT"LOOKING CLOSELY, YOU WILL NOTE THAT THE MONTHLY"
1140 PRINT"PAYMENTS ON THE";P2;"PERCENT LOAN ARE $";INT(1ØØ*(M1-M2
     ))/1ØØ
```

```
1150 PRINT"CHEAPER THAN THE";P1;"PERCENT LOAN. ALSO, IF"
1160 PRINT"YOU TRADED AT THE END OF THE YEAR, YOU WOULD"
1170 PRINT"STILL OWE $";T1-(M1*12);"WITH THE";P1;"PERCENT"
1180 PRINT"LOAN. AT";P2;"PERCENT, YOU WOULD ONLY OWE"
1190 PRINT"$";T2-(M2*12);". THIS GIVES YOU MORE TRADE-IN."
1200 PRINT:PRINT"HIT 'ENTER' TO CONTINUE":INPUTB$
1210 IFB$=""GOTO710
1220 PRINT"THAT MAKES THE";P1;"PERCENT LOAN"
1230 PRINT@188,USINGA$;T2-T1;:PRINT" CHEAPER THAN THE"
1240 PRINTP2;"PERCENT LOAN."
1250 PRINTSTRING$(55,"=");
1260 PRINT:PRINT"YOU CAN SEE THAT THE MONTHLY PAYMENTS"
1270 PRINT"ON THE";P1;"PERCENT LOAN ARE $";INT(100*(M2-M1))/100
1280 PRINT"CHEAPER THAN THE"P2;"PERCENT LOAN. ALSO,"
1290 PRINT"IF YOU TRADED AT THE END OF THE FIRST YEAR,"
1300 PRINT"YOU WOULD STILL OWE $";T2-(M2*12);"WITH"
1310 PRINT"THE";P2;"PERCENT LOAN. AT";P1;"PERCENT, YOU"
1320 PRINT"WOULD ONLY OWE $";T1-(M1*12)
1330 PRINT:PRINT"HIT 'ENTER' TO CONTINUE":INPUTB$
1340 IFB$=""GOTO710
1350 REM   MONTH CHANGE ROUTINE
1360 CLS
1370 INPUT"HOW MANY MONTHS IS THE LOAN FOR";N2
1380 IFN2=18THENE=1:GOTO1500
1390 IFN2=24THENE=2:GOTO1500
1400 IFN2=30THENE=3:GOTO1500
1410 IFN2=36THENE=4:GOTO1500
1420 IFN2=42THENE=5:GOTO1500
1430 IFN2=48THENE=6:GOTO1500
1440 IFN2=54THENE=7:GOTO1500
1450 IFN2=60THENE=8:GOTO1500
1460 IFN2<>18ORN2<>24ORN2<>30ORN2<>36ORN2<>42ORN2<>48ORN2<>54ORN2<
     >60GOTO1470
1470 PRINT"SORRY - I CAN ONLY HANDLE FINANCING FOR 18, 24, 30, 36,
     "
1480 PRINT"42, 48, 54 OR 60 MONTHS."
1490 GOTO1370
1500 PRINT@916,"COMPUTING...."
1510 LETR=P1-8
1520 LETX2=S(E,R)
1530 LETM2=C/X2:T2=M2*N2:I2=T2-C
1540 CLS:PRINT"NOW THAT $";C;"CAR WILL COST YOU"
1550 PRINT"$";T2;"OVER A";N2;"MONTH PERIOD."
1560 IFT2>T1GOTO1710
1570 PRINT"THAT MAKES THE";N1;"MONTH LOAN"
1580 PRINT@188,USINGA$;T1-T2;:PRINT" MORE EXPENSIVE THAN THE"
1590 PRINTN2;"MONTH LOAN."
1600 FORZ=1TO500:NEXTZ
1610 PRINTSTRING$(55,"=");
1620 PRINT:PRINT"UPON EXAMINATION, YOU CAN SEE THAT THE MONTHLY"
1630 PRINT"PAYMENTS ON THE";N2;"MONTH LOAN ARE $";INT(100*(M2-M1))
     /100
1640 PRINT"MORE THAN THE";N1;"MONTH LOAN. IF YOU TRADED"
1650 PRINT"AT THE END OF THE YEAR, YOU WOULD STILL OWE"
1660 PRINT"$";T2-(M2*12);"ON THE";N2;"MONTH LOAN, BUT"
1670 PRINT"IF YOU HAD THE";N1;"MONTH LOAN, YOU WOULD OWE"
1680 PRINT"$";T1-(M1*12)
1690 PRINT:PRINT"HIT 'ENTER' TO CONTINUE":INPUTB$
1700 IFB$=""GOTO710
1710 PRINT"THAT MAKES THE";N1;"MONTH LOAN "
1720 PRINT@188,USINGA$;T2-T1;:PRINT" CHEAPER THAN THE"
1730 PRINTN2;"MONTH LOAN."
1740 FORZ=1TO500:NEXTZ
1750 PRINTSTRING$(55,"=");
1760 PRINT:PRINT"NOTICE THAT THE MONTHLY PAYMENTS ON THE";N1
1770 PRINT"MONTH LOAN ARE $";INT(100*(M1-M2))/100;"MORE THAN THE";
     N2
1780 PRINT"MONTH LOAN. IF YOU TRADED AT THE END OF THE YEAR,"
1790 PRINT"YOU WOULD STILL OWE $";T2-(M2*12);"ON THE";N2
1800 PRINT"MONTH LOAN, BUT IF YOU HAD THE";N1;"MONTH LOAN,"
1810 PRINT"YOU WOULD OWE $";T1-(M1*12)
1820 PRINT:PRINT"HIT 'ENTER' TO CONTINUE":INPUTB$
1830 IFB$=""GOTO710
1840 REM   EXIT PROGRAM
1850 CLS:PRINT:PRINT:PRINT:PRINT"          AREN'T YOU GLAD YOU ASKED
      FOR MY HELP?"
1860 PRINT:PRINT:PRINT"                  $ $  CAVEAT  EMPTOR  $ $"
1870 PRINT:PRINT:PRINT:PRINT"              G O O D   L U C K !"
1880 END
1890 DATA16.779,16.651,16.524,16.398,16.274,16.151,16.030,15.909,1
     5.790,15.673,15.556,15.441,15.327,15.214,15.102,14.992
1900 DATA21.889,21.671,21.456,21.243,21.034,20.828,20.624,20.424,2
     0.226,20.030,19.838,19.648,19.461,19.276,19.094,18.914
1910 DATA26.775,26.447,26.125,25.808,25.496,25.190,24.889,24.593,2
     4.302,24.016,23.735,23.458,23.186,22.918,22.655,22.396
1920 DATA31.447,30.991,30.545,30.108,29.679,29.259,28.847,28.444,2
     8.048,27.661,27.281,26.908,26.543,26.185,25.833,25.489
1930 DATA35.914,35.315,34.730,34.158,33.600,33.054,32.521,32.000,3
     1.492,30.994,30.508,30.032,29.568,29.113,28.669,28.235
1940 DATA40.185,39.428,38.691,37.974,37.275,36.595,35.931,35.285,3
     4.656,34.043,33.445,32.862,32.294,31.940,31.120,30.673
1950 DATA44.179,43.246,42.342,41.465,40.612,39.786,38.983,38.203,3
     7.446,36.712,35.997,35.303,34.629,34.073,33.297,32.717
1960 DATA48.173,47.065,45.993,44.955,43.950,42.977,42.035,41.122,4
     0.237,39.380,38.550,37.745,36.964,36.207,35.473,34.761
1970 KB$=INKEY$:IFKB$=""GOTO1970
1980 KB=ASC(KB$):RETURN


DONE
```

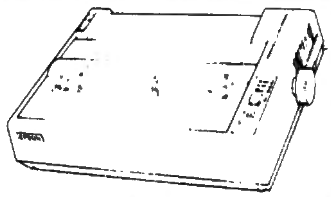



BUSINESS COMPUTERS

WE HONOR master charge

*TM TANDY CORP.

THE MX-80 NOT ONLY DOES EVERYTHING, IT DOES EVERYTHING WELL.



This is the new Epson MX-80 dot matrix printer. It does just about everything you could ask a printer to do. Quickly. Quietly. Reliably. In fact, for OEM installations, the MX-80 may be the single best, all-round printer you can buy. But that's not the best reason to buy it.

The MX-80 prints bidirectionally at 80 CPS in a user-defined choice of 40, 80, 66 or 132 columns. And if that's not fast enough, its logical seeking function minimizes print head travel time. The MX-80 prints 96 ASCII, 64 graphic and eight international characters with a tack-sharp 9x9 matrix. For a long time. Epson printers are known for reliability and the MX-80 is no exception. But that's not the best reason to buy it either.

The print head has a life expectancy of up to 100 x 10^6 characters, and when it wears out, just throw it away. A new one costs less than $30 and the only tool you need to change it is attached to the end of your arm. The MX-80 is compact weighs only 12 lbs., and the whole unit, including the two stepper motors controlling carriage and paper feeding functions, is precisely controlled by an internal microprocessor. But even that isn't why you should specify the MX-80.

The best reason is this: because Epson makes more printers than anyone else in the world, we can afford to sell each one for a little less.

...Call For Unbelievably 'Low Price

317

OKIDATE MICROLINE 80 PRINTER - $479

The Microline is built on a rugged cast aluminum base to withstand the rigors of continuous business use. It is driven by two motors and will run all day with no duty cycle limitations. Microline printers use a dense 9 x 7 dot pattern to produce crisp, clean copies, first copy to last. The seven pins in the head are "fired" using energy stored in tension members. This technology permits the use of short, low mass pins made with an extremely hard alloy. The head produces less heat, thereby extending its life. 356

900 C.P.S. PRINTER

That's right nine hundred characters per second. That's the print speed that can be obtained using the new Florida Printer Model 600 C. The 600 C uses a unique "Stored-Energy" print head to obtain a print speed of 600 CPS in normal print font and 900 CPS in a condensed print font. This printer is not inexpensive but it is definitely the fastest dot matrix printer in the world. Call for prices... 211

AW...WHAT THE HECK

RAM Memory Chips for the TRS-80

It is the policy of American Business Computers to offer merchandise at the lowest price possible. Serveral months back we began selling RAM Memory Chips for the TRS-80 for $45.00 per set. Someone else sold chips for $44.00. We sold them for $38.00. They sold them for $37.95. So we say "AW ..... WHAT THE HECK!" Let's see the other guys beat this price.

$24

PER 16K SET

These chips are brand new "4116's". These 200 nanosecond chips are fully compatible with all TRS-80 produces. Instructions for insertion are included, however the dip shunts required for converting a 4K Model I to a 16K Model I are not included at this low price.

322

Epson.

This printer costs less than $450. just how much less we can't tell you - But if you will give us a call we think you will be pleasantly suprised.

This is the Epson MX-70. The lowest priced dot matrix printer you can buy. Now, that in itself should make it very attractive to a lot of people. But you ain't heard the half of it.

To begin with, the MX-70 has a lot more in common with our now-famous MX-80 than just the name. Like unequalled Epson reliability. And technological breakthroughs like the world's first disposable print head. But frankly, the MX-80 packs a lot more power than some people need. So Epson built the MX-70 to be a no-frills printer. At a no-frills price.

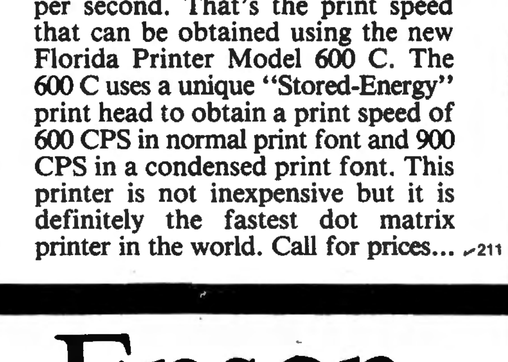
But the MX-70 is still a great little printer. They give you 80 CPS unidirectional printing. Top-of-form recognition. Programmable line feed and form lengths. Plain paper printing. An easy-to-read 5x7 matrix. Self test. And an adjustable tractor feed.

That's what you'd expect from a basic little printer. But here's something you wouldn't expect: the finest graphics package on the market today. Free.

They call it GRAFTRAX II. And it means 480 dots across the page, resolution to 60 dots per inch, and a graphic image free of the jitter and overlap that plagues other printers. You get cleaner grays and finer point resolution.

So now you've got a choice. You want more power and extra functions, you buy the MX-80. You want a basic little printer that prints, and keeps on printing, you buy the MX-70. They're both at American Business Computers.

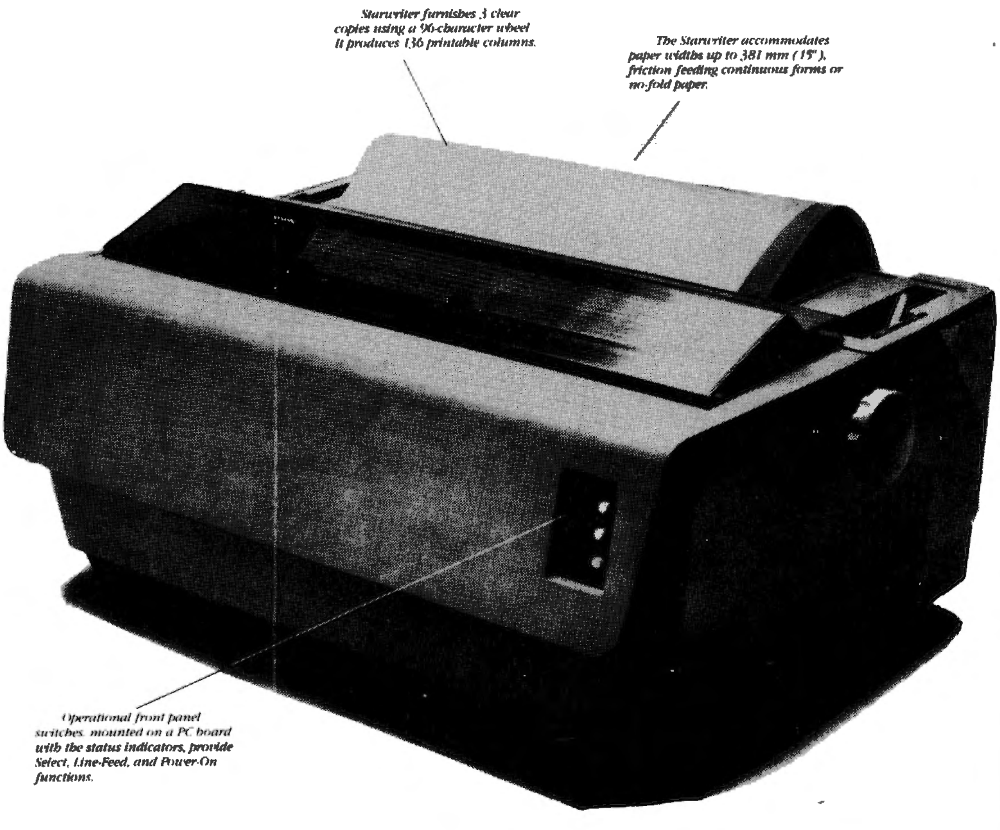
Call for unbelievably low price.



196

NEED A LETTER QUALITY PRINTER BUT DON'T WANT TO PAY THE PRICE?

Then American Business Computers Has Just The Printer For You!



The Starwriter Daisy Wheel Printer is designed to fill a distinct gap in the peripheral marketplace. A speed of 25 cps and its low price make it the ideal choice for today's systems designer interested in achieving superior price/performance ratios.

Incorporating the latest LSI technology, the Starwriter is built by one of the world's most respected computer peripheral manufacturers. The printer is furnished complete and ready-to-use, requiring no changes in hardware or software. Starwriter offers an industry standard parallel interface, as well as RS232-C interface. Total plug compatibility and a wide variety of interface matching capabilities help lower the system integration costs for OEMs and end-users.

Using a 96-character wheel, the Starwriter produces letter-quality printing on 3 sharp copies with up to 136 columns. Starwriter offers the highest degree of vertical and horizontal positioning, resulting in the most precise character placement in the industry. The easy-to-change character wheel also makes our printer a perfect choice for international applications. Compatible with sheet feeders, starwriter accommodates paper widths to 381 mm (15") and uses industry standard ribbon cartridges.

Along with a self-test capability and a programmable VFU (Vertical Format Unit), Starwriter furnishes the operator with all desirable status functions, commands and program selectable switches. Panel lamps indicate the current Paper, Select and Power status.

The easy plug-in compatibility of Starwriter and its outstanding print performance makes the Starwriter perfect for TRS-80*, Apple*, and S-100 users.

Starwriter is backed up by a one-year warranty (3 months on parts and labor, 9 succeeding months on parts) and is further supported by a strong, fully trained and technically proficient service organization. The Starwriter is available for immediate shipment in any quantity desired.

For more information, call American Business Computer, 118 So. Mill Street, Pryor, Oklahoma 74361; Telephone 918-825-4844.

*TM of Tandy Corp
*TM of Apple Computer Corp.

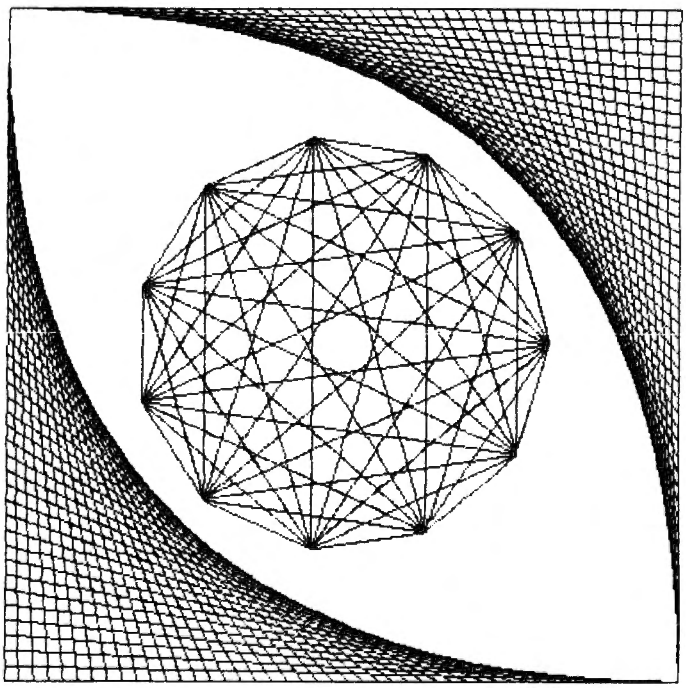
AMERICAN BUSINESS COMPUTERS ✓267

118 So. Mill Street
Pryor, Oklahoma 74361
Telephone 918-825-4844

Dealer Inquiries Invited

Advanced Graphic Techniques—Part II

Bob Boothe
4651 Browndeer Lane
Rolling Hills Estate, CA 90274



Design created in two minutes, printed on a Base II dot matrix printer.

I have always enjoyed creating designs and plotting shapes and surfaces with my TRS-80. I started plotting three dimensional surfaces on the video screen, then went to the printer because of the increased resolution I could get. I developed a program for plotting a picture on many sheets of paper, and then assembling the sheets into one big picture. For my first and only run of this, I made a plot which covered twenty sheets of paper. It took approximately thirty hours of computer time to create it.

Things were getting out of hand: I had to find a way to cut down the time. My first breakthrough was using the graphic capabilities of my Base 2 printer. I created a replica of my twenty page plot by using single dots instead of characters, which took about eight hours to complete.

The number of points which can be stored in memory is staggering. Considering that we have 32,000 bytes and that each dot needs only one bit, 32,000 bytes times eight bits per byte gives us 256,000 points which can be stored. That is a quarter of a million points. I'll bet you never realized that a TRS-80 had that much power.

The first design we will create will be 406 by 406 dots. This size gives very good resolution, and could not be done on the video even if a high resolution system was developed which would allow control over every single dot. A picture this size gives us a total of 164,836 dots which we can use. Estimating that only ten percent of the dots will be turned on in any design, about 16,000 dots will need to be turned on.

It takes about 30 seconds for a FOR-NEXT loop to go through 16,000 loops; it could take several hours to set 16,000 points using a BASIC routine. Machine language seems to be the only answer. However, trigonometric functions are used to create all of my designs. I have never written a cosine routine in machine language, so we have to scratch that plan.

My graphic system provides the best of both worlds—all the designs in this article were created with straight lines. Yes, all the curves are made by combining straight lines in specific trigonometric patterns. Aha!

Each design contains a few hundred straight lines. The endpoints are computed in BASIC, and the lines are drawn with machine language. This way the complicated computations can be done in BASIC, so they can easily be changed and corrected. All the time-consuming and repetitive tasks are done in the machine language program. Since the machine language routine only draws lines as specified by the BASIC program, it never needs to be changed.

Picking Parameters

Before I explain the programs themselves, I am going to explain some of the computations I made before writing the programs. Most of you will need to do some conversions to make the program print on your printer.

I decided to use about three quarters of available memory to store the design, leaving 8K for the programs. The Base 2 printer accepts graphical data for seven dots at a time (bits 0-6). The seventh bit is ignored. In order to make my printout routine as simple as possible, I stored the data in only the first

"Those of you lucky enough to have a Level II TRS-80 with 32K RAM, no disks and a Base 2 printer should be able to run the program as is."

seven bits of each byte and wasted the last bit, just as the printer does. Out of my 24,000 bytes, I used seven dots each, for a total of 168,000 points.

I wanted my designs to be square, so I found the square root, which is about 410. Furthermore, the number should be divisible by seven to make the print-out routine as easy as possible. 406 is a good number, making 58 lines on the print-out and taking about two minutes (graphics are done at half speed). Fifty eight lines times 406 bytes per line gives 23,548 bytes needed to store the design. Those with 16K machines should be able to shrink everything down and squeeze out a smaller picture.

The first problem is to make 406 horizontal dots the same length as 406 vertical dots. Each line on the Base 2 printer consists vertically of seven printed dots and five dots for spacing: Twelve dots per line times six lines per inch is 72 dots per inch. To match this on the horizontal scale, it takes 72 dots per inch divided by six dots per horizontal character (five dots for the character and one for the space) or 12 characters per inch. Fortunately, 96 characters per line is exactly 12 characters per inch. This can all be software selectable by sending a specific series of numbers to the printer.

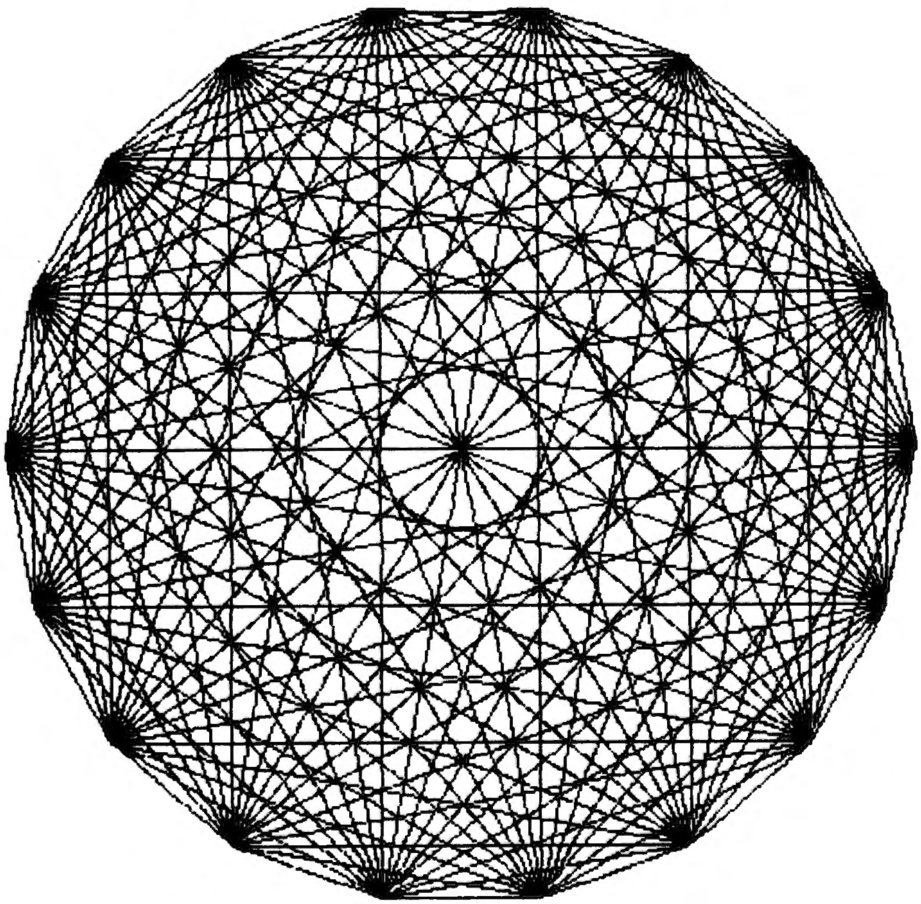


Figure 1: A common computer generated design, created by drawing all the diagonals in an N-sided regular polygon.

Machine Language

Those of you lucky enough to have a Level II TRS-80 with 32K RAM, no disks and a Base 2 printer should be able to run the program as is. Those with other printers and 32K should only need to modify the printing section. If your printer does not have graphics capabilities, you may want to modify the program to use the video monitor (explained in Part III).

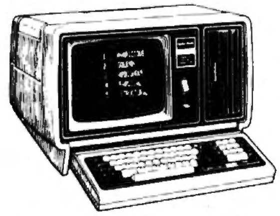
Three disk commands all used in this program. The first is for the line drawing routine; LINE fits very well. The second routine initializes the design memory by filling the entire block with zeros. I call this routine OPEN, since it is the starting process. The last routine is the printing routine, which I call CLOSE because it is the finishing process. The program presented last month POKED the address of our routine into location from the table. This time, since the program will be loaded using the SYSTEM command, the disk commands can be initialized while the program loads.

EDTASM doesn't care where we put the program, so we can load directly into reserved RAM. I set each origin in the program one byte below the corresponding address from the table, to compensate for the jump which has to be saved in order to write the address.

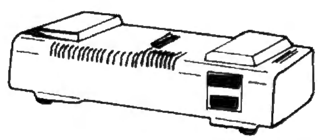
The LINE routine uses four values to draw a line. These are the start and the end of the line for the horizontal and vertical directions. My labels are STARTX, STARTY, ENDX and ENDY. The easiest way to supply these is by POKEing them into the program.

```
100 REM  DESIGN NO. 1
110 INPUT"NUMBER OF POINTS (SAMPLE HAS 18) ";N
120 DIM A(N),B(N)
130 FORT=0TO2*PI-.001 STEP 2*PI/N
140 Z=Z+1
150 A(Z)=COS(T)*200+202:B(Z)=SIN(T)*200+202
160 NEXT T
170 FORS=1TON-1:FORD=S+1TON
180 X1=A(S):Y1=B(S)
190 X2=A(D):Y2=B(D)
270 GOSUB1500
280 NEXT D,S
```

meet a REAL PLUS

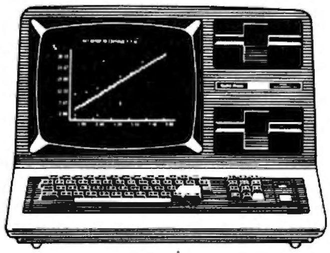


Model II 64K $3375



32K Exp. Int. $362

Color Computer 4K $310



Model III 16K $859

check out these unusual package values for price and warranty

Here are just a few of our fine offers . . . call toll-free for full information.

COMPUTERS

Model II 64K	$3375
Model III 4K LEV I	599
Model III 16K	859
Model III 32K	981.50
+ Model III 32K	915.50
Model III 48K	1104
+ Model III 48K	972
Model III 32K 2 Disc & RS232 c	2149
Color Computer 4K	310
Color Computer 16K	439.95
+ Color Computer 16K	366.50
Color Computer 16K w/extended basic	489
Pocket Computer	199
VIDEOTEX	329
APPLE 48K only	1279
ATARI 800 16K	789

+ Computer Plus New Equipment. 180 Day Extended Warranty

PERIPHERALS

Expansion Interface 0K	$249
Expansion Interface 16K	359
+ Expansion Interface 16K	305.50
Expansion Interface 32K	469.95
+ Expansion Interface 32K	362
16K RAM N.E.C. 200 N.S. chips	39

MODEMS

Lynx Direct Connect	219
COMM 80 Interface	159.95
Chatterbox Interface	239
Telephone Interface II	169

PRINTERS

Line Printer IV	849
Daisy Wheel II	1695
Line Printer VI	999
NEC Spinwriter 5530	2595
OKI DATA Microline 80	499
EPSON MX80	519

DISK DRIVES

Model III 1-Drive	712
PERCOM TFD 100	389
TEAC 40 Track	319

MODEL I 32K E.I., 1st. Drive, Line Printer IV, 10 Diskettes — Pure R.S. $1875; combined warranty 1699

MODEL II 64K w/2 Disks, Daisy Wheel II, Scripsit — Pure R.S. 6350

MODEL III 48K, 1st. Drive, Line printer IV, Scripsit Word Processor, 10 Diskettes — Pure R.S. 2800; combined warranty 2699

COLOR COMPUTER 16K CC w/extended basic cassette recorder, joy sticks, dust cover — Pure R.S. 569

POCKET COMPUTER w/interface — Pure R.S. 239

ATARI
48K Atari 800, Atari 825 Printer, Atari 810 Disk Drive — 2125

APPLE
48K Apple II & Apple II Disk w/controller, D.O.S. 3.3, Printer Card and Cable, Epson MX80 Printer — 2510

PLUS real back-up warrantees —

Pure Radio Shack equipment warranteed at any Radio Shack store or dealer. Factory warrantees on Apple and Atari equipment. Other equipment carries manufacturer's warranty or Computer Plus 180 day extended warranty. Combined warrantees carry Computer Plus 180 day warranty or original manufacturer's warranty.

DEALER INQUIRIES ARE INVITED (617) 486-3193

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

TRS-80 is a registered trademark of Tandy Corp.

call TOLL FREE 1-800-343-8124

computer plus ✓130

Write for your free catalog ...

Dept. D
245A Great Road
Littleton, MA 01460

These values will range from 0 to 405. Because 405 is too large to be stored in a single byte, an entire word will have to be used.

The process used to draw a straight line becomes tricky. Let's take the simplest case first. We want to draw a line from (3,6) to (7,6), a short horizontal line. The points which should be set are: (3,6), (4,6), (5,6), (6,6) and (7,6). A slightly more difficult line is from (8,8) to (11,11). The points for this line are: (8,8), (9,9), (10,10) and (11,11). Consider a line from (2,2) to (6,4); the best points would be: (2,2), (3,2), (3,3), (4,3), (5,3), (5,4) and (6,4).

Try writing a program to draw an unbroken line on the video between any two points, and you will see there is a problem. The third example has a total movement of four units in the X direction and two units in the Y direction. This would mean a movement of one-half Y for each X.

The program uses fractions to determine the amount to move each step. The labels are DIRX and DIRY. The fraction for the X direction is computed in lines 320-420. The registers are first saved with an EXX. Four bytes are used to store each variable used to draw the line: Two bytes for the whole part, and two for the fraction. The fraction or LSW (least significant word) is stored first, followed by the whole part.

The step is computed by subtracting the start from the end value. The difference is used as the step, and is actually divided by 65,536. If the difference is negative, the MSW is also made negative. This procedure is then repeated for the y direction in lines 430-530. Lines 540-760 double both fractions until one is greater than or equal to one-half. This optimization allows the program to operate as rapidly as possible by making the total number of points to be set as small as possible. The process is done by shifting the fractions left until the most significant bit changes from zero to one for positive directions, or from one to zero for negative directions.

The next section of the program draws the line. It first sets the starting point and then adds directions to the start variables until the start is equal to the end. It sets each point as it goes along, and checks to see if STARTX is equal to ENDX after each point is set. The X variables should always meet at the same time, except for the one case of vertical lines. This is why the Y variables are compared in the MAYBE routine at line 1000. If the Y directions are also the same, the registers are restored with an EXX, and the routine returns control to the

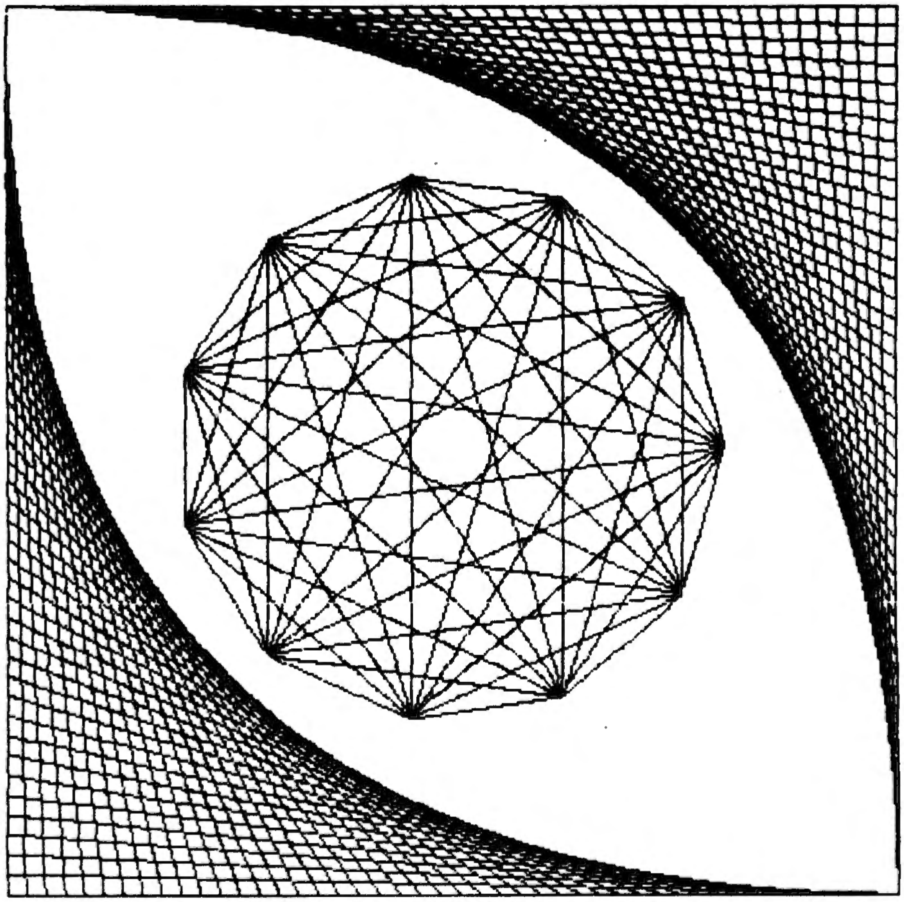
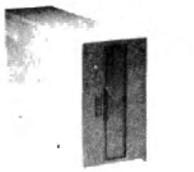


Figure 2: This is the same design which was seen last month. It combines three shapes to form an eye.

```
40 REM  DESIGN NO. 2
50 FOR Q=0TO405STEP9
60 X1=0:Y1=Q:X2=Q:Y2=405
70 GOSUB1500
80 X1=Q:Y1=0:X2=405:Y2=Q
90 GOSUB1500
100 NEXT Q
110 INPUT"NUMBER OF POINTS (SAMPLE HAS 18) ";N
120 DIM A(N),B(N)
130 FORT=0TO2*PI-.001 STEP 2*PI/N
140 Z=Z+1
150 A(Z)=COS(T)*125+202:B(Z)=SIN(T)*125+202
160 NEXT T
170 FORS=1TON-1:FORD=S+1TON
180 X1=A(S):Y1=B(S)
190 X2=A(D):Y2=B(D)
270 GOSUB1500
280 NEXT D,S
```

WE WILL NOT BE UNDERSOLD

DISK DRIVES



FOR TRS-80* Model I

CCI-100 TEAC	5¼", 40 Track (102K)	**$314**
CCI-100 MPI	5¼", 40 Track (102K)	**$319**
CCI-280	5¼", 80 Track (204K)	**$429**

ADD-ON DRIVES FOR ZENITH Z-89

CCI-189	5¼", 40 Track (102K)	**$394**
CCI-289	5¼", 80 Track (204K)	**$499**
Z-87	Dual 5¼" system	**$995**

External card edge and power supply included. 90 day warranty/one year on power supply.

RAW DRIVES 8" SHUGART 801R **$425**
5¼" MPI, TEAC or TANDOM **$ CALL** POWER SUPPLIES **$ CALL**

MORROW DESIGNS/THINKER TOYS™

DISCUS 2D	1 DRIVE	**$ 938**	2 DRIVE	**$1635**
DISCUS 2+2	1 DRIVE	**$1259**	2 DRIVE	**$2245**
DISCUS Hard Disk	M26	**$3990**	M10	**$2999**

DEI CARTRIDGE TAPE BACK-UP

For your hard disk. With either S-100 control card or piggy back board for single board Z-80 computers. **$2995**

DISKETTES — Box of 10 **with plastic library case**

5¼"	**Scotch**	**$35**	**Maxell**	**$40**	**BASF/Verbatim**	**$27.95**
8"	**Scotch**	**$50**	**Maxell**	**$55**	**BASF/Verbatim**	**$36.00**

PLASTIC CASE-Holds 50 5¼" diskettes **$19.00**
SCOTCH HEAD CLEANING DISKETTE **$25.00**
FLOPPY SAVER **$11.95** RINGS **$6.95**

COMPLETE SYSTEMS

ARCHIVES	64K, 2-Drives, 77 Track			**$5495**
ALTOS	ACS8000 Series			**$ CALL**
TRS-80*	II-64K	**$3499**	III-16K	**$ 899**
TRS-80*	Expansion Interface			**$ 299**
ZENITH	48K, all-in-one computer			**$2200**
ZENITH	Z-19			**$ 735**
TELEVIDEO	920C	**$ 748**	950	**$1049**
IBM	3101 Display Terminal			**$1189**
ATARI	400	**$ 479**	800	**$ 795**
MATTEL	INTELLIVISION			**$ 229**
APPLE PERIPHERALS				**$ CALL**

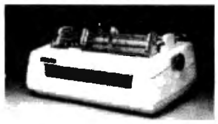
MONITORS

APF	9" B & W TVM-10			**$120**
BELL & HOWELL	9" B & W BHD911			**$220**
LEEDEX	12" B & W	**$129**	13" Color	**$369**
SANYO	9" B & W VM4509			**$155**
SANYO	12" B & W DM5012			**$226**
SANYO	12" Green Screen DM5112			**$238**
SANYO	13" Color DMC6013			**$416**
ZENITH	13" Color			**$349**

TELECOMMUNICATIONS

LIVERMORE STAR MODEM 2-year guarantee	**$125**
UNIVERSAL DATA SYSTEMS UDS-103	**$179**
D-CAT HARD WIRED DIRECT MODEM	**$189**
AUTO-CAT Auto Answer, Direct Connect Modem	**$229**
D.C. HAYES MICRO-MODEM	**$329**

PRINTERS



NEC Spinwriter
Letter Quality High Speed Printer

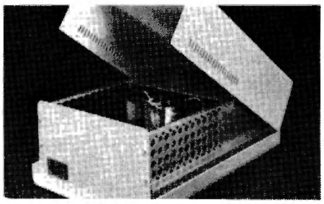
R.O.	**$2395**
R.O. with tractor feed	**$2595**
KSR with tractor feed	**$2895**

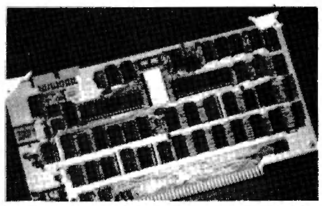
C.ITOH	Starwriter	**$1695**	Starwriter II	**$1895**
EPSON	MX-80	**$479**	MX-70	**$ 399**
PAPER TIGER				
IDS 445	Graphics & 2K buffer			**$ 699**
IDS 460	Graphics & 2k buffer			**$1050**
IDS 560	Graphics			**$1450**
ANADEX	DP-8000	**$849**	DP-9500/01	**$1345**
OKIDATA				
Microline 80	Friction & pin feed			**$ 420**
Microline 80	Friction, and pin & tractor feed			**$ 520**
Microline 82	Friction & pin feed feed			**$ 620**
Microline 83	120 cps, uses up to 15" paper			**$ 849**
CENTRONICS	730 **$ 595**	737 **$ 749**	779	**$ 969**
TI-810				
TRS-80* software, compressed print & vert. form control				**$1865**

16K RAM KITS **2 for $56** **$30**

200 ns for TRS-80*, Apple II, (specify): **Jumpers $2.50**

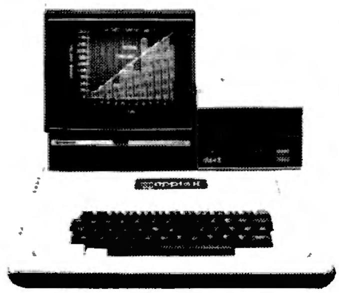
S-100 CALIFORNIA COMPUTER SYSTEMS





MAINFRAME	Model 2200A	**$349**
Z80 CPU	Model 2810	**$269**
MOTHER BOARD	Model 2501	**$106**
16K STATIC RAM, 200ns	Model 2116C	**$309**
32K STATIC RAM, 200ns	Model 2032C	**$619**
64K DYNAMIC RAM	Model 2065C	**$599**
FLOPPY DISC CONTROLLER	Model 2422A	**$359**
EXTENDER BOARD	Model 2520K	**$ 52**
2P + 2S I/O	Model 2718A	**$309**

APPLE ACCESSORIES AND SOFTWARE



VISICALC	**$120.00**
DB MASTER	**$159.00**
Z-80 SOFTCARD	**$269.00**
VIDEX BOARD	**$259.00**
16K CARD	**$169.00**
APPLE JOYSTICK	**$ 49.00**
SUPERMOD	**$ 29.00**
CCS CARDS	**$ CALL**
GALAXIAN	**$ 22.95**
SPACE ALBUM	**$ 35.00**
ASTEROIDS	**$ 17.95**
FLIGHT SIMULATOR	**$ 29.00**
WIZARD & PRINCESS	**$ 28.00**
SARGON 2	**$ 29.00**
HI-RES FOOTBALL	**$ 35.00**
MYSTERY HOUSE	**$ 21.00**

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP

298

TO ORDER CALL TOLL FREE 1-800-343-6522
TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept.M05M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/®Digital Research

master charge THE INTERBANK CARD

VISA

See List of Advertisers on page 306

program.

The SET routine first gets the Y coordinate and divides it by seven to determine how many lines down the point is. The program then adds 406 bytes for each line down in the Y direction. The sum is added to 25603, which is the first byte of the design memory. After this is done, the X coordinate is added to the previous sum. The remainder from the earlier division by seven is equal to the bit which needs to be set. A table provides bit patterns which are combined with the previous contents of the byte being changed.

The OPEN routine is considerably simpler. It puts a 128 in the first byte of the design memory and then copies it through the entire block. One hundred twenty eight is used instead of zero because the printer ignores the seventh bit, and if zero is sent to the printer, the printer will never know it is there.

The PRTOUT routine in lines 1480–1540 is used to check the printer status and then output the byte in the A register. This routine is modified from an incorrect routine in the T-BUG manual.

The first section of the CLOSE routine outputs the control sequence to select 96 characters per line and seven dots per line spacing. Seven dots (actually fourteen half dots) is the standard linefeed spacing when using graphics, eliminating all space between lines. (The control codes used by the Base 2 are industry standard codes, but this doesn't necessarily mean that other printers will work this program.)

Line 1660 sets up the counter for 58 lines of printing. Inside the loop, the printer is told to accept graphical data with the control sequence 27, 99. Once it is in the graphic mode, 85 blanks are sent, then 406 characters, and then another 85 blanks; 85 + 406 + 85 = 576, the number of bytes which must be sent in the 96 character per line mode. Finally, a carriage return is sent to the printer to cause a line feed. This process is repeated for all 58 lines, and then the routine is ended.

The last address of the program should be 25602 (6402H), which is just below the start of the design memory (25603).

The entry point for entering the slash after loading was set at the entry point to Level II BASIC from T-BUG. This sometimes gives an OM error, but it is fixed by a CLEAR. A better way to exit the system mode is to press the break key. Create the source code and load it before you start typing in the BASIC programs.

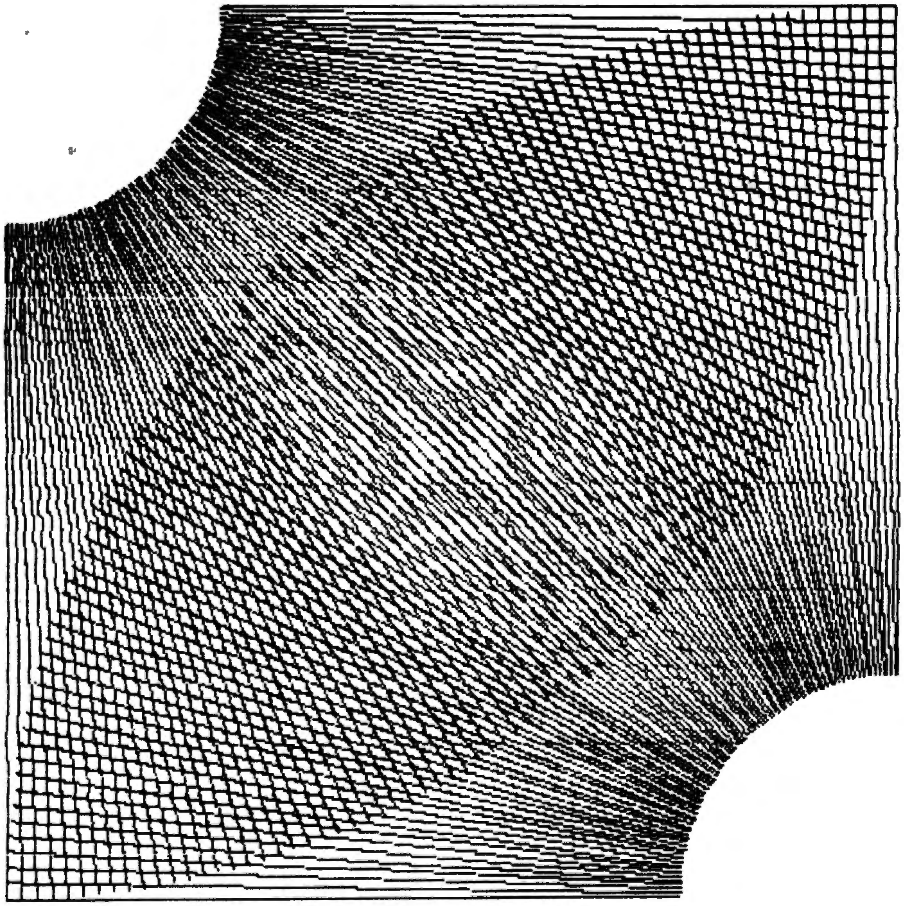


Figure 3: A moire pattern created by the interference of regularly spaced radii of two circles.

```
100 REM  DESIGN NO. 3
110 FOR T=0TOPI/2STEPPI/180
120 X1=FIX(COS(T)*100):Y1=SIN(T)*100
130 X2=FIX(COS(T)*405):Y2=SIN(T)*405
140 GOSUB1500
150 X1=405-X1:Y1=405-Y1
160 X2=405-X2:Y2=405-Y2
170 GOSUB1500
180 NEXT T
```

By the way, I've discovered an easy way to load system programs. Instead of typing EDTASM to load the Editor/Assembler, you can type E, ED, EDT, EDTA, EDTAS, or even EDTASMBLER. I don't know why this works, but you might as well use it. Don't forget to set memory size to 25239.

Back to BASIC

Program 1 is a simple BASIC routine which controls the machine language routines. Line 20 contains the OPEN which clears the design memory. If this statement is left out, each new design will be added to the previous contents of the memory, mak-

"My original intention was to produce a square spiral in which the distance between the lines increased as the length of the sides grew longer. I ended up with this design which I call "The Tunnel."

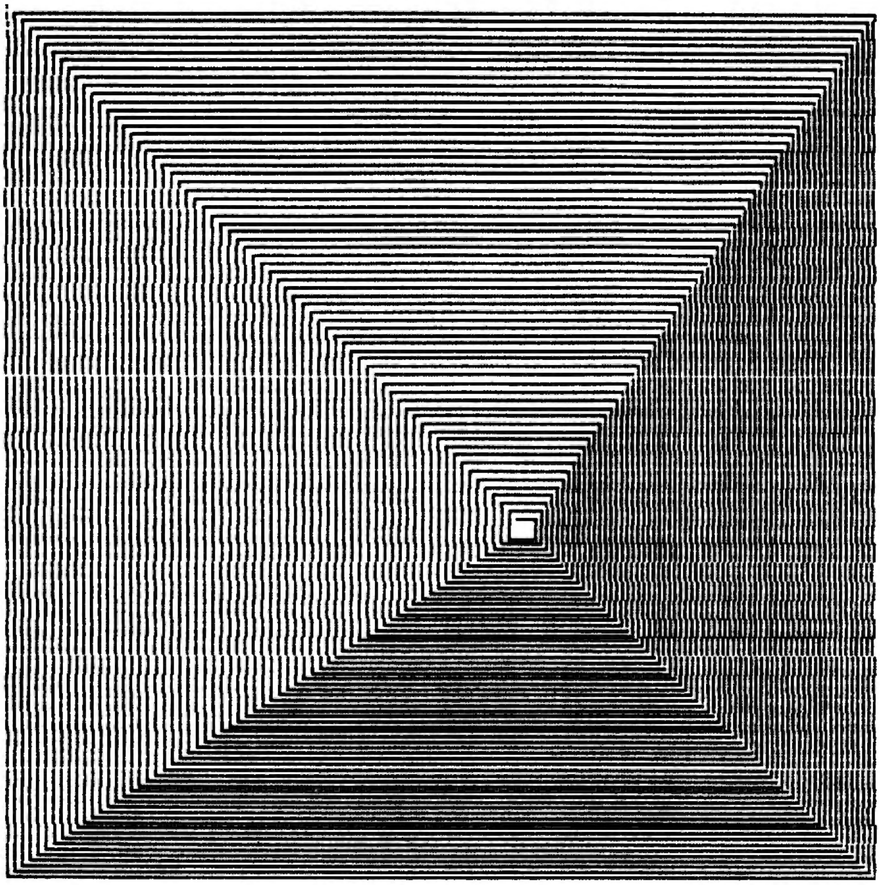


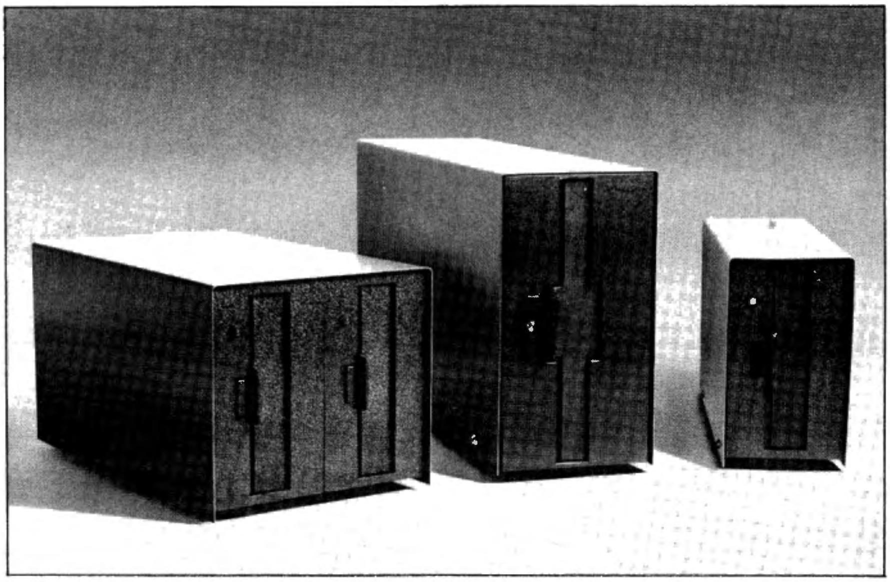
Figure 4: "The Tunnel" is a square spiral.

ing a genuine mess. Line 30 sets the value of π, which is used in almost all of the designs. The empty section from line 31 to line 999 can be used for writing individual design programs. Line 1000 provides an opportunity to stop a design before it is printed. Line 1010 contains the CLOSE routine which prints the design on the printer. Once the printing is started, there is no convenient way to stop it.

Line 1500 is the BASIC portion of the LINE routine. At the start in line 1510, a < sign is printed to signify that the routine was started. These signs will help you estimate the number of lines which are drawn in each design. For example, if only three symbols were printed, there was probably an error in the program and it should be stopped. The next four lines check to see if a value is out of range. If a number is too large or small, the program is terminated. Line 1560 checks to see if the two endpoints of the line are the same. If they are the same, a warning message is printed, but the program is not ended.

```
100 REM  DESIGN NO. 4
110 X1=270:Y1=270
120 FORQ=1TO64
130 X2=X1+5*Q+2:Y2=Y1
140 GOSUB1500
150 X1=X2:Y1=Y2+5*Q+3
160 GOSUB1500
170 X2=X1-5*Q-5:Y2=Y1
180 GOSUB1500
190 X1=X2:Y1=Y2-5*Q-6
200 GOSUB1500
210 NEXTQ
```

Spiral designs usually have values too



More power to you.

Disk drives, hardware and software— now more affordable *and* more available!

NOW IN STOCK! MODEL III DISK CONTROLLER BOARDS!

All retrofit packages include circuitry, power supplies, mounting brackets, cables and everything required to turn your Model III into a reliable, disk-based system. 100% compatible with Radio Shack hardware!

AM-1 — Supports 4 single-sided, double-density 5¼-inch disk drives **$445**

AM-2 — Supports 4 double-sided, double-density 5¼-inch disk drives **$545**

AM-3 — Supports 2 internal 5¼-inch single-sided, double-density disk drives and 2 external 8-inch single-sided, double-density disk drives **$645**

TRS-80™ SOFTWARE

MAKE 80©
Converts standard 35 or 40 track diskettes into 80-track-readable diskettes **$14.95**

Model III TRSDOS
40-track Model III TRSDOS & manual **$20**
80-track Model III TRSDOS & manual **$35**

Complete Service Center:
Speedy repairs, including modified TRS-80's.

New Store Hours:
Monday-Friday 10-6, Sat. 10-5

™TRS-80 is a trademark of Tandy Corp.

SPECIALS!

SAVE ON COMPLETE DISK DRIVE PACKAGES!

5¼-inch drives
40-track MPI 51 with case, power supply and extender cables **$320**
80-track MPI 91 with case, power supply and extender cable **$425**
Dual-headed 80-track drive with case, power supply **$550**

8-inch drives
Single Siemens 8-inch drive with single case & power supply **$695**

TEAC DISK DRIVES— THE BEST WE'VE SEEN TO DATE!

Teac 40-track disk drive
Call for our low price!
Teac 80-track disk drive
Call for our low price!

PRINTER SPECIAL!

Epson MX-80 Printer
Call for our low price!

The power behind the drives©

452

A.M. ELECTRONICS, INC.
3366 Washtenaw Ave.
Ann Arbor, Michigan 48104
(313) 973-2312

NEW TRS-80™ MODEL III! NOW WITH DISK DRIVES!

Model III with dual 80-track disk drives—704K of reliable disk storage on only two drives!

- Features two double-density 80-track drives
- 48K RAM
- Includes A.M. Electronics controller board and versatile MAKE80© program
- Complete, ready-to-run

Now only **$2295!**

The Ultimate Small Business Computer: Model III with four dual-headed 80-track disk drives!

- Features four drives
- 48K RAM
- 2.8 Megabytes reliable disk storage
- Includes A.M. Electronics controller board and versatile MAKE80© program
- Complete, ready-to-run

Now only **$3,795!**

CASES & POWER SUPPLIES

5½-inch enclosures
Single drive unit case and power supply..................... **$85**
Dual drive unit case and power supply.................... **$120**
(Extender cables $15 each extra)

8-inch enclosures
Single drive unit case and power supply.................... **$150**

Attention OEM's & Distributors:
Call us for details on our attractive pricing!

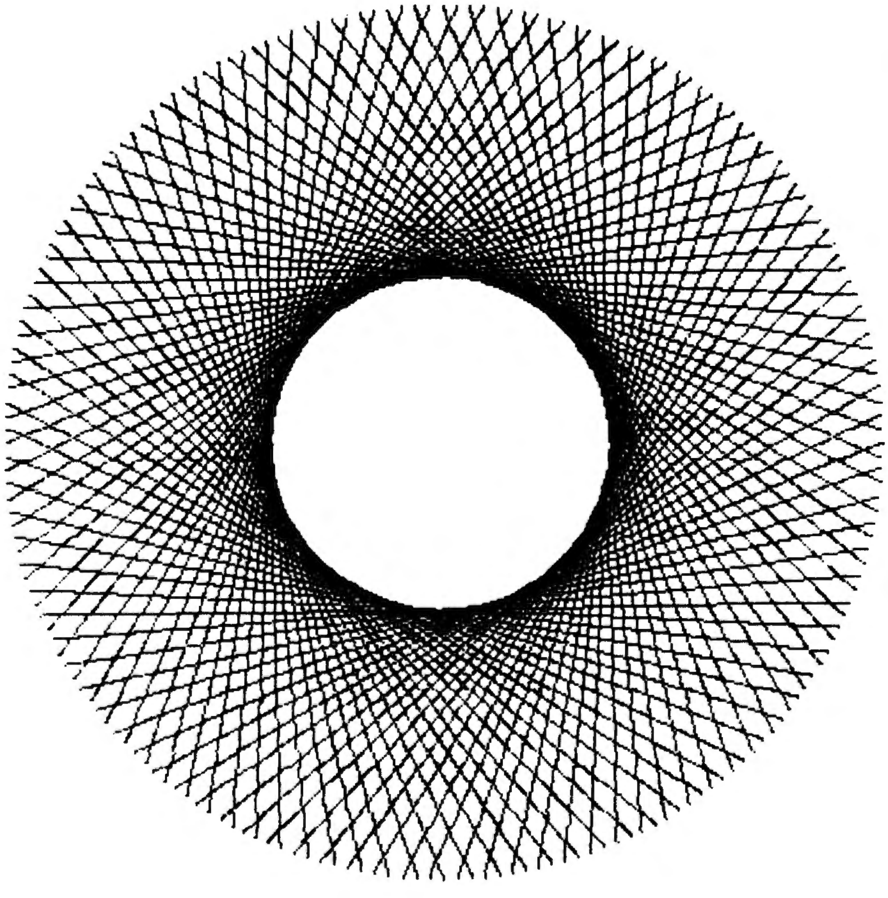


Figure 5: A very simple disk made with one hundred straight lines.

```
100 REM  DESIGN NO. 5
110 FOR T=0 TO 2*PI STEP PI/50
120 X1=COS(T)*200+202:Y1=SIN(T)*200+202
130 A=T+3*PI/4
140 X2=COS(A)*200+202:Y2=SIN(A)*200+202
150 GOSUB 1500
160 NEXT T
```

close together in the very center of the design. These checks are important because if two points are the same, both directions will be zero, and the optimization portion will become an endless loop, all bits in the fraction being zero. The range checks prevent the routine from drawing lines through forbidden territory such as the area containing the program.

The LINE routine should be provided with the value of points (X1,Y1) and (X2,Y2). After these values are checked for validity, they are divided into LSBs and MSBs. These are POKEd into the most significant part of the machine language routine's variable section. If you modify the program, you will need to adjust these POKE addresses. Be careful when you type this section because one wrong POKE can cause real problems. Line 1660 calls the LINE routine, and the last two lines print a > sign and return to the calling program.

✓See List of Advertisers on page 306

> *"Four of the designs presented in this article are not all like I had planned. It turns out that the accidental designs are often the best. . ."*

To use this routine, put the endpoints of a line in X1, Y1, X2 and Y2 and then GOSUB 1500. Never type the word LINE without having previously POKEd values into the program, or the program will hang up.

If It's Good, It's Wrong

Four of the designs presented in this article are not at all like I had planned. It turns out that the accidental designs often look the best, however. "If it's good, it's wrong" might be a good rule to add as one of Murphy's corollaries.

Fig. 1 is a commonly seen computer design. It consists of an N-sided regular polygon with all of the diagonals connected. It can be done with any number of side points, but figures with more than 25 points start to become too dark. Odd numbers make designs with a blank circle in the center, while even numbers have a filled center.

Line 110 asks for the number of side points, and line 120 dimensions the array with enough space. A is the X coordinate of

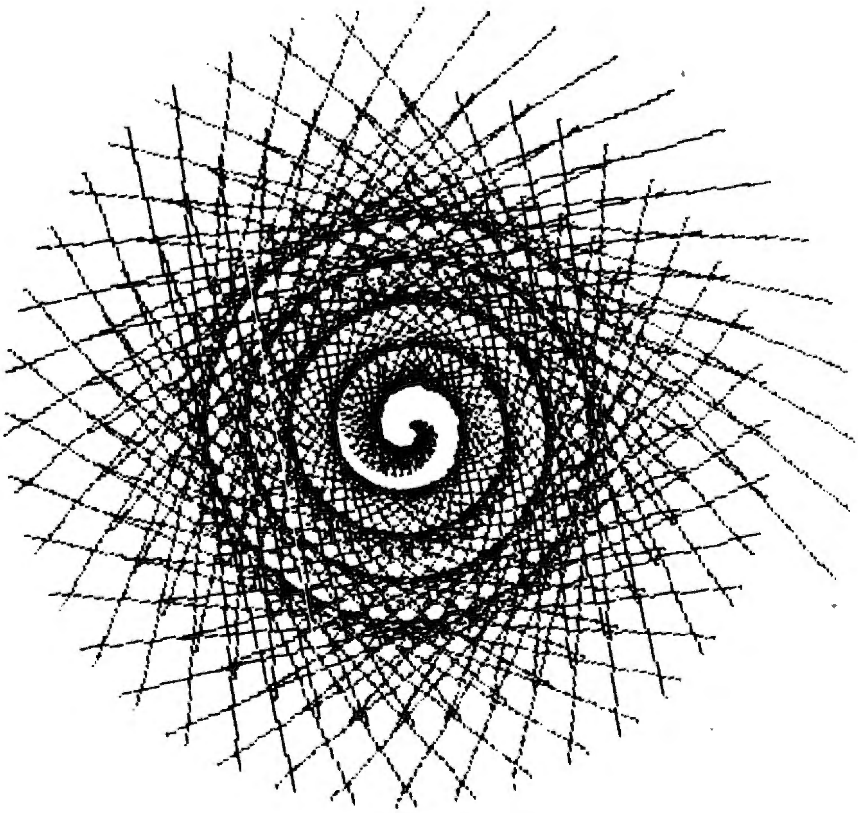


Figure 6: An overlapping spiral made of two hundred lines.

```
100 REM  DESIGN NO. 6
110 FOR T=0 TO 10*PI STEP PI/20
120 X1=COS(T)*5*T+202:Y1=SIN(T)*5*T+202
130 A=T+2*PI/3
140 X2=COS(A)*5*A+202:Y2=SIN(A)*5*A+202
150 GOSUB 1500
160 NEXT T
```


✓See List of Advertisers on page 306

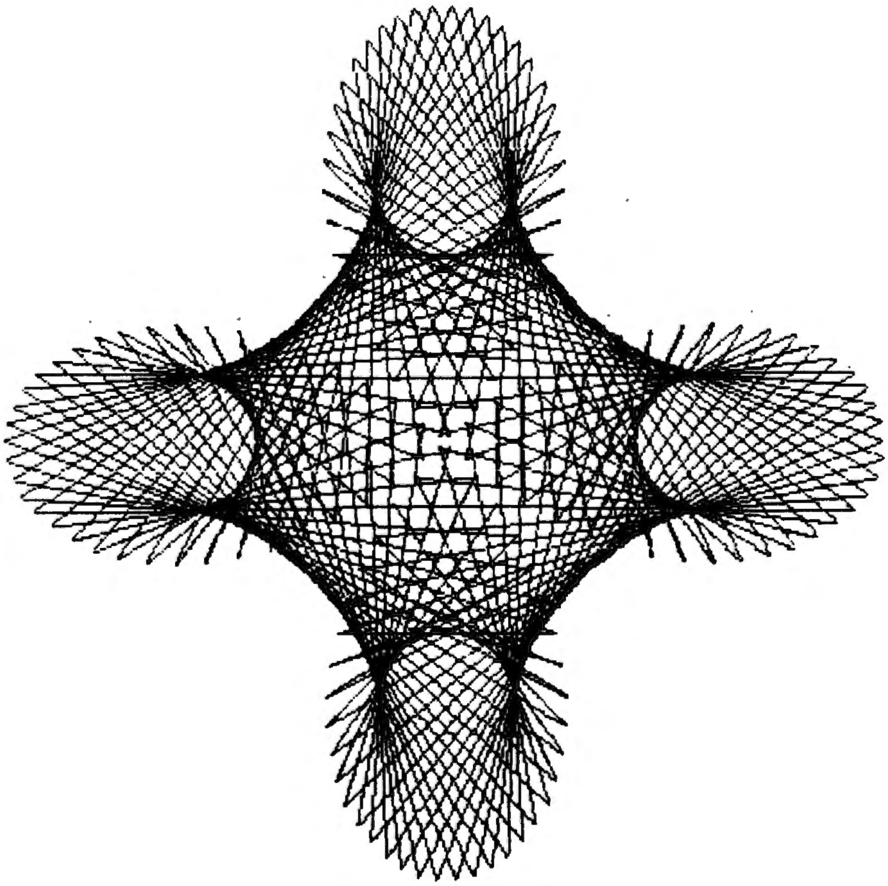


Figure 7: This figure used polar coordinates, and is based on a four leaf rose.

```
100 REM  DESIGN NO. 7
110 FORT=0TO2*PI  STEP PI/75
120 R=COS(2*T)*200
130 X1=COS(T)*R+202:Y1=SIN(T)*R+202
140 A=T+PI/3
150 R2=COS(2*A)*200
160 X2=COS(A)*R2+202:Y2=SIN(A)*R2+202
170 GOSUB 1500
180 NEXT T
```

the vertices and B is the Y coordinate. Lines 130-160 compute the position of all of the vertices. Since all of the vertices are on a circle, the X coordinate can be found by using the cosine of the angle theta, and the Y coordinate can be found with the sine.

A small number is subtracted from two π in line 130 to stop the angle two π from being used, because zero is the same angle. (Remember, the computer uses radians instead of degrees).

Line 170 combines all the vertices in all possible combinations. The values of the endpoints are then assigned to the variables, and the LINE routine is called. After this section of the program is completed, the program will ask whether or not to continue. If everything seems all right, press enter and pray that the CLOSE routine works.

Fig. 2 uses the same program as Fig. 1. Line 150 was changed to make the center design smaller. The center design has eleven vertices. The first portion of the program draws both corner designs. I originally planned to have the center design tangent to the insides of the corner designs, but the insides of the corner designs do not actually form circles as I expected.

Fig. 3 is a moire pattern created by the interference of regularly spaced radii of a circle. Each endpoint can be easily calculated by using the sine and cosine of the angle theta (T). The corners were not filled because they are almost entirely black. Change line 120 to "X1 = 0:Y1 = 0", and you will see the difference.

This type of pattern can be created with

```
100 REM  DESIGN NO. 8
110 FORT=0TO2*PI STEP PI/60
120 R=COS(2*T)*200
130 X1=COS(T)*R+202:Y1=SIN(T)*R+202
140 A=T+PI/2
150 R2=COS(2*A)*200
160 X2=COS(A)*R2+202:Y2=SIN(A)*R2+202
170 GOSUB 1500
180 NEXT T
```

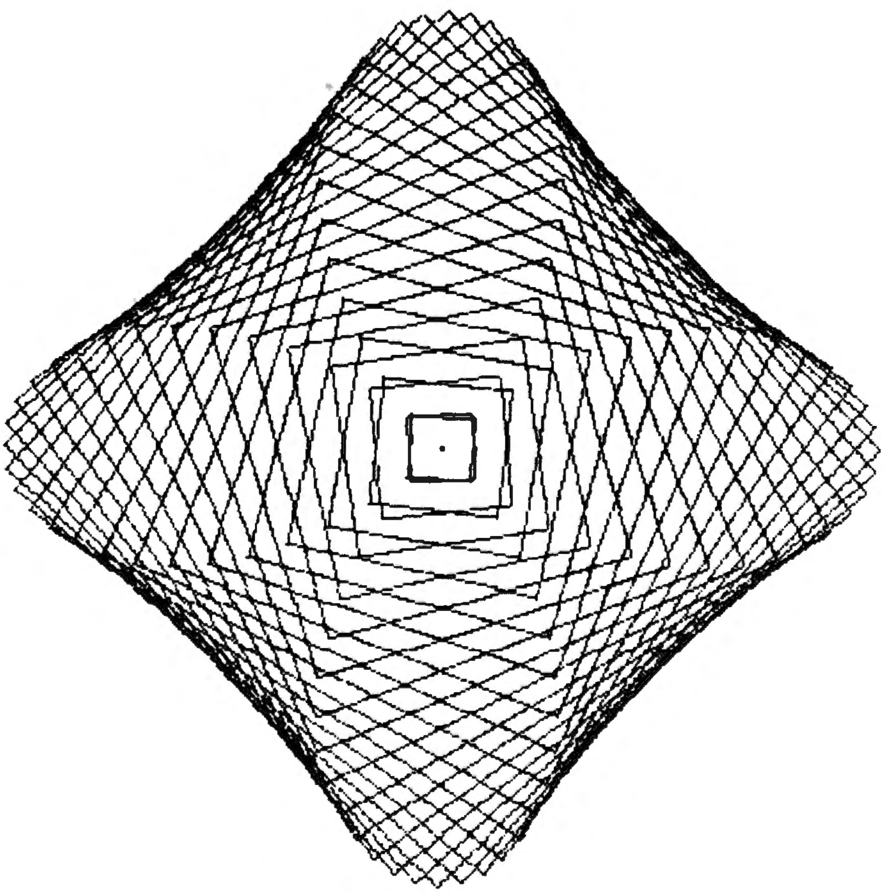


Figure 8: This figure is completely made of squares.

Applied Economic Analysis

Dr. David Chereb

A noted authority on microcomputing, Dr. Chereb has developed these software systems for TRS-80® users.

BUSINESS PLANNING & FORECASTING PACKAGE

This is the most powerful and comprehensive fore—casting system available for any microcomputer.

■ **COMPREHENSIVE**
This package can fully analyze all your important data. Whether you want to find the trend line for your sales and plot the results or build a simultaneous equation forecasting model, this system will do the job. This is a product you will grow into, not out of. The package contains several forecasting methods to fit your precise needs, and includes:

- Advanced Multiple Regression with Auto Correlation correction
- 10 Data Transformation options
- Disc Data Management System
- Adaptive Filtering option... And much more

While these are sophisticated techniques, this package has been designed to be easily used by the non technical businessman. Small businesses through large corporations have reaped the benefits of AEA's Business Planning & Forecasting Package (BPFP).

■ **VERSATILE**
With the "BPFP", you can create a data series, store it on a disk, and analyze the data by using the various system modules. The results of the analysis can then be listed as a table or plotted as a graph. As an option, results from the "BPFP" can be stored on disk for use as a VISICALC® File.

Applied Economic Analysis
P.O. Box 302
Allamuchy, NJ 07820
(201) 852-3573

■ **CONVENIENT**
To use the package, just load "BPFP" and you're off. Every part of the process has a menu to simplify responses. A second disk in the package, the Econdata File, covers over 30 vital topics, including the GNP, retail sales, money supply, and mortgage rates from the 1st quarter of 1955, through the 4th quarter of 1980. Users will also receive our User Notes for the "BPFP", which contain useful analytical examples.

■ **PACKAGE CONTENTS**
A sturdy three ring binder
Over 70 pages of instruction and documentation
2 mini diskettes with the programs on one disk and the Econdata File on the other

■ **HARDWARE REQUIREMENTS**
TRS-80® with at least 48 K of RAM for Mod I&III, 64 K of RAM for Mod II
One disk (two are preferred)
TRSDOS®

To order, check software desired:

☐ Business Planning & Forecasting	Model I&III 199.00	Model II 299.00
☐ Box-Jenkins Forecasting	Model I&III 199.00	Model II 299.00
☐ United States Simulation Model	Model I&III 299.00	Model II 399.00
☐ Risk Analysis	Model I&III 149.00	Model II 249.00
☐ Econdata File	Model I&III 29.00	Model II 39.00

47

Applied Economic Analysis

Make check or money order payable to:
Applied Economic Analysis
P.O. Box 302, Allamuchy, NJ 07820
(201) 852-3573

10% Discount when ordering 2 or more programs of any mix.
New Jersey residents add 5% sales tax.
Shipping and handling prepaid in the U.S.
For overseas airmail delivery, add $19.00 shipping charges.

Name (print) ____________________ Title ____________________
Company ____________________ Address ____________________
City ____________ State ________ Zip ________ Tele. No. ____________

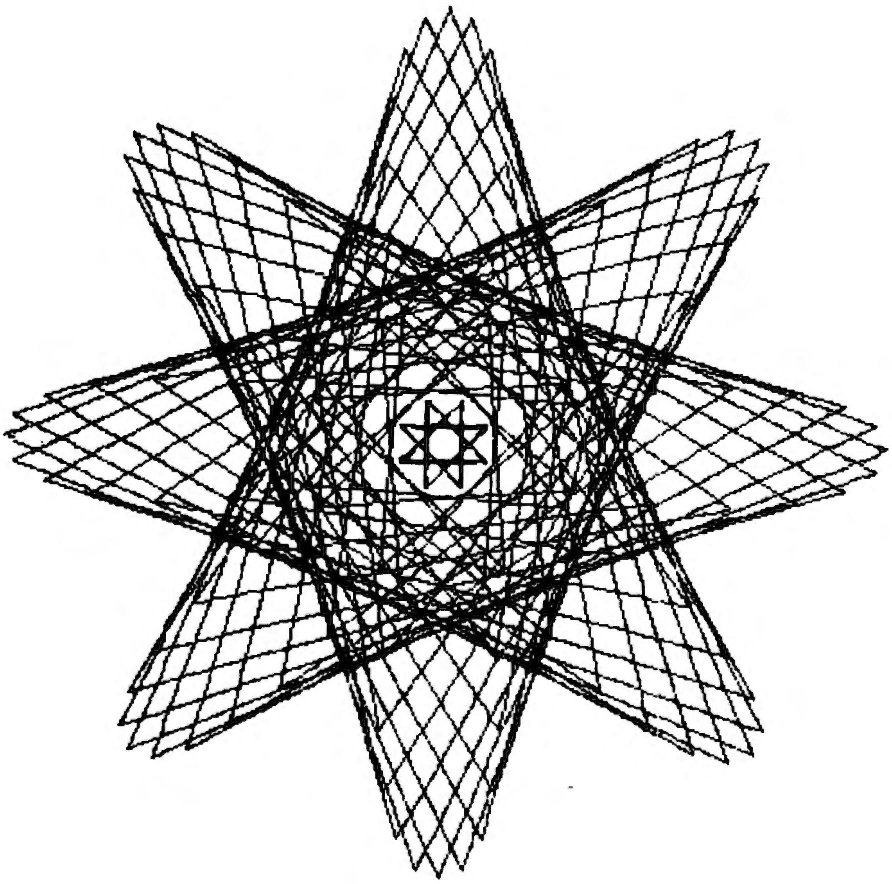


Figure 9: This is based on an eight leaf rose; the small star in the center was an accident.

```
100 REM  DESIGN NO. 9
110 FORT=0TO2*PI STEP PI/60
120 R=COS(4*T)*200
130 X1=COS(T)*R+202:Y1=SIN(T)*R+202
140 A=T+PI/4
150 R2=COS(4*A)*200
160 X2=COS(A)*R2+202:Y2=SIN(A)*R2+202
170 GOSUB 1500
180 NEXT T
```

curves as well as straight lines, as long as you begin with two sets of lines, such as the set from each of the circles in the example. These lines should be nearly parallel at some point.

Fig. 4 was one of my great failures. It was first done with a simpler program, but I forgot to save the program and had to write a longer one to reproduce it. My original intention was to produce a square spiral in which the distance between the lines increased as the length of the sides grew longer. I ended up with this design which I call "The Tunnel". This design sounds very odd when it is printing because of the regular spacing of the vertical lines.

Fig. 5 is a very simple, attractive, disk. It was created by taking 100 regularly spaced point on a circle and connecting them with points 3/4 π further around the cir-

"(This figure) is entirely made of equilateral triangles. The spiral shape comes from an Archimedes Spiral which is computed in line 120."

cle. By changing the number in line 130, the size of the internal circle can be controlled.

The remainder of the designs, including Fig. 5, all use mathematical formulae of

```
100 REM  DESIGN NO. 10
110 FORT=0TO2*PI STEP PI/30
120 R=T*31
130 X1=COS(T)*R+202:Y1=SIN(T)*R+202
140 A=T+2*PI/3
150 X2=COS(A)*R+202:Y2=SIN(A)*R+202
160 GOSUB 1500
170 B=T+4*PI/3
180 X1=COS(B)*R+202:Y1=SIN(B)*R+202
190 GOSUB 1500
200 X2=COS(T)*R+202:Y2=SIN(T)*R+202
210 GOSUB 1500
220 NEXT T
```

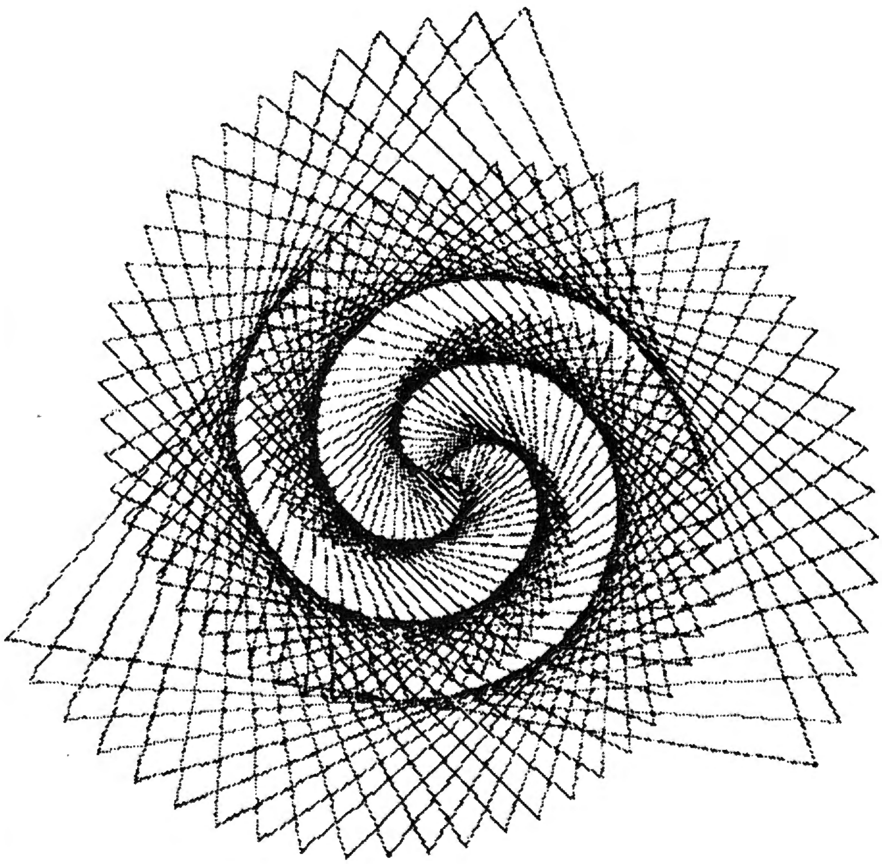


Figure 10: This is completely made of triangles.

See List of Advertisers on page 306

common polar coordinate shapes. Fig. 5 uses the conversion formulae from polar coordinates to Cartesian coordinates. The formulae are: X = Rcos(T), Y = Rsin(T). In Fig. 5, the radius, R, is 200. The design in polar coordinates would have its center at the pole or the origin in Cartesian coordinates. Just as the center of the video screen is not (0,0) but (63,23), the center of the design is (202,202). To translate the center from (0,0) to (202,202), 202 must be added to the conversion formulae. These formulae are now: X = 200•cos(T) + 202, Y = 200•sin(T) + 202.

Fig. 6 is an example of an Archimedes spiral. The formula in polar coordinates is simply R = T. R = 5T produces an identically shaped spiral, which widens more rapidly. Line 110 provides angles for five revolutions with forty points on each revolution. Line 120 puts the (X1,Y1) point on the spiral, and lines 130 and 140 put the (X2,Y2) point one-third of a revolution further around the spiral.

An unlimited number of spirals can be created by using different numbers in this program. One of my favorite designs is made up of points on one spiral connected to equivalent points on another spiral which rotates in the opposite direction. (The best way to create patterns is by trial and error. From more than a hundred different designs, these eleven are the best).

Fig. 7 was another accident. I really didn't know what I would get with this program. This design uses a shape known as the four leaf rose. Its formula in polar coordinates is R = cos(2T). An endless set of roses with different numbers of petals can be created by changing the number two in the equation. If the equation were R = cos(3T), a three leaf rose would be produced. R = cos(4T) will produce an eight leaf rose.

A very small change in a program using polar coordinates can cause a major change in the design. The only differences between Fig. 7 and Fig. 5 are a different step in line 110 and a different offset in line 140. Fig. 8 is another of my favorites because it is completely made up of squares. It was another accident.

Fig. 9 is also very similar to Fig. 8, using an eight leaf rose rather than the four leaf rose. This is the fourth of four accidents. The small center star makes this design an amazing design instead just average.

The relationship between the step in line 110 and the offset in line 140 determines whether the ends of lines will meet. In Fig. 9,

Continue to 142

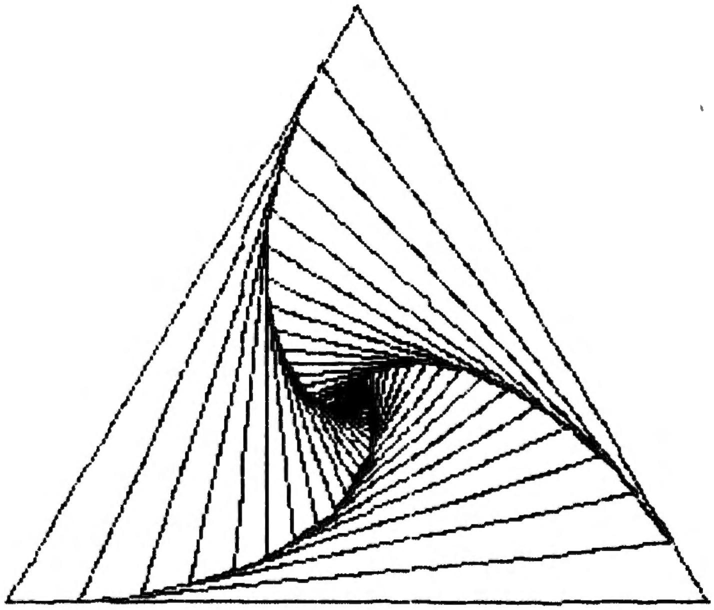


Figure 11: This design is made of triangles.

```
100 REM  DESIGN NO. 11
105 R=1
110 FORT=0TO3.24 STEP PI/30  :REM MAKES 32 TRIANGLES
120 R=R*1.17557
130 X1=COS(T)*R+202:Y1=SIN(T)*R+202
140 A=T+2*PI/3
150 X2=COS(A)*R+202:Y2=SIN(A)*R+202
160 GOSUB 1500
170 B=T+4*PI/3
180 X1=COS(B)*R+202:Y1=SIN(B)*R+202
190 GOSUB 1500
200 X2=COS(T)*R+202:Y2=SIN(T)*R+202
210 GOSUB 1500
220 NEXT T
```

Program Listing 1. BASIC Error Checking Routine

```
10 REM  PROGRAM NO. 1    BASIC ERROR CHECKING AND CALLI
     NG
20 OPEN
30 PI=3.1416
100 REM  DESIGN NO. 1
110 INPUT"NUMBER OF POINTS (SAMPLE HAS 18) ";N
120 DIM A(N),B(N)
130 FORT=0TO2*PI-.001 STEP 2*PI/N
140 Z=Z+1
150 A(Z)=COS(T)*200+202:B(Z)=SIN(T)*200+202
160 NEXT T
```

Fastload

FOR TRS-80* MODEL I USERS ONLY

**16 Times
Normal Speed**

*TRS-80 is a trademark of Tandy Corp.

- High speed load TRS-80* Level II cassettes
- Input 15K byte Level II program in 15 seconds
- Search BASIC or SYSTEM programs by name

Unlike other high speed tape input devices, FASTLOAD uses standard format cassettes. Therefore, there is no need to re-record on other media. At 8000 baud, FASTLOAD is faster than disk for short programs. FASTLOAD reads tapes at the fast-forward speed of the CTR-41 cassette recorder. The recorder can also be used for CSAVE at the normal speed.

FASTLOAD connects to the 40 pin I/O or to the Expansion box. The control program does not use computer memory because it is in a built-in PROM. Other valuable features are keyboard debounce program, automatic key repeat routine and key-beep via cassette speaker. Price is $188.00 for FASTLOAD and $95.00 for the modified CTR-41 recorder.

✓422

Personal Micro Computers Inc.

475 Ellis Street, Mountain View, CA 94043 (415) 968-1604

✓See List of Advertisers on page 306

Throw It Away

(your dictionary, that is)

Now you can proofread and correct ten pages of text in less than one minute, with **MICROPROOF** dictionary software

- **EASY TO USE:** Prepare your text on any Z-80 based microcomputer, using any of a number of popular word processing programs. When you are finished, enter the appropriate command, and MICROPROOF proofreads your document, displaying misspellings and typos on the screen. Then MICROPROOF displays each error separately, requesting you to enter the correct spelling for each. Finally, MICROPROOF corrects your document. All in less than a minute.
- **COMPREHENSIVE:** MICROPROOF comes with a 50,000 word vocabulary. That's equivalent to a Webster's Pocket Dictionary. And MICROPROOF's dictionary is INFINITELY EXPANDABLE. You can add your own technical words and jargon.
- **RECOGNIZES** prefixes, suffixes, hyphenation and comment lines.
- **REQUIRES** ONLY 32K of memory and one single or double density 5¼ inch or 8 inch disk drive.
- **AFFORDABLE:** Available in three forms: fully independent program to identify errors, independent program to identify and correct errors, or with a conversion program that will permit MICROPROOF to operate from within your own word processing software.

PRICES

TRS-80® Model I version	$125.00	Correcting Feature	$60.00
TRS-80® Model II version	$200.00	Word Processing Conversion	
CP/M® version	$200.00	**Scripsit® or Electric Pencil®**	$35.00
APPLE® version	available soon	Separate User's Manual	$5.00

Master Charge and Visa accepted. No COD please. California residents add 6% sales tax.
Send for a brochure—specify type of microcomputer and word processing software used. Dealer inquiries welcome.

CORNUCOPIA SOFTWARE ✓378
Post Office Box 5028
Walnut Creek, California 94596
(415) 524-8098

"An unlimited number of spirals can be created by using different numbers in this program."

```
170 FORS=1TON-1:FORD=S+1TON
180 X1=A(S):Y1=B(S)
190 X2=A(D):Y2=B(D)
270 GOSUB1500
280 NEXT D,S
1000 INPUT"CONTINUE";A
1010 CLOSE
1020 END
1500 REM LINE ROUTINE
1510 PRINT"<";
1520 IF X1 < 0 OR X1 > 405 THEN PRINT"X1 ILLEGAL":END
1530 IF X2 < 0 OR X2 > 405 THEN PRINT"X2 ILLEGAL":END
1540 IF Y1 < 0 OR Y1 > 405 THEN PRINT"Y1 ILLEGAL":END
1550 IF Y2 < 0 OR Y2 > 405 THEN PRINT"Y2 ILLEGAL":END
1560 IF INT(X1)=INT(X2) AND INT(Y1)=INT(Y2) THENPRINT"V
     ALUE TOO CLOSE":RETURN
1600 X3=INT(X1/256):X4=INT(X2/256)
1610 Y3=INT(Y1/256):Y4=INT(Y2/256)
1620 POKE25241,X1-X3*256:POKE25242,X3
1630 POKE25245,Y1-Y3*256:POKE25246,Y3
1640 POKE25249,X2-X4*256:POKE25250,X4
1650 POKE25253,Y2-Y4*256:POKE25254,Y4
1660 LINE
1670 PRINT"> ";
1680 RETURN
```

Program Listing 2. Line Drawing Routine

```
                ;LINE DRAWING ROUTINE FOR BASE 2 PRINTER AND 32K LEVEL II
                ;BOB BOOTHE 7-8-80
41A3                    ORG     16804-1         ;LINE LOCATION
41A3 C3AF62             JP      LINE            ; AUTOMATICALLY ENABLE
4179                    ORG     16762-1
4179 C39563             JP      OPEN
4185                    ORG     16774-1
4185 C3B063             JP      CLOSE
6297                    ORG     25239           ;SO END IS BELOW 25603
6297 0000       STARTX  DEFW    0       ;RESERVE 4 BYTES FOR EACH VARIABLE
6299 0000               DEFW    0
629B 0000       STARTY  DEFW    0
629D 0000               DEFW    0
629F 0000       ENDX    DEFW    0
62A1 0000               DEFW    0
62A3 0000       ENDY    DEFW    0
62A5 0000               DEFW    0
62A7 0000       DIRX    DEFW    0        ;LEAST SIGNIFICANT
62A9 0000               DEFW    0        ;MOST SIGNIFICANT
62AB 0000       DIRY    DEFW    0
62AD 0000               DEFW    0
62AF D9         LINE    EXX                     ; SAVE REGISTERS
62B0 2AA162             LD      HL,(ENDX+2)     ; GET END OF X POINT
62B3 ED5B9962           LD      DE,(STARTX+2)   ; START OF X
62B7 B7                 OR      A               ; RESET CARRY FLAG
62B8 ED52               SBC     HL,DE           ; FIND DIFFERENCE
62BA 22A762             LD      (DIRX),HL
62BD CB7C               BIT     7,H             ; IS THE RESULT NEGATIVE
62BF 2805               JR      Z,DX0
62C1 21FFFF             LD      HL,-1           ; YES, THEN MAKE MSW NEG
62C4 1803               JR      PDX
62C6 210000     DX0     LD      HL,0            ; NO, MAKE MSW 0
62C9 22A962     PDX     LD      (DIRX+2),HL
62CC 2AA562             LD      HL,(ENDY+2)     ; REPEAT ABOVE FOR Y
62CF ED5B9D62           LD      DE,(STARTY+2)
62D3 B7                 OR      A
62D4 ED52               SBC     HL,DE
62D6 22AB62             LD      (DIRY),HL
62D9 CB7C               BIT     7,H
62DB 2805               JR      Z,DY0
62DD 21FFFF             LD      HL,-1
62E0 1803               JR      PDY
62E2 210000     DY0     LD      HL,0
62E5 22AD62     PDY     LD      (DIRY+2),HL
62E8 2AA762             LD      HL,(DIRX)
62EB ED5BAB62           LD      DE,(DIRY)
62EF 7C                 LD      A,H
62F0 E680               AND     80H             ; PUT BIT 7 OF H IN B
```


✓See List of Advertisers on page 306

DOES YOUR SMART PRINTER SUDDENLY BECOME DUMB WHEN YOU PRINT FROM SCRIPSIT?

Is your printer capable of underlining
but not from Scripsit?

Is your printer capable of **Bold Printing**
but not from Scripsit?

Can your printer super script and sub script
but not from Scripsit?

Can your printer change pitches
but not from Scripsit?

The answer is SCRIPMOD

SCRIPMOD does not require a separate printer driver. With SCRIPMOD control codes can be embedded in the test of your document. You use the same format line syntax you're used to now. SCRIPMOD adds one format instruction to Scripsit & two control codes.

>C=N J=N **CC=H**

The above line tells SCRIPMOD to justify text and to backspace the printer upon encountering a Text Character in the document. The Text Character is entered by placing the cursor at the place in text where the control code is to take place and typing on **@T**. Of course your printer must be capable of backspacing and must use the ASCII code 8 (or H) as the control for backspace. If the code is something different there is no problem as the correct code may always be sent. Any control code your printer is capable of using from 1 to 31 with or without an **ESCAPE** lead in many be sent.

The second control code which is added in the **MENU** command. You press an **@M** and the screen clears and prompts you to select a drive from 0 through 3 or return to the text. All visible files can be displayed on the screen at this time. When you select to return to the text, the cursor is placed on the exact character it was on when you selected **@M**.

The minimum system required for SCRIPMOD is the Mod I 32K disk system with either the RS lower case mod or the EP lower case mod.

SCRIPMOD is supplied on disk with full documentation for **$39.95.**

PENCIL-FIX Save your warranty. Use PENCIL-FIX to avoid custom control key on your keyboard. Redefines the control key for EP to be the @ key. Use RS lower case mod or the EP mod without the control key. Disk based EP only . . . **$14.95**

PRINT-CENTRAL Send any control code directly to your smart printer from the BASIC command mode or from DOS. Avoid having to type such things as "LPRINT CHR$(31) just to change pitch. Use CLEAR right arrow (two key strokes) instead. Any code from 1 to 31 may be sent. Mod I 32K disk . . . **$24.95**

SPOOL-REL An in-memory print buffer that runs in Mod I 32K or 48K disk systems. Fully relocatable code and buffer. A true background spooler at an unbelievably low price . . . **$24.95**

TIGGER-GRAF Create engineering, scientific, business, or just plain fun on your IDS 440G or 460G printer. Resolution is 495 x 575. Easy BASIC programs provided for data entry and machine language module for speed. Includes setting individual points, drawing lines, shading shapes. Several graphs may be catenated along the Y-axis for larger graphs. Requires Mod I 32 or 48K 1 disk . . . **$149.95**

VERN STREET PRODUCTS ✓111

THE COMPUTER STORE 4949 S. Peoria • Tulsa, OK 74105 • 918-747-2550

ACCEL/ACCEL2 SPEEDUPS

TRS-80 Model I BASIC Compilers

Table below shows the BASIC subset translated by **ACCEL** and **ACCEL2** to machine code. Figures represent the minimum expected ratio of execution times, compiler to interpreter. All other BASIC statements and functions run at interpreter speed after compilation.

	INTEGER	SINGLE	DOUBLE	STRING
Assignment (LET)	**115**	3.3	3.4	7.6
Array Reference (1-dim)	**35**	78	66	34.5
AND or OR	**41**	2.5	2.0	
Compare (<,etc)	**30**	1.6	1.4	4.2
Add, Subtract, Concat	**47**	2.0	1.5	4.9
Multiply (*)	3.3	2.0	1.5	
Divide (/)	2.0	2.0	1.02	
Reference to a constant	**69**	65	54	2.1
FOR with NEXT	**15**			
POKE	**82**	4.6	3.6	
SET or RESET	**6.7**	3.1	2.6	
IF THEN ELSE	**11.1**	3.0	2.3	7.6
ON expression GOTO	**15.8**	3.2	2.8	
Functions				
VARPTR	**33**	47	47	44
USR	**11.2**	3.7	2.8	
POINT	**6.9**	3.0	2.5	
PEEK	**52**	4.4	3.5	
LEN				43
MID$				4.1
LEFT$				3.0
RIGHT$				2.8
CHR$				4.7
ASC				30
CVI				28
Flow of Control				
GOSUB with RETURN	**137**			
COTO	**204**			
All other BASIC statements and functions	**1.0**	**1.0**	**1.0**	**1.0**

ACCEL: For 16K TRS-80 Model I. Compiles boldface subset in INTEGER variable type. Compile-time size 2816 bytes, run-time size 256 bytes. Trade up later to ACCEL2 for the price difference. $44.95 + .75 shipping

ACCEL2: For 32K TRS-80 Model I. Compiles subset in all variable types. Compile-time size 5120 bytes, run-time size 1024 bytes. Local and Global options help control growth of the compiled code. Output save to Disk, Stringy/Floppy. TRSDOS and most vendor DOSs supported. Use ACCEL2 during BASIC program development for easy huge speedups. $88.95 + 1.50 shipping

TSAVE: Writes ACCEL/ACCEL2 compiler output to independent SYSTEM tape. $9.95

Developed in Britain
by Southern Software

ALLEN GELDER SOFTWARE ✓79
Box 11721 Main Post Office
San Francisco, CA 94101

Mastercharge/Visa
Ca add 6%

TRS-80, TRSDOS tm Radio Shack
Stringy/Floppy tm exatron, inc.

CONVERT YOUR TRS-80 MODEL-I INTO A

DEVELOPMENT SYSTEM

Now you can develop Z-80 based, *stand-alone* devices such as games, robots, instruments and peripheral controllers, by using your TRS-80 as a development system. The DEVELOPMATE plugs into the expansion connector of your TRS-80 and adds **PROM PROGRAMMING** and **IN-CIRCUIT-EMULATION** capabilities to your system (with or without expansion interface).

Complete instructions and sample schematics are included to help you design your own simple stand-alone microcomputer systems. THESE SYSTEMS CAN BE AS SIMPLE AS FOUR ICs: one TTL circuit for clock and reset, a Z-80, an EPROM, and one peripheral interface chip.

When the In-Circuit-Emulation cable is plugged into the Z-80 socket of your stand-alone system, the system becomes a part of your TRS-80: You can use the full power of your editor/assembler's debug and trace programs to check out both the *hardware and the software*. Simple test loops can be used to check out the hardware, then the system program can be run to debug the logic of your stand-alone device.

Since the program is kept in TRS-80 RAM, changes can be made quickly and easily. When your stand-alone device works as desired, you use the Developmate's PROM PROGRAMMER to copy the program into a PROM. With this PROM, and a Z-80 in place of the emulation cable, your stand-alone device will work by itself.

The DEVELOPMATE is extremely compact: Both the PROM programmer and the In-Circuit-Emulator are in one small plastic box only 3.2"×5.4". A line-plug mounted power supply is included. The PROM programmer has a "personality module" which defines the voltages and connections of the PROM so that future devices can be accommodated. However, the system comes with a "universal" personality module which handles 2758,2508(8K), 2716,2516(16K), 2532(32K), as well as the new electrically alterable 2816 and 48016(16K EEPROMs).

The COMPLETE DEVELOPMATE 81, with software, power supply, emulation cable, TRS-80 cable, and "universal" personality module, is ONLY $329!
The PROM PROGRAMMER is available *separately* for ONLY $239.

VISA

ORION INSTRUMENTS ✓376
172 Otis Avenue, Dept. M, Woodside, CA 94062
(415) 851-1172
Master Charge and Visa phone orders accepted.
California residents please add 6% sales tax.

master charge

```
62F2 47                     LD     B,A
62F3 7A                     LD     A,D
62F4 E680                   AND    80H             ; PUT BIT 7 OF D IN C
62F6 4F                     LD     C,A
62F7 CB25          SHIFT    SLA    L               ; EXPAND DIRECTION UNTIL
62F9 CB14                   RL     H               ; BIT 7 CHANGES
62FB CB23                   SLA    E               ; THIS GIVES MAX. SPEED
62FD CB12                   RL     D
62FF 7C                     LD     A,H
6300 E680                   AND    80H
6302 B8                     CP     B
6303 2008                   JR     NZ,DSHIFT
6305 7A                     LD     A,D
6306 E680                   AND    80H
6308 B9                     CP     C
6309 2002                   JR     NZ,DSHIFT
630B 18EA                   JR     SHIFT
630D 22A762        DSHIFT   LD     (DIRX),HL       ; DONE SHIFTING SO STORE
6310 ED53AB62               LD     (DIRY),DE
6314 CD6163        NXTBLO   CALL   SET             ; SET 1ST POINT
6317 AF                     XOR    A               ; RESET CARRY FLAG
6318 2AA162                 LD     HL,(ENDX+2)
631B ED5B9962               LD     DE,(STARTX+2)
631F ED52                   SBC    HL,DE           ; DOES START = END YET
6321 2830                   JR     Z,MAYBE
6323 2A9762        NOTYET   LD     HL,(STARTX)     ; ADD DIRECTION TO START
6326 ED5BA762               LD     DE,(DIRX)
632A 19                     ADD    HL,DE           ; ADD LSW'S FIRST
632B 229762                 LD     (STARTX),HL
632E 2A9962                 LD     HL,(STARTX+2)
6331 ED5BA962               LD     DE,(DIRX+2)     ; (DIRX+2) SHOULD BE 0
6335 ED5A                   ADC    HL,DE           ; ADD CARRY FROM LSW'S
6337 229962                 LD     (STARTX+2),HL   ; REPEAT FOR Y'S
633A 2A9B62                 LD     HL,(STARTY)
633D ED5BAB62               LD     DE,(DIRY)
6341 19                     ADD    HL,DE
6342 229B62                 LD     (STARTY),HL
6345 2A9D62                 LD     HL,(STARTY+2)
6348 ED5BAD62               LD     DE,(DIRY+2)
634C ED5A                   ADC    HL,DE
634E 229D62                 LD     (STARTY+2),HL
6351 18C1                   JR     NXTBLO
6353 B7            MAYBE    OR     A               ; ARE Y'S ALSO EQUAL
6354 2AA562                 LD     HL,(ENDY+2)
6357 ED5B9D62               LD     DE,(STARTY+2)
635B ED52                   SBC    HL,DE
635D 20C4                   JR     NZ,NOTYET       ; NO, GO BACK AND FINISH
635F D9                     EXX                    ; GET BACK OLD REGISTERS
6360 C9                     RET                    ; RETURN TO PROGRAM
6361 0600          SET      LD     B,0             ; ROUTINE TO SET A POINT
6363 2A9D62                 LD     HL,(STARTY+2)
6366 110700                 LD     DE,7            ; GOING TO DIVIDE BY 7
6369 B7                     OR     A               ; RESET CARRY FLAG
636A ED52          DIV7     SBC    HL,DE           ; DIVISION BY SUBTRATION
636C 3803                   JR     C,DONDIV
636E 04                     INC    B               ; COUNT THE 7'S
636F 18F9                   JR     DIV7
6371 19            DONDIV   ADD    HL,DE           ; PUT BACK LAST 7
6372 4D                     LD     C,L             ; PUT REMAINDER IN C
6373 04                     INC    B               ; SO B ISN'T 0
6374 119601                 LD     DE,406          ; 406 BYTES PER LINE
6377 216D62                 LD     HL,25603-406    ; COMPENSATE FOR (INC B)
637A 19            MULT     ADD    HL,DE           ; ADD 406 FOR EACH LINE
637B 10FD                   DJNZ   MULT
637D ED5B9962               LD     DE,(STARTX+2)   ; GET X COLUMN
6381 19                     ADD    HL,DE           ; MAKE ADDRESS
6382 41                     LD     B,C
6383 04                     INC    B               ; SO B DOESN'T EQUAL 0
6384 118D63                 LD     DE,TABLE-1      ; COMPENSATE FOR (INC B)
6387 13            FINDT    INC    DE
6388 10FD                   DJNZ   FINDT           ; FIND LOCATION IN TABLE
638A 1A                     LD     A,(DE)
638B B6                     OR     (HL)            ; SET THAT BIT
638C 77                     LD     (HL),A          ; STORE
638D C9                     RET                    ; FINALLY DONE
638E 01            TABLE    DEFB   01H             ; BIT 0
638F 02                     DEFB   02H             ; BIT 1
6390 04                     DEFB   04H             ; SO ON
6391 08                     DEFB   08H
6392 10                     DEFB   10H
6393 20                     DEFB   20H
6394 40                     DEFB   40H
```

"This design took more work than any other design because the numbers had to be precisely calculated . . ."

```
6395 D9       01400 OPEN   EXX                  ; SAVE REGISTERS
6396 210364   01410        LD     HL,25603      ; START PICTURE
6399 110464   01420        LD     DE,25604      ; START +1
639C 01FB5B   01430        LD     BC,23547      ; LENGTH
639F 3680     01440        LD     (HL),128      ; BIT 7 ON, REST OFF
63A1 EDB0     01450        LDIR                 ; COPY ALL THE WAY
63A3 D9       01460        EXX                  ; GIVE 'EM OLD REGISTERS
63A4 C9       01470        RET                  ; GET BACK TO FUN STUFF
63A5 E5       01480 PRTOUT PUSH   HL            ; SAVE HL
63A6 21E837   01490        LD     HL,37E8H      ; LP POINTER
63A9 CB7E     01500 PRTLP8 BIT    7,(HL)        ; BIT 7 ON MEANS BUSY
63AB 20FC     01510        JR     NZ,PRTLP8
63AD 77       01520        LD     (HL),A        ; LP READY, SO OUTPUT
63AE E1       01530        POP    HL            ; GET BACK HL
63AF C9       01540        RET                  ; BACK TO WORK
63B0 D9       01550 CLOSE  EXX                  ; YOU KNOW WHAT THIS DOES
63B1 3E1B     01560        LD     A,27          ; ESCAPE
63B3 CDA563   01570        CALL   PRTOUT
63B6 3E32     01580        LD     A,50          ; 96 CHARACTERS PER LINE
63B8 CDA563   01590        CALL   PRTOUT
63BB 3E1B     01600        LD     A,27          ; ESCAPE
63BD CDA563   01610        CALL   PRTOUT
63C0 3E62     01620        LD     A,98          ; SET VERTICAL LINE
63C2 CDA563   01630        CALL   PRTOUT
63C5 3E0E     01640        LD     A,14          ; SPACING TO 14 HALF DOTS
63C7 CDA563   01650        CALL   PRTOUT
63CA 063A     01660        LD     B,58          ; NUMBER OF LINES
63CC 210364   01670        LD     HL,25603      ; FIRST BYTE LOCATION
63CF C5       01680 LOOP   PUSH   BC            ; SAVE LINE COUNTER
63D0 3E1B     01690        LD     A,27          ; ESCAPE
63D2 CDA563   01700        CALL   PRTOUT
63D5 3E63     01710        LD     A,99          ; TRANSMIT GRAPHICAL DATA
63D7 CDA563   01720        CALL   PRTOUT
63DA 0655     01730        LD     B,85          ;NUMBER OF BLANKS
63DC 3E80     01740 BLANK1 LD     A,128         ; IT IGNORES BIT 7
63DE CDA563   01750        CALL   PRTOUT
63E1 10F9     01760        DJNZ   BLANK1
63E3 019601   01770        LD     BC,406        ;NO. CHARS FROM MEMORY
63E6 7E       01780 CHAR   LD     A,(HL)
63E7 CDA563   01790        CALL   PRTOUT
63EA 23       01800        INC    HL            ; INC ADDRESS AFTER EACH
63EB 0B       01810        DEC    BC            ; DEC COUNTER AND CHECK
63EC 79       01820        LD     A,C
63ED B0       01830        OR     B
63EE 20F6     01840        JR     NZ,CHAR
63F0 0655     01850        LD     B,85          ; MORE BLANKS AT END
63F2 3E80     01860 BLANK2 LD     A,128
63F4 CDA563   01870        CALL   PRTOUT
63F7 10F9     01880        DJNZ   BLANK2        ; THAT FINISHES ONE LINE
63F9 3E0D     01890        LD     A,13          ; CAUSES LINE FEED
63FB CDA563   01900        CALL   PRTOUT
63FE C1       01910        POP    BC            ; GET LINE COUNTER
63FF 10CE     01920        DJNZ   LOOP
6401 D9       01930        EXX
6402 C9       01940        RET         ;ADDRESS SHOULD BE 6402H WHEN ASSEMBLING
1A19          01950        END    1A19H         ;ENTRY TO LEVEL II BASIC
00000 TOTAL ERRORS
```

60/4 = 15. Because 15 is an integer, the ends will meet. However, in Fig. 5, 50*(3/4) = 37.5. Since 37.5 is not an integer, the ends will not meet.

Fig. 10 is made up entirely of equilateral triangles. The spiral shape comes from an Archimedes spiral which is computed in line 120. The first corner is computed in line 130, and the other corners are computed by adding an angle to it without changing the radius. All previous programs calculated a new radius whenever the angle was changed.

Fig. 11 is the same as Fig. 10 except that it is based on a logarithmic spiral instead of an Archimedes spiral. This design took more work than any other design because the numbers had to be precisely calculated so that each triangle would just touch but not overlap the previous one.

Some very good designs can be produced by combining several triangles. I have a more complicated program which draws six of these triangles together to form a fascinating hexagonal pattern.

A simple exercise in graphics is to make Fig. 1 in the shape of an elipse instead of a circle. A slightly more difficult problem is to make Fig. 10 rotate in the opposite direction. A very tricky problem is to make a design similar to Fig. 8 out of triangles or even hexagons. I will leave the rest to you, your imagination and luck. ■

Next month: the third dimension.

Bob Boothe is 18 years old and will graduate from Rolling Hills High School in June. He presently plans to enter the University of California at San Diego (the home of Pascal), in September, to major in Computer Science.

See List of Advertisers on page 306

SPRING SALE

MICROCOMPUTER TECHNOLOGY INCORPORATED

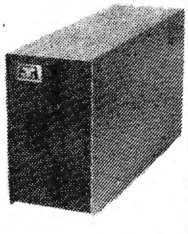
CLEARANCE SPECIAL
SHUGART SA400 (TF-3)

35 track disk drive
includes power supply
and chassis.
LIMITED SUPPLY

$299

FREE FULL YEAR WARRANTY

with purchase of new
SHUGART SA400 (TF-3L)
comes complete with
power supply, chassis,
cable & NEW DOS 80
ONLY **$439**



THE MICROCONDUCTOR™
the ultimate data base manager for your TRS-80®

The MICROCONDUCTOR™ is not just a file manager but a true Data Base Management System suitable for both the novice and professional users.

DATA FILES—No limit on the number of records a file can have.
FIELDS—Any type (string, interger, single, double). Eight entry modes (including defaults, counting, and suppress).
REPORTS—Four ways to generate reports. Total numeric column(s). Print on any paper in any format (statements, labels, etc.).
SORT—Any field(s) (i.e. multiple-key sort). Any size file, numeric or ASCII. Ascending or descending.
MAINT.—Command anticipation. Record duplication. Direct access and sequential search.
UPDATING/MERGING—Add, subtract, multiply, divide fields. Combine results from previous calculations. Test for any condition and take action.

TRS-80® MODEL I	$159
TRS-80® MODEL II	$399

NEW FOR TRS-80®

TF-8 80 TRACK DISK DRIVE by MPI
Double your capacity. Single head mini floppy. More than 200K bytes of storage. Complete with power supply and chassis.
$619

TF-9 DUAL 80 TRACK DISK DRIVE by MPI
Quadruple your capacity. 400K bytes of storage (like having 4-40 track drives in one unit). Complete with power supply and chassis.
$769

MOD II 8" Disk System

- 1 Drive System $949
- 2 Drive Expansion System $1,445

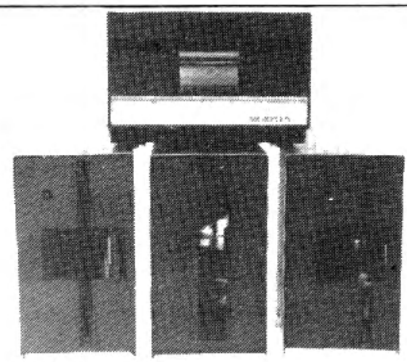
Disk Drive Sale!
with FREE power supply and chassis.

Shugart SA400L, 40 trk.	$329
Pertec FD200, 40 trk.	$379
TF-5 MPI B51, 40 trk.	$349
TF-7 Mircopolis, 77 trk.	$559
TDH-1 Dual sided drive, 35 trk.	$449
TF-3M Drive sys. 2 Shugart 40 trk.	$649

SOFTWARE SALE

NEWDOS 80	$149		
NEWDOS+ 40 trk.	$110	30 trk.	$99
AJA Business Pkg.	$359		
The Source	$100	Basic Compiler	$195

MONTHLY SPECIAL
Shugart SA801 Bare Drive **$459**



MOD I Disk Expansion Systems

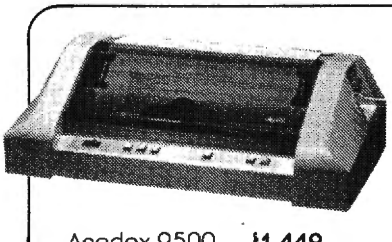
- 2 Shugart SA400 TF-3 .... $658
- 1 Two-Drive Cable $26
- 1 Expansion Interface 32K $439
- 1 DOS+ $99

TOTAL LIST PRICE $1,222
SPECIAL PRICE ONLY **$1,077**

★ BARE DRIVES FOR ANY MICROCOMPUTER

Pertec FD200	$282	FD250	$359
Shugart SA400 35 track	$269	SA400L 40 track	$279
MPI B51	$279	B52	$349
MPI B91	$399	B92	$525

Memory Kit (16K)	$43.00
AC Isolator (6 socket)	$49.95
Disk Head Cleaner	$19.95
Diskettes Verbatim (10) hard ring	$30.00



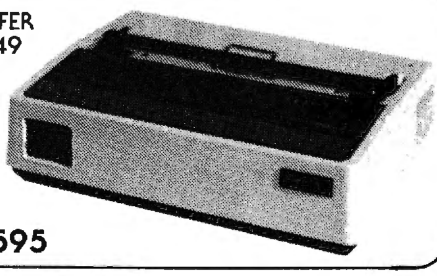
Anadex 9500 .... $1,449

Printers

Microline 82	$849
Microline 83	$1,060
Centronics 779	$1,069
Centronics 700	$1,129
Base 2	$649
Centronics 737	$839
Centronics 702-9	$1,995
Malibu	$2,093
Paper Tiger	$895
Daisy Wheel	$1,799

INTRODUCTORY OFFER
SAVE $300 LIST $949

Okidata Microline 80
$599



EPSON PRINTER $595

® Registered trademark of Radio Shack

TRS-80® MODEL III OWNERS

We think our disk system for expanding your capacity is just as good as theirs - and we know it costs much less . . .

COMPLETE SYSTEMS AND COMPONENTS*

TRS-80® MOD III SYSTEM

with 2 disk drives, 48K*

SAVE $500 OFF LIST PRICE

$1,899

16K RAM, Model III Basic $899

MTI DISK DRIVES for MOD III

Internal Kits

Disk Drive 1 .... $649

Disk Drive 2 .... $279

External Add-on Kits

Disk Drive 3 .... $379

Disk Drive 4 .... $359

Model III DOS & Manual $21.95

16K RAM Kit $59

RS-232 Serial Interface $95

New low cost internal disk kit available to expand your storage capability. With this kit, now, you may expand your Model III computer up to four MTI 40 track disk drives, giving 175,000 bytes of storage per drive for a total of 700K.

The kit includes one 40 track disk drive, controller, power supply, cables, mounting hardware and installation documentation.

Manufacturer Direct Price ONLY.................. $649

MICROCOMPUTER TECHNOLOGY, INC.

3304 W. MACARTHUR, SANTA ANA, CA 92704

★ PHONE (714) 979-9923 ★ TELEX #678401TAB IRIN

*Uses MTI Memory, Disk Drives & Components

® TRS-80® IS A REGISTERED TRADEMARK OF TANDY CORPORATION

M3-B

MasterCard VISA

ALL PRICES CASH DISCOUNTED
FREIGHT FOB FACTORY
ASK FOR FREE CATALOG

See List of Advertisers on page 306

Here's a RAM/ROM tutorial that even includes memory test routines.

Memories Are Made of This

Robert D. Randall
841 So. 50th St.
Lincoln, NE 68510

This article should answer some of the questions you have about computer memory and how it works in the TRS-80. I will also take a look at expanding memory in the keyboard and the expansion interface, as well as what to do if your computer memory ever goes on the blink.

To begin, let's look at the computer memory of a few decades ago. Magnetic core memory was probably the most popular for small computers and also used as secondary or external memory for larger systems in the 1950's and 60's. Magnetic core memory was developed at the Massachusetts Institute of Technology for the Whirlwind I computer. It had an access time of a few microseconds, (millionths of a second), was reliable, and was somewhat less expensive when compared to other types of memory in use at that time. It was not uncommon to have memories in the range of a few thousand to 32,000 words of storage. The drawbacks to these systems were their large physical size, their high power consumption, and their high price tags.

The magnetic core was made up of ferrite material, usually in the shape of little doughnuts. These doughnut shaped core elements were a millimeter or so in diameter and were strung by the thousands on a grid of wires. To store data the ferrite core was magnetized to read data, the magnetized ferrite core was used to induce an electrical current.

Random Acess Memory

Large scale integration has revolutionized the design of computers and other electronic devices. This is the process whereby tens of thousands of tiny circuits are fabricated upon a tiny chip of silicon, at times no more than a quarter of an inch on a side. What this has meant to computers should be apparent every time you power up your TRS-80.

If you own a TRS-80 with 16K memory, you will have eight 4116 16K RAM chips in your keyboard unit. 4116 is the model number of the RAM chip; 16K means the chip contains over 16,000 bits of storage; and RAM is an acronym for Random Access Memory.

RAM is a memory device that will write into or read from its storage any data the CPU/microprocessor requires, at any address the CPU specifies. The time required to do this is known as the RAM chips' access time, which is usually expressed in nanoseconds (NS). When you see an advertisement for extra memory, you will usually see a number like 450 ns or 250 ns after the chip model, meaning that the chip has an access time of 450 nanoseconds or 250 nanoseconds, which is fast when you think that a nanosecond is one billionth of a second. A nanosecond is to one second what one second is to 31.7 years.

There are eight RAM chips in the keyboard of the TRS-80 because the microprocessor chip is an eight-bit processor. This means that data is read into and from the chip eight bits at a time. In computer jargon, four bits equal one nybble and eight bits equal one byte. Twelve to sixteen bits equal a word.

Eight RAMs are necessary because only one bit may be addressed at a time in the 4116 RAM. In order to access eight bits of storage, you would need to access the

Table 1.

Phase 1	FF FF OO FF FF OO FF FF	FF = Major	OO = Minor
Phase 2	OO OO FF OO OO FF OO OO	OO = Major	FF = Minor
Phase 3	FF OO FF FF OO FF FF OO	FF = Major	OO = Minor
Phase 4	OO FF OO OO FF OO OO FF	OO = Major	FF = Minor
Phase 5	OO FF FF OO FF FF OO FF	FF = Major	OO = Minor
Phase 6	FF OO OO FF OO OO FF OO	OO = Major	FF = Minor

✓See List of Advertisers on page 306

same RAM chip eight times at eight different locations, increasing the time needed to access memory. Using eight RAM chips, all eight may be accessed simultaneously.

If you refer to the *TRS-80 Technical Reference Handbook*, you will see these eight RAM chips. They are represented by Z13 through Z20 on the first schematic sheet. The four most significant bits (MSB) of a byte are stored in RAM Z13, Z14, Z20, and Z15, and the four least significant bits (LSB) of a byte are stored in RAM chips Z19, Z18, Z16, and Z17. (The MSB of the hex number 7F is 7, the LSB is F.)

How Ram Works

Bits are stored in tiny cells in a rectangular array within the chip. To access any one of these cells, a coded binary address is input to an address decoder within RAM that selects one column and one row of the array. At the same time, a signal is sent from the CPU telling the RAM chip if it is to be a read or write operation. The desired data is then sent through the data lines.

The most simple storage cell in use today is made up of a small capacitor used to store a small electrical charge that represents a binary value of either 1 or 0. Usually 1 is represented by a plus charge of up to five volts and 0 is represented by ground, or one or two volts. An on/off switch (made of a small transistor) is used to connect the storage capacitor to a data line. There may be a large number of these storage capacitors sharing the same line, but only one cell can use the data line at any one time.

A storage cell is accessed as follows. The data lines act as a column indicator to the memory array. Selection lines correspond to rows at right angles to the data lines. When a selection line is activated, all storage cells on that line have their transistors turned on; because only one of these storage cells is on an activated data line at a time, only one storage cell is accessed.

The charge in the capacitor of a memory cell must be refreshed from time to time because of loss through capacitor leakage and accesses to it. Devices in RAM known as thresholding amplifiers, located on each of the RAM data lines, are responsible for this task. Whenever RAM is accessed the thresholding amplifier takes a sample of the signal and compares it to the threshold level. The threshold level is usually the charge midway between binary 1 and 0. The amplifier then regenerates the necessary binary level if the signal is low.

Regeneration of the threshold level is known as the refresh cycle of the RAM chip. If a RAM needs to be refreshed, it is known as a dynamic memory RAM. The TRS-80 is of this type. If a RAM does not need a refresh cycle, it is known as a static RAM.

Other Memory Devices

The TRS-80 also uses a memory device called a ROM, which contains BASIC and other routines needed for operation. ROM is an acronym for Read Only Memory. You

Listing 1.

```
1 CLS:INPUT"STARTING ADDRESS(XXXXD)";S:INPUT"ENDING ADDRESS(XXXXD)";E
5 D=225:FOR L=S TO E:POKE L,D:IF PEEK(L)<>D THEN 15 ELSE NEXTL
10 PRINT"TEST COMPLETED, NO ERRORS":STOP
15 PRINT L;"SHOULD'VE BEEN 225, BUT IT CONTAINS";PEEK(L):STOP
```

Listing 2.

```
                ;****************************************************
                ;*        SET-UP CONSTANTS FOR RAMTST
                ;****************************************************
                ;
3C00            DCOUNT   EQU  3C00H          ;COUNT DSP.
3C95            DPHASE   EQU  3C00H+149      ;PHASE DSP
3C9B            DPHASD   EQU  3C00H+155      ;DATA DSP
3CA9            DMAJBT   EQU  3C00H+169      ;MAJOR DSP
3CC0            DINFOM   EQU  3C00H+192      ;INFORM DSP
3CC9            DCONTS   EQU  3C00H+201      ;CONTENTS DSP
3CDD            DMABYT   EQU  3C00H+221      ;MAJOR BYTE DSP
3C37            DERROR   EQU  3C00H+55       ;ERROR DSP
3D1C            DTITLE   EQU  3C00H+284      ;TITLE DSP
3DC0            DPROMP   EQU  3C00H+448      ;PROM DSP
3DCE            DAUTHR   EQU  3C00H+14       ;PROGR NAME
3DDB            DFLASH   EQU  3C00H+475      ;FLASH DSP
3FC0            DTSTLO   EQU  3C00H+960      ;LOW EXT. DSP
3FD2            DSTART   EQU  DTSTLO+18      ;START DSP
3FDD            DSPEND   EQU  DTSTLO+29      ;END DSP
3C06            DSPCND   EQU  3C00H+6        ;COUNT DATA
3C3E            DSPERD   EQU  DERROR+7       ;ERROR DATA
4020            CURSOR   EQU  4020H          ;CURSOR POINTER
28A7            PRINT    EQU  28A7H          ;SUB PRINT
05D9            INPUTK   EQU  05D9H          ;SUB INPUT
                ;
                ;****************************************************
                ;*        START OF MAIN DRIVER
                ;****************************************************
                ;        INITL. RAM LOCATIONS
42E9                     ORG  42E9H          ;START OF PROG
42E9  218747             LD   HL,PHASE       ;GET LOC
42EC  3600               LD   (HL),00H       ;ZAP PHASE
42EE  218547             LD   HL,ERROR       ;GET LOC
42F1  3600               LD   (HL),00H       ;ZAP ERROR
42F3  CDC901             CALL 01C9H          ;CLS
42F6  111C3D             LD   DE,DTITLE      ;GET LOC
42F9  ED532040           LD   (CURSOR),DE    ;LOAD CURSOR
42FD  21CA45             LD   HL,TEXT1       ;POINT TO TEXT
4300  CDA728             CALL PRINT          ;DISPLAY
4303  11CE3D             LD   DE,DAUTHR      ;GET LOC
4306  ED532040           LD   (CURSOR),DE    ;LOAD CURSOR
430A  21D345             LD   HL,TEXT2       ;POINT TO TEXT
430D  CDA728             CALL PRINT          ;DISPLAY
4310  0604      LDSTUP   LD   B,4            ;LOAD COUNTER
4312  C5        GODLAY   PUSH BC             ;SAVE B
4313  010000             LD   BC,0           ;ZAP B
4316  CD6000             CALL 0060H          ;DELAY SUB
4319  C1                 POP  BC             ;GET B
431A  10F6               DJNZ GODLAY         ;LOOP UNTIL B=0
431C  CD8544             CALL CLPRMP         ;ZAP PROMPT
431F  11C03D             LD   DE,DPROMP      ;GET LOC.
4322  ED532040           LD   (CURSOR),DE    ;LOAD CURSOR
4326  21F745             LD   HL,TEXT3       ;POINT TO TEXT
4329  CDA728             CALL PRINT          ;DISPLAY IT
432C  2A2040             LD   HL,(CURSOR)    ;SET UP POINTER
432F  CDD905             CALL INPUTK         ;SUB INPUT
4332  7E                 LD   A,(HL)         ;GET 1ST BYTE
4333  FE2A               CP   02AH           ;IS IT '*'
4335  CAEB43             JP   Z,AUTO         ;YES-DO AUTO
4338  CD0244             CALL COVINP         ;NO-CONVERT INP
433B  ED537D47           LD   (START),DE     ;STORE START
433F  2A8147             LD   HL,(AUTOLO)    ;GET FIXED LOW
4342  35                 INC  DE             ;ADJUST
4343  B7                 OR   A              ;ZAP FLAG
4344  ED52               SBC  HL,DE          ;SUBTRACT
4346  FA5B43             JP   M,GETEND       ;INPUT OK-GO ON
4349  CD8544             CALL CLPRMP         ;ZAP PROMPT
434C  11C03D             LD   DE,DPROMP      ;GET LOCATION
434F  ED532040           LD   (CURSOR),DE    ;LOAD CURSOR
4353  215947             LD   HL,TEXT15      ;POINT TO TEXT
4356  CDA728             CALL PRINT          ;DISPLAY
4359  18B5               JR   LDSTUP         ;GOTO DELAY
435B  CD8544    GETEND   CALL CLPRMP         ;ZAP PROMPT
435E  11C03D             LD   DE,DPROMP      ;GET LOCATION
4361  ED532040           LD   (CURSOR),DE    ;LOAD CURSOR
4365  211E46             LD   HL,TEXT4       ;POINT TO TEXT
4368  CDA728             CALL PRINT          ;DISPLAY
```

Program continues

```
436B 2A2040             LD   HL,(CURSOR)     ;SET POINTER
436E CDD905             CALL INPUTK          ;SUB INPUT
4371 CD0244             CALL COVINP          ;CONVERT TO HEX
4374 ED537F47           LD   (END),DE        ;STORE END
4378 CD8544    GETCNT   CALL CLPRMP          ;ZAP PROMPT
437B 11C03D             LD   DE,DPROMP       ;GET LOCATION
437E ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
4382 213C46             LD   HL,TEXT5        ;POINT TO TEXT
4385 CDA728             CALL PRINT           ;DISPLAY
4388 2A2040             LD   HL,(CURSOR)     ;SET POINTER
438B CDD905             CALL INPUTK          ;SUB INPUT
438E CD0244             CALL COVINP          ;CONVERT TO HEX
4391 ED538347           LD   (COUNT),DE      ;STORE COUNT
4395 11003C             LD   DE,DCOUNT       ;GET LOCATION
4398 ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
439C 215F46             LD   HL,TEXT6        ;POINT TO TEXT
439F CDA728             CALL PRINT           ;DISPLAY
43A2 CD4944             CALL CNTDSP          ;DISPLAY COUNT
43A5 11373C             LD   DE,DERROR       ;GET LOCATION
43A8 ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
43AC 216646             LD   HL,TEXT7        ;POINT TO TEXT
43AF CDA728             CALL PRINT           ;DISPLAY
43B2 CD5B44             CALL ERRDSP          ;DSP ERROR DATA
43B5 CD8544             CALL CLPRMP          ;ZAP PROMPT
43B8 11C03F             LD   DE,DTSTLO       ;GET LOCATION
43BB ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
43BF 216E46             LD   HL,TEXT8        ;POINT TO TEXT
43C2 CDA728             CALL PRINT           ;DISPLAY
43C5 CD6644             CALL DSPLOC          ;DISP EXTENTS
43C8 11C03D             LD   DE,DPROMP       ;GET LOCATION
43CB ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
43CF 219146             LD   HL,TEXT9        ;POINT TO TEXT
43D2 CDA728             CALL PRINT           ;DISPLAY
43D5 CD9344             CALL TEST            ;MAIN SUB
43D8 CD8544    PGDONE   CALL CLPRMP          ;ZAP PROMPT
43DB 11C03D             LD   DE,DPROMP       ;GET LOCATION
43DE ED532040           LD   (CURSOR),DE     ;LOAD CURSOR
43E2 21AB46             LD   HL,TEXT10       ;POINT TO TEXT
43E5 CDA728             CALL PRINT           ;DISPLAY
43E8 C3191A             JP   1A19H           ;END-RETURN TO BASIC
               ;****************************************************************
               ;*        END OF DRIVER PROGRAM
               ;****************************************************************
               ;****************************************************************
               ;*                 SUBROUTINES
               ;* AUTO *  FIND THE TOP OF MEMORY
               ;****************************************************************
               ;
43EB 2A8147    AUTO     LD   HL,(AUTOLO)     ;GET LOW MEMORY
43EE 227D47             LD   (START),HL      ;STORE IN START
43F1 36AA      FILLUP   LD   (HL),0AAH       ;STICK IN AA
43F3 7E                 LD   A,(HL)          ;READ BACK
43F4 FEAA               CP   0AAH            ;IS IT AA?
43F6 2807               JR   Z,LOOP          ;NO-CONTINUE
43F8 2B                 DEC  HL              ;YES-CORRECT POINTER
43F9 227F47             LD   (END),HL        ;STORE END OF TEST
43FC C37843             JP   GETCNT          ;GET REPETITIONS
43FF 23        LOOP     INC  HL              ;ADJUST POINTER
4400 18EF               JR   FILLUP          ;LOOK HIGHER
               ;
               ;****************************************************************
               ;* COVINP   CONVERT ASCII TO HEX
               ;****************************************************************
4402 CD0B44    COVINP   CALL ASXHEX          ;CONVERSION SUB
4405 57                 LD   D,A             ;STORE HEX IN D
4406 CD0B44             CALL ASXHEX          ;CONVERSION SUB
4409 5F                 LD   E,A             ;STORE HEX IN E
440A C9                 RET                  ;RETURN W/HEX IN DE
440B 0E00      ASXHEX   LD   C,00H           ;ZAP REG C
440D 7E                 LD   A,(HL)          ;GET 1ST BYTE
440E CD1844             CALL CVERT           ;CALL BYTE CONVERTER
4411 23                 INC  HL              ;POINT TO NEXT BYTE
4412 7E                 LD   A,(HL)          ;GET 2ND BYTE
4413 CD1844             CALL CVERT           ;CONVERT BYTE
4416 23                 INC  HL              ;POINT TO NEXT
4417 C9                 RET                  ;BYTE CONVERTED
4418 CB21      CVERT    SLA  C               ;SHIFT C LEFT 1
441A CB21               SLA  C               ;AGAIN
441C CB21               SLA  C               ;AGAIN
441E CB21               SLA  C               ;DO ONE MOR TIME
4420 D630               SUB  30H             ;SUBTRACT 30H
4422 FE0A               CP   10              ;GREATER THAN 10
4424 FA2944             JP   M,JUMP          ;NO-STORE IT
4427 D607               SUB  7               ;SUBTRACT 7
4429 81        JUMP     ADD  A,C             ;PUT WITH REG C
442A 4F                 LD   C,A             ;COPY BACK TO C
442B C9                 RET                  ;RETURN-HEX IN REG C
               ;
               ;****************************************************************
               ;* BXASH  CONVERT HEX BACK TO ASCII AND DISPLAY
               ;****************************************************************
442C 3EF0      BXASH    LD   A,0F0H          ;PUT MASK IN A
442E A1                 AND  C               ;MASK OFF C
442F 0F                 RRCA                 ;ROTATE RIGHT
```

Program continues

cannot write into ROM, only read from it. Data is put into the ROM at the time it is manufactured. ROMs are like RAMs, except their storage capacitors are replaced with an open circuit or are connected to ground, representing one or the other of the two binary states.

Another memory device is called PROM. PROM is an acronym for Programmable Read Only Memory. When a PROM is manufactured, its storage cells have a small fusible link from their transistors to ground in place of the storage capacitors. Data is placed in the array by applying a pattern of electrical signals strong enough to break the unwanted links. Once a PROM has been programmed, it cannot be altered. A PROM will retain its data after power has been removed from it.

The next device is an EPROM, an acronym for Erasable Programmable Read Only Memory. Data is written into an EPROM through electronic means, but only after all storage cells have been erased. In order to erase the cells, ultraviolet light is used. Once data has been written into the cells, they cannot be altered until all cells are once again erased.

Enlarge Your Memory

The Z-80 microprocessor in the TRS-80 will address up to 65536 bytes of RAM. If you have a 4K system, you can add up to 16K of total RAM within the keyboard by buying one of many conversion kits or by having Radio Shack install one. Most of these kits come with the RAM chips, jumpers needed for installation and instructions.

(Remember that if you open up your keyboard, you will void your warranty if it is still in effect.)

If you already have 16K of RAM in your keyboard and want more, you will need to buy an expansion interface. This can give you up to 48K of memory.

If you buy the interface without memory, it's easy to install. If you are inserting only 16K of RAM, find RAM sockets Z9 through Z16 in the interface. If you are installing 32K of RAM at one time, you needn't worry about which sockets you fill first. For 32K, you will fill sockets Z1 through Z8 and sockets Z9 through Z16. You will want to use low power 4116 RAMs with an access of at least 450 ns.

Caution is the word when handling RAM chips. They will arrive sitting in conductive foam or in a conductive tube or foil (any static discharge will damage the chips). When handling MOS (Metal Oxide Semiconductor) devices you might consider grounding yourself with special straps. Try not to wear clothing that contains static. Using an IC insertion tool, such as Radio Shack's 276-1574, is also helpful when working with chips.

The number one pin of the chip must agree with the number one hole of the socket. The first pin usually has a small circle over it, or a notch cut into the end where the number one pin is located. If you are un-

See Opposite Page ⇨

sure about the pins, find someone to help you. If you plug the chip in backwards, you could fry it when you turn on the power.

After you have the chip sitting in the socket, make sure you haven't bent any pins under the body of the chip. Don't use too much force when you press the chip into the socket.

Once you've seated the chips, put everything back together and power up your system. Remember, if you do not have a disk on your expansion interface, hold the Break key down when you power up. This avoids a screen full of garbage. Type in "? MEM", and your new amount of RAM should be displayed in decimal on the CRT.

Memory Failure

RAM problems may show up in a variety of ways. I will list only a few of the more common symptoms.

If you are running a program known to be error free and the program begins to bomb out (abend), or if your known variables begin to become different values for no reason, or the graphics begin to look like garbage on the tube, it might not be a bad idea to check your memory. If your memory checks out okay, I would suggest taking your system to a Radio Shack Repair Center.

"If . . . your program begins to bomb out . . . it might not be a bad idea to check your memory."

The best way to find a RAM problem is to let your computer find it. Load a diagnostic program into your system that will read and write specific data into and out of specific address locations. When the program finds a location that does not respond in the way that it should, your problem is half solved. The next problem is to determine which of the eight RAM chips are bad.

An easy approach to the first problem would be to write a simple program in BASIC (Listing 1). This program will write a value of 255 (FF in hex) into the RAM locations you input through prompts from the program. The program will first write 255 into an address and then read the data back and compare it. If the compare is not equal, the program will branch to an error display routine and stop. If no errors are encountered, the program will display this information at the end of the test.

Listing 1 is not very fast, nor is it completely accurate. What, for example, would happen if a bit in RAM was stuck at 1 (a bit that would not register a 0)? Program Listing 1 would never detect such an error. What would happen if the error was in the area of RAM that our program must execute out of? The program might not work, or the problem might not be detected.

```
4430  0F              RRCA              ;AGAIN
4431  0F              RRCA              ;AGAIN
4432  0F              RRCA              ;ONE MORE TIME
4433  CD3D44          CALL  CVERT1      ;DO CONVERSION
4436  3E0F            LD    A,0FH       ;LOAD A WITH OF
4438  A1              AND   C           ;MASK OFF C
4439  CD3D44          CALL  CVERT1      ;CONVERT
443C  C9              RET               ;CONVERSION DONE
443D  C630     CVERT1 ADD   A,30H       ;ADD 30H
443F  FE3A            CP    3AH         ;IS IT LESS THEN 10?
4441  FA4644          JP    M,JUMP1     ;YES-GET READY TO STORE
4444  C607            ADD   A,7         ;NO-ADJUST
4446  77       JUMP1  LD    (HL),A      ;DISPLAY IT
4447  23              INC   HL          ;POINT TO NEXT
4448  C9              RET               ;RETURN AND GET NEXT BYTE
               ;
               ;*******************************************************
               ;* CNTDSP  DISPLAY PROGRAM REPETITIONS
               ;*******************************************************
4449  E5       CNTDSP PUSH  HL          ;SAVE ON STACK
444A  21063C          LD    HL,DSPCND   ;POINT FOR DISPLAY
444D  ED5B8347        LD    DE,(COUNT)  ;GET COUNT
4451  4A              LD    C,D         ;LOAD MSB INTO C
4452  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4455  4B              LD    C,E         ;LOAD LSB INTO C
4456  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4459  E1              POP   HL          ;GET HL BACK
445A  C9              RET               ;RETURN-COUNT DISPLAYED
               ;*******************************************************
               ;* ERRDSP  DISPLAY NUMBER OF ERRORS
               ;*******************************************************
445B  213E3C   ERRDSP LD    HL,DSPERD   ;POINT FOR DISPLAY
445E  3A8547          LD    A,(ERROR)   ;GET ERROR COUNT
4461  4F              LD    C,A         ;LOAD INTO C
4462  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4465  C9              RET               ;RETURN-COUNT OF ERRORS
                                          DISP.
               ;*******************************************************
               ;* DSPLOC  DISPLAY MEMORY LOCATION
               ;*******************************************************
4466  21D23F   DSPLOC LD    HL,DSTART   ;POINT TO SCREEN
4469  ED5B7D47        LD    DE,(START)  ;GET START
446D  4A              LD    C,D         ;LOAD MSB
446E  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4471  4B              LD    C,E         ;LOAD LSB
4472  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4475  21DD3F          LD    HL,DSPEND   ;POINT TO DISPLAY
4478  ED5B7F47        LD    DE,(END)    ;GET END
447C  4A              LD    C,D         ;LOAD MSB
447D  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4480  4B              LD    C,E         ;LOAD LSB
4481  CD2C44          CALL  BXASH       ;CONVERT TO ASCII
4484  C9              RET               ;RETURN-COMPLETE
               ;*******************************************************
               ;* CLPRMP   ZAP PROMPT POSITION
               ;*******************************************************
4485  11C03D   CLPRMP LD    DE,DPROMP   ;GET LOCATION
4488  ED532040        LD    (CURSOR),DE ;LOAD CURSOR
448C  21EA46          LD    HL,BLANK    ;POINT TO TEXT
448F  CDA728          CALL  PRINT       ;DISPLAY
4492  C9              RET               ;PROMPT CLEAR-RETURN
               ;*******************************************************
               ;* TEST MAIN SUBROUTINE  TEST MEMORY
               ;*******************************************************
4493  3E02     TEST   LD    A,2         ;LOAD SPACE KEY
4495  328647   NEXONE LD    (CNTPOS),A  ;SAVE IT
4498  3EFF            LD    A,0FFH      ;LOAD A WITH FF
449A  4F       GETNEX LD    C,A         ;STORE IN C
449B  CD9D45          CALL  PHASEP      ;DISPLAY PHASE
449E  CDB744          CALL  FILMEM      ;LOAD UP MEMORY-ALL
44A1  CDC644          CALL  FILLOP      ;LOAD UP MEMORY-SOME
44A4  CDDB44          CALL  VERIFY      ;CHECK-OUT LOADS
44A7  79              LD    A,C         ;GET FILL BYTE
44A8  2F              CPL               ;COMPLIMENT A
44A9  B7              OR    A           ;IS IT ZERO?
44AA  28EE            JR    Z,GETNEX    ;YES-DO NEXT PHASE
44AC  3A8647          LD    A,(CNTPOS)  ;GET SPACE
44AF  FE00            CP    00          ;ARE WE DONE?
44B1  CA0D45          JP    Z,CHKCNT    ;TEST DONE-IS PROGRAM?
44B4  3D              DEC   A           ;NO-DO NEXT SPACE
44B5  18DE            JR    NEXONE      ;CONTINUE
44B7  21DB3D   FILMEM LD    HL,DFLASH   ;GET LOCATION
44BA  3657            LD    (HL),57H    ;DISPLAY W
44BC  CDFC44          CALL  GETEXT      ;GET EXTENTS
44BF  71       STUFF  LD    (HL),C      ;STUFF BYTE
44C0  CD0245          CALL  CHKEND      ;IS MEMORY FULL
44C3  C8              RET   Z           ;IF Z THEN RETURN
44C4  18F9            JR    STUFF       ;KEEP STUFFING
44C6  CDFC44   FILLOP CALL  GETEXT      ;GET EXTENTS
44C9  47              LD    B,A         ;LOAD B WITH SPACE
44CA  FE00            CP    00          ;TEST ALL SPACES DONE?
44CC  2005            JR    NZ,SPADWN   ;NO-SPACE DOWN
44CE  79       STUFIT LD    A,C         ;LOAD A
44CF  2F              CPL               ;COMPLIMENT A
```

Program continues

```
44D0  77                  LD    (HL),A        ;STUFF A
44D1  0603                LD    B,3           ;LOAD SPACE
44D3  CD0245     SPADWN   CALL  CHKEND        ;IS MEMORY FULL
44D6  C8                  RET   Z             ;RETURN IF YES
44D7  10FA                DJNZ  SPADWN        ;SPACE UNTIL RIGHT
44D9  18F3                JR    STUFIT        ;DO ANOTHER STUFF
44DB  21D83D     VERIFY   LD    HL,DFLASH     ;GET LOCATION
44DE  3652                LD    (HL),52H      ;DISPLAY R
44E0  CDFC44              CALL  GETEXT        ;GET EXTENTS
44E3  47                  LD    B,A           ;LOAD SPACE
44E4  FE00                CP    00            ;IS SPACE 00?
44E6  2007                JR    NZ,VERFAL     ;CHECK MAJOR
44E8  79         VERSOM   LD    A,C           ;LOAD A
44E9  2F                  CPL                 ;COMPLIMENT A
44EA  BE                  CP    (HL)          ;ARE THEY EQUAL?
44EB  0603                LD    B,3           ;LOAD SPACE
44ED  1802                JR    ERROUT        ;CHECK FOR ERROR
44EF  79         VERFAL   LD    A,C           ;LOAD A
44F0  BE                  CP    (HL)          ;IS MEMORY CORRECT?
44F1  C42445     ERROUT   CALL  NZ,FAIL       ;NO-GOTO FAIL
44F4  CD0245              CALL  CHKEND        ;END OF MEMORY?
44F7  C8                  RET   Z             ;YES,RETURN
44F8  10F5                DJNZ  VERFAL        ;CHECK MAJOR
44FA  18EC                JR    VERSOM        ;CHECK MINOR
44FC  2A7D47     GETEXT   LD    HL,(START)    ;GET START
44FF  3A8647              LD    A,(CNTPOS)    ;GET SPACE
                 ;*****************************************************
                 ;* CHKEND  CHECK FOR END OF MEMORY EXTENT
                 ;*****************************************************
4502  E5         CHKEND   PUSH  HL            ;SAVE HL ON STACK
4503  EB                  EX    DE,HL         ;LOAD DE
4504  2A7F47              LD    HL,(END)      ;PUT END IN HL
4507  B7                  OR    A             ;CLEAR FLAG
4508  ED52                SBC   HL,DE         ;ARE WE AT END OF MEMORY?
450A  E1                  POP   HL            ;GET HL BACK
450B  23                  INC   HL            ;POINT 1 HIGHER IN CASE NOT
450C  C9                  RET                 ;RETURN-IF Z THEN EXIT
                                               CALLER
                 ;*****************************************************
                 ;* CHKCNT  CHECK ON PROGRAM REPETITIONS
                 ;*****************************************************
450D  2A8347     CHKCNT   LD    HL,(COUNT)    ;GET PROGRAM COUNT
4510  2B                  DEC   HL            ;DEC. ONE FROM COUNT
4511  228347              LD    (COUNT),HL    ;STORE NEW COUNT
4514  CD4944              CALL  CNTDSP        ;DISPLAY NEW COUNT
4517  3E00                LD    A,0           ;ZAP
4519  BC                  CP    H             ;IS MSB=0?
451A  C29344              JP    NZ,TEST       ;NO-DO ANOTHER TEST
451D  BD                  CP    L             ;IS LSB=0?
451E  C29344              JP    NZ,TEST       ;NO-DO ANOTHER TEST
4521  C3D843              JP    PGDONE        ;YES-PROGRAM ABOUT DONE
                 ;*****************************************************
                 ;* FAIL    BAD MEMORY BYTE HAS BEEN FOUND
                 ;*****************************************************
4524  C5         FAIL     PUSH  BC            ;SAVE BC
4525  E5                  PUSH  HL            ;SAVE HL
4526  E5                  PUSH  HL            ;AGAIN
4527  C5                  PUSH  BC            ;SAVE BC AGAIN
4528  E5                  PUSH  HL            ;SAVE HL ONE MORE TIME
4529  11C03C              LD    DE,DINFORM    ;GET LOCATION
452C  ED532040            LD    (CURSOR),DE   ;LOAD CURSOR
4530  213B47              LD    HL,TEXT14     ;POINT TO TEXT
4533  CDA728              CALL  PRINT         ;DISPLAY IT
4536  E1                  POP   HL            ;GET MEMORY LOCATION
4537  4E                  LD    C,(HL)        ;LOAD CONTENTS
4538  21C93C              LD    HL,DCONTS     ;POINT TO CRT
453B  CD2C44              CALL  BXASH         ;CONVERT TO ASCII AND
                                               DISPLAY
453E  C1                  POP   BC            ;GET MAJOR BYTE
453F  21DD3C              LD    HL,DMABYT     ;POINT TO CRT
4542  CD2C44              CALL  BXASH         ;CONVERT AND DISPLAY
4545  218547              LD    HL,ERROR      ;GET ADDRESS
4548  34                  INC   (HL)          ;ADD 1 TO ERROR CONTENTS
4549  CD5B44              CALL  ERRDSP        ;DISPLAY ERRORS
454C  CD8544              CALL  CLPRMP        ;ZAP PROMPT
454F  11C03D              LD    DE,DPROMP     ;GET LOCATION
4552  ED532040            LD    (CURSOR),DE   ;LOAD CURSOR
4556  21C446              LD    HL,TEXT11     ;POINT TO TEXT
4559  CDA728              CALL  PRINT         ;DISPLAY
455C  2A2040              LD    HL,(CURSOR)   ;GET READY
455F  D1                  POP   DE            ;GET BAD ADDRESS
4560  4A                  LD    C,D           ;LOAD MSB
4561  CD2C44              CALL  BXASH         ;CONVERT TO ASCII
4564  4B                  LD    C,E           ;LOAD LSB
4565  CD2C44              CALL  BXASH         ;CONVERT TO ASCII
4568  222040              LD    (CURSOR),HL   ;LOAD CURSOR
456B  21D746              LD    HL,TEXT12     ;POINT TO TEXT
456E  CDA728              CALL  PRINT         ;DISPLAY
4571  2A2040              LD    HL,(CURSOR)   ;GET READY FOR INPUT
4574  CDD905              CALL  INPUTK        ;GET INPUT
4577  7E                  LD    A,(HL)        ;GET 1ST CHARACTER
4578  FE59                CP    59H           ;IS IT Y?
457A  C2D843              JP    NZ,PGDONE     ;NO-GET READY TO END
457D  CD8544              CALL  CLPRMP        ;YES-CLEAR PROMPT
```

Program continues

RAMTST

I wrote a machine language program to test the RAM in my system. The name of my program is RAMTST. RAMTST (Listing 2) is a machine language program written with the help of the Radio Shack Editor/Assembler (EDTASM). With all the comments that are in the source of the program, you will need a system of at least 32K to assemble it. The advantage of using EDTASM would be that you could compile the program to use any address you want to execute from. If you do not have EDTASM or 32K RAM, you could use T-BUG to inset the code into RAM, which might be a big job, as RAMTST has 1182 bytes of object code to type in.

The version of RAMTST I use is loaded into addresses 42E9H through 4787H. The program tests RAM locations 4788 to the top of memory. RAMTST is loaded like any other system tape, and loads in about 25 seconds. Type / (enter) after the program has loaded to begin execution.

RAMTST will first ask for the starting address of the location of RAM to be tested. If the location entered is below 4788 (hex), the program will respond with an error message. (The program will not allow you to kill it by writing over it in RAM.) If you respond with "*", the program will do an automatic phase that will find the top of memory without your help, and go on to the third prompt.

"Large scale integration has revolutionized the design of computers. . ."

The second prompt asks for the end address of memory that you want tested. The third prompt asks for how many repetitions of RAMTST you want. You can enter up to 65000 plus if you want, but it will take a while to execute. After all prompts are satisfied, the program will begin execution. In a system of 32K RAM, one repetition of RAMTST will take about 40 seconds.

What RAMTST Does

RAMTST will process through six phases. We will take a look at each phase in turn.

Phase one writes FF H into all bytes of RAM. It then writes OO H into every third RAM byte (Table 1). The upper left-hand corner of your screen will display the repetitions left to complete RAMTST. The right-hand corner will display the number of errors encountered. The middle of the screen will show that the program is running, as well as what the major byte is.

In phase one, the major byte would be FF H and the minor byte would be OO H. The program will also display what phase is currently processing, as well as whether it is writing out data or whether it is verifying data displayed as W for write and R for read

```
4580  11C03C          LD   DE,DINFOM       ;POINT TO CRT
4583  ED532040        LD   (CURSOR),DE     ;LOAD CURSOR
4587  21EA46          LD   HL,BLANK        ;POINT TO BLANK
458A  CDA728          CALL PRINT           ;DISPLAY
458D  11C03D          LD   DE,DPROMP       ;GET LOCATION
4590  ED532040        LD   (CURSOR),DE     ;LOAD CURSOR
4594  219146          LD   HL,TEXT9        ;POINT TO TEXT
4597  CDA728          CALL PRINT           ;DISPLAY
459A  E1              POP  HL              ;GET ADDRESS
459B  C1              POP  BC              ;GET BYTE
459C  C9              RET                  ;ERROR DISPLAYED RETURN
                ;*****************************************************
                ; PHASEP  DISPLAY CURRENT PROGRAM PHASE
                ;*****************************************************
459D  C5        PHASEP  PUSH BC            ;SAVE BC ON STACK
459E  C5                PUSH BC            ;AGAIN
459F  218747            LD   HL,PHASE      ;GET PHASE ADDRESS
45A2  34                INC  (HL)          ;ADD 1 TO PHASE
45A3  7E                LD   A,(HL)        ;PUT PHASE INTO A
45A4  FE07              CP   7             ;IS IT = TO 7?
45A6  2002              JR   NZ,PHASEDP    ;NO-DISPLAY IT
45A8  3601              LD   (HL),1        ;RESET TO 1 IF YES
45AA  11953C    PHASDP  LD   DE,DPHASE     ;GET LOCATION
45AD  ED532040          LD   (CURSOR),DE   ;LOAD CURSOR
45B1  212647            LD   HL,TEXT13     ;POINT TO TEXT
45B4  CDA728            CALL PRINT         ;DISPLAY IT
45B7  218747            LD   HL,PHASE      ;POINT TO PHASE
45BA  4E                LD   C,(HL)        ;LOAD PHASE INTO C
45BB  219B3C            LD   HL,DPHASD     ;POINT TO CRT
45BE  CD2C44            CALL BXASH         ;CONVERT TO ASCII AND DSP
45C1  C1                POP  BC            ;GET BACK BYTE
45C2  21A93C            LD   HL,DMAJBT     ;POINT TO CRT
45C5  CD2C44            CALL BXASH         ;CONVERT AND DISPLAY
45C8  C1                POP  BC            ;GET BYTE
45C9  C9                RET                ;RETURN-PHASE DISPLAYED
                ;
                ;*****************************************************
                ;*   TEXTS AND STOREAGE LOCATIIONS
                ;*****************************************************
                ;
45CA  52414D20  TEXT1   DEFM 'RAM TEST'
45CE  54455354
45D2  00                DEFB 00H
45D3  42592052  TEXT2   DEFM 'BY ROBERT D. RANDALL  AUG. 31, 1980'
45D7  4F424552
45DB  5420442E
45DF  2052414E
45E3  44414C4C
45E7  20202041
45EB  55472E33
45EF  312C2031
45F3  393830
45F6  00                DEFB 00H
45F7  454E5445  TEXT3   DEFM 'ENTER *,OR STARTING ADDRESS (XXXXH) '
45FB  5220202A
45FF  202C4F52
4603  20535441
4607  5254494E
460B  47204144
460F  44524553
4613  53202858
4617  58585848
461B  2920
461D  00                DEFB 00H
461E  454E5445  TEXT4   DEFM 'ENTER ENDING ADDRESS (XXXXH) '
4622  5220454E
4626  44494E47
462A  20414444
462E  52455353
4632  20285858
4636  58584829
463A  20
463B  00                DEFB 00H
463C  454E5445  TEXT5   DEFM 'ENTER REPETITIONS OF TEST (XXXXH) '
4640  52205245
4644  50455449
4648  54494F4E
464C  53204F46
4650  20544553
4654  54202858
4658  58585848
465C  2920
465E  00                DEFB 00H
465F  434F554E  TEXT6   DEFM 'COUNT='
4663  543D
4665  00                DEFB 00H
4666  4552524F  TEXT7   DEFM 'ERRORS='
466A  52533D
466D  00                DEFB 00H
466E  54455354  TEXT8   DEFM 'TESTING LOCATIONS=    H THRU     H'
4672  494E4720
4676  4C4F4341
467A  54494F4E
467E  533D2020
4682  20204820
```

Program continues

or verify. At the bottom of the display, the program will display the area of RAM being tested. This area will be the one that was entered through prompts from the program or the address that was located by the auto function of RAMTST.

If an error should be encountered, a display will show you the location where the error occurred, as well as what the contents of the byte are and what the contents should have been. You will then be prompted as to whether you want to continue with the test or end RAMTST and return to BASIC mode. If you wish to continue testing, the error counter will update itself to show the error.

If every byte checks out in phase one, the program will switch the major and minor byte and fill all locations as in phase one. This will be phase two.

In phase three, the major byte becomes FF H and the minor bytes becomes OO H. All core positions are loaded with the major byte, but this time the first minor byte will be loaded one byte sooner than in phase one (Table 1). Every third byte is then loaded in turn with the minor byte, as before.

Phase four switches the major and minor bytes and loads as in phase three. See Table 1 for what happens in phases five and six.

If an error is found and you decide not to continue with the test and respond NO to the prompt, you will return to BASIC mode. If you wish to return to RAMTST, from BASIC, you may do so by typing SYSTEM and / (enter). Caution: this will only work if you have not disturbed RAMTST in memory. If RAMTST has found a bad location in memory, you must find out which one of the RAM chips is at fault.

In the TRS-80, each bit in a byte will be found in one RAM chip. The most significant four bits of a byte are in RAM chips Z13, Z14, Z20, and Z15, and RAM chips Z19, Z18, Z16, and Z17 contain the least significant four bits. As an example, assume that byte 32781 contains A3 H or 10100011 binary. At location 32781 of each of the eight RAM chips, the bit would be as follows: Z13 would be a one, Z14 would be a zero, Z20 would be a one, and Z15 would be a zero. Of the least significant four bits, Z19 and Z18 would have a zero, and Z16 and Z17 would both have a one at the position.

Knowing where there is a possible RAM problem and how the RAM chips store data, it should not be much of a problem discovering which of the RAM chips is at fault. You can use a program such as Listing 3 to find any stuck bits.

Listing 3 will display what it was trying to write and what the byte contains. From this you should be able to determine the bit at fault. In other words, if the program was trying to write 04 H (00000100 B) and the byte contained OO H (00000000 B), this would point to RAM chip Z18. Substituting a good RAM chip at this socket and running the program again without errors, will assure you theat you have found the bad RAM chip. ■

```
4686  54485255
468A  20202020
468E  2048
4690  00        2090            DEFB 00H
4691  52414D20  2095 TEXT9      DEFM 'RAM TEST IS NOW EXECUTING'
4695  54455354
4699  20495320
469D  4E4F5720
46A1  45584543
46A5  5554494E
46A9  47
46AA  00        2100            DEFB 00H
46AB  52414D20  2105 TEXT10     DEFM 'RAM TEST IS NOW COMPLETE'
46AF  54455354
46BE  20495320
46B7  4E4F5720
46BB  434F4D50
46BF  4C455445
46BF  4C455445
46C8  00        2110            DEFB 00H.
46C4  2B455252  2115 TEXT11     DEFM '+ERROR+ LOCATION='
46CB  4F522B20
46CC  4C4F4341
46D0  54494F4E
46D4  3D22
46D6  00        2120            DEFB 00H
46D7  48202843  2125 TEXT12     DEFM 'H (CCNTINUE? Y-N) '
46DB  4F4E5449
46DF  4E55453F
46E3  20592D4E
46E7  2920
46E9  00        2130            DEFB 00H
46EA  20202020  2135 BLANK      DEFM '
46EF  20202020
46F3  20202020
46F4  20202020
46F8  20202020
46FC  20202020
4700  20202020
4704  20202020
4708  20202020
470C  20202020
4710  20202020
4714  20202020
4718  20202020
471C  20202020
4720  20202020
4724  20
4725  00        2140            DEFB 00H
4726  50484153  2145 TEXT13     DEFM 'PHASE=  MAJOR BYTE='
472A  453D2020
472E  204D414A
4732  4F522042
4736  5954453D
473A  00        2150            DEFB 00H
473B  434F4E54  2155 TEXT14     DEFM 'CONTENTS=   SHOULD HAVE BEEN='
473F  454E5453
4743  3D202020
4747  53484F55
474B  4C442048
474F  41564520
4753  4245454E
4757  3D
4758  00        2160            DEFB 00H
4759  2A434155  2165 TEXT15     DEFM '*CAUTION* ADDRESS TOO LOW, RE-INPUT'
475D  54494F4E
4761  2A204144
4765  44524553
4769  5320544F
476D  4F204C4F
4771  572C2052
4775  452D494E
4779  505554
477C  00        2170            DEFB 00H
477D  0000      2175 START      DEFW 0000H
477F  0000      2180 END        DEFW 0000H
4781  8847      2185 AUTOLO     DEFW 4788H
4783  0000      2190 COUNT      DEFW 0000H
4785  00        2195 ERROR      DEFB 00H
4786  00        2200 CNTPOS     DEFB 00H
4787  00        2205 PHASE      DEFB 00H
42E9            2210 END        42E9H
0000 TOTAL ERRORS
```

Listing 3.

```
1  CLS:INPUT"ENTER RAM ADDRESS(XXXXXD(";L:D = 1
5  POKE L,0:IF PEAK(L)<>0 THEN PRINT"BYTE";L;"WON'T CLEAR TO ZERO, IT CONTAINS";PEAK(L):STOP
10 FOR X = TO 8:POKE L,D:IF PEAK(L) D THEN 20
15 D=D*2:IF D>128 THEN PRINT"TEST COMPLETE, NO ERRORS":STOP:ELSE NEXT X
20 PRINT"ERROR AT";L;" SHOULD'VE CONTAINED";D;" BUT IT CONTAINED";PEEK(L);"INSTEAD":STOP
```

FMG CORPORATION

M-530

FMG CORPORATION NOW CARRIES GRAHAM-DORIAN & PEACHTREE SOFTWARE

VISA® master charge

P.O. Box 16020
Fort Worth, Texas
76133
(817) 294-2510

✓12

NEW VERSATILITY

For Your TRS-80

CP/M® 2

CONTROL PROGRAM FOR MICROCOMPUTERS ENABLING YOU TO RUN SOFTWARE PUBLISHED FOR CP/M 1.4 ON THE TRS-80

CP/M is considered the industry standard disk operating system because it gives you the hardware-independent interface you need to make your computer work for you. CP/M 2.0 is the latest in the evolution of a proven reliable and efficient software system. FMG CORPORATION NOW OFFERS THE CP/M 2.0 FOR THE TRS-80.

It features an enhanced upward compatible file system and powerful new random access capabilities. The CP/M 2.0 from FMG provides the ability to run software published for the CP/M system, on the TRS-80 Model II. From minidisks, floppy disks, all the way to high-capacity hard disks, the flexibility of CP/M 2.0 makes it a truly universal operating system. The package includes an 8" system disk, editor, assembler and debugger for the TRS-80

Available in Format A, B, C, G only . . . $200/$25

MP/M™

MULTI-PROGRAMMING MONITOR

NEW INDUSTRY STANDARD

A deluxe operating system that provides big computer facilities at small computer prices. MP/M is a monitor program which operates with your microcomputer to provide multi-terminal access with multiprogramming at each terminal. Best of all, it's CP/M compatible which means you can run a wide variety or programming languages, applications packages, and development software.

You can run simultaneous editors, program translators, and background printer spoolers. Or you can use MP/M for data entry or data-base access from remote terminals. Or you can use MP/M realtime features to monitor an assembly line and automatically schedule programs for execution throughout the day. MP/M makes an excellent focal point for a cluster of connected microcomputers. The possibilities are limitless.

(Format B) . . . . $450/$35
(Format G) . . . . $300/$35

*CP/M and MP/M are trademarks of Digital Research.
Z80 is a trademark of Zilog, Inc.
TRS-80 is a trademark of Tandy Corp.
Pascal/M is a trademark of Sorcim.

All FMG Software Products Include All Necessary Manuals

• MICROPRO INTERNATIONAL

SOFTWARE WITH MANUAL / MANUAL ALONE

(M) **SUPER-SORT I** – Sort, merge, extract utility as absolute executable program or linkable module in Microsoft format. Sorts fixed or variable records with data in binary, BCD, Packed Decimal, EBCDIC, ASCII, floating & fixed point, exponential, field justified, etc. Even variable number of fields per record! $225/$25

SUPER-SORT II – Above available as absolute program only . . . . $175/$25

SUPER-SORT III – As II without SELECT/EXCLUDE . . . . $125/$25

WORD-STAR – Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace. Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning . . . . $495/$40

WORD-STAR Customization Notes – For sophisticated users who do not have one of the many standard terminal or printer configurations in the distribution version of WORD-STAR . . . . NA/$95

WORD-MASTER Text Editor – In one mode has superset of CP/M's ED commands including global searching and replacing, forwards and backwards in file in video mode, provides full screen editor for users with serial addressable-cursor terminal . . . . . . $150/$25

FLOPPY SAVER – Protection for center holes of 5" and 8" floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool and tough 7 mil mylar reinforcing rings for 25 diskettes.

5", Kit $14.95
5", Rings only $7.95
8", Kit $16.95
8", Rings only $8.95

HEAD CLEANING DISKETTE – Cleans the drive Read/Write head in 30 seconds. Diskette absorbs loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use.

8" $32.00
5¼" $30.00

(M) • **DESPOOL** — Allows flexibility and efficiency. (Disk file printing can be accomplished while simultaneously using the computer for other tasks) Slower printers do not tie up the computer. Requires 32K minimum . . . . . . $75/$10

(M) • **SCREEN EDIT** — Text editor for program entry — allows user the ability to see entries as they are being made. Has command which enables user to move the viewed position of the file anywhere within the current data file OR add information anywhere in the file. Requires 16K minimum . . . . $125/$25
(Also available in TRS DOS format.
Specify model or TRS-80)

(M) • **MAC** — Disk-based, powerful macro assembler utilizes Standard Intel Mnemonics. Includes macro processor.
The CP/M 8080 Macro Assembler reads assembly language statement from a diskette file and produces an Intel "HEX" format object file on the disk suitable for processing in the TRS-CP/M environment. Requires 32K minimum and CP/M . . . . $100/$25

(M) • **ZSID** — Efficient and reliable program testing system for Z80 microcomputers. Capabilities include traceback and histogram facilities. Allows real time break points.
ZSID is a symbolic debugger which expands upon the features of the TRS-CP/M standard debugger, providing greatly enhanced facilities for assembly language program check-out. Requires 32K minimum and CP/M . . . . . . $99/$25

(T) **MAIL LIST** — Mailing list maintenance package. No sorting required to print normal address labels in zip code sequence. Supports new larger zip code. Sorts and selects on multiple fields. Labels may be printed in user selectable formats. Includes sort and select utilities $300/$35

• PEACHTREE SOFTWARE SYSTEMS

(T) **GENERAL LEDGER** – Records details of all financial transactions. Generates a balance sheet and an income statement. Flexible and adaptable design for both small businesses and firms performing client writeup services. Produces reports as follows: Trial Balance, Transaction Registers, Balance Sheet, Prior Year Comparative Balance Sheet, Income Statement, Prior Year Comparative Income Statement and Department Income Statements. Interactive with other PEACHTREE accounting packages. Supplied in source code for Microsoft BASIC . . . . $990/$30

(T) **ACCOUNTS PAYABLE** – Tracks current and aged payables and incorporates a check writing feature. Maintains a complete vendor file with information on purchase orders and discount terms as well as active account status. Produces reports as follows: Open Voucher Report, Accounts Payable Aging Report and Cash Requirements. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft. BASIC . . . . $990/$30

(T) **ACCOUNTS RECEIVABLE** – Generates invoice register and complete monthly statements. Tracks current and aged receivables. Maintains customer file including credit information and account status. The current status of any customer account is instantly available. Produces reports as follows: Aged Accounts Receivable, Invoice Register, Payment and Adjustment Register and Customer Account Status Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC . . $990/$30

(T) **PAYROLL** – Prepares payroll for hourly, salaried and commissioned employees. Generates monthly, quarterly and annual returns. Prepares employee W-2's. Includes tables for federal withholding and FICA as well as withholding for all 50 states plus up to 20 cities from pre-computed or user generated tables. Will print checks, Payroll Register, Monthly Summary and Unemployment Tax Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC . . . . $990/$30

(T) **INVENTORY** – Maintains detailed information on each inventory item including part number, description, unit of measure, vendor and reorder data, item activity and complete information on current item costs, pricing and sales. Produces reports as follows: Physical Inventory Worksheet, Inventory Price List, Departmental Summary Report, Inventory Status Report, The Reorder Report and the Period-to-Date and Year-to-Date reports. Supplied in source code for Microsoft BASIC . . . . $1,190/$30

(T) **MAILING ADDRESS** – Keeps track of name and address information and allows the selective printing of this information in the form of mailing lists or address labels. Allows the user to tailor the system to his own particular requirements. User-defined format and print-out system uses a special format file which tells programs how to print the mailing list or address labels. Standard format files are included with system. Automatic sorting of data uses indexed file management routines which allow the name and address information to be sequentially retrieved and printed without file sorting. Supplied in source code for Microsoft BASIC . . . . $790/$30

• GRAHAM-DORIAN SOFTWARE SYSTEMS

(T) **GENERAL LEDGER** – An on-line system; no batching is required. Entries to other GRAHAM-DORIAN accounting packages are automatically posted. User establishes customized C.O.A. Provides transaction register, record of journal entries, trial balances and monthly closings. Keeps 14 month history and provides comparison of current year with previous year. Requires CBASIC-2. Supplied in source . . . $995/$35

(T) **ACCOUNTS PAYABLE** – Maintains vendor list and check register. Performs cash flow analysis. Flexible — writes checks to specific vendor for certain invoices or can make partial payments. Automatically posts to GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source . . . . $995/$35

(T) **ACCOUNTS RECEIVABLE** – Creates trial balance reports, prepares statements, ages accounts and records invoices. Provides complete information describing customer payment activity. Receipts can be posted to different ledger accounts. Entries automatically update GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source . . . . $995/$35

(T) **PAYROLL SYSTEM** – Maintains employee master file. Computes payroll withholding for FICA, Federal and State taxes. Prints payroll register, checks, quarterly reports and W-2 forms. Can generate ad hoc reports and employee form letters with mail labels. Requires CBASIC-2. Supplied in source . . . . . . $590/$35

(T) **INVENTORY SYSTEM** – Captures stock levels, costs, sources, sales, ages, turnover, markup, etc. Transaction information may be entered for reporting by salesman, type of sale, date of sale, etc. Reports available both for accounting and decision making. Requires CBASIC-2. Supplied in source . . . $690/$35

(T) **JOB COSTING** – Designed for general contractors. To be used interactively with other GRAHAM-DORIAN accounting packages for tracking and analysing expenses. User establishes customized cost categories and job phases. Permits comparison of actual versus estimated costs. Automatically updates GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source $995/$35

Sample Program Disk For Each Graham-Dorian Business Package. Specify Package $45

The sale of each proprietary software package conveys a license for use on one system only.

Prices F.O.B. Fort Worth, Tex. Shipping, handling and C.O.D. charges extra.

• MICROSOFT PRODUCTS

(M) **BASIC-80** – Disk Extended BASIC, ANSI compatible with long variable names, WHILE/WEND, chaining, variable length file records . . . . $350/$25

(M) **BASIC COMPILER** – Language compatible with BASIC-80 and 3-10 times faster execution. Produces standard Microsoft relocatable binary output. Includes MACRO-80. Also linkable to FORTRAN-80 or COBOL-80 code modules . . . . $395/$25

(M) **FORTRAN-80** – ANSI 66 (except for COMPLEX) plus many extensions. Includes relocatable object compiler, linking loader, library with manager. Also includes MACRO-80 (see below) . . . . $500/$25

(M) **COBOL-80** – Level 1 ANSI '74 standard COBOL plus most of Level 2. Full sequential, relative, and indexed file support with variable file names. STRING, UNSTRING, COMPUTE, VARYING/UNTIL, EXTEND, CALL, COPY, SEARCH, 3-dimensional arrays, compound and abbreviated conditions, nested IF. Powerful interactive screen-handling extensions. Includes compatible assembler, linking loader, and relocatable library manager as described under MACRO-80 . . . . $750/$25

(M) **MACRO-80** – 8080/Z80 Macro Assembler. Intel and Zilog mnemonics supported. Relocatable linkable output. Loader, Library Manager and Cross Reference List utilities included . . . . $150/$25

(T) **XMACRO-86** – 8086 cross assembler. All Macro and utility features of MACRO-80 package. Mnemonics slightly modified from Intel ASM86. Compatibility data sheet available . . . . $300/$25

(T) **PASCAL/M*** – Compiler generates P code from extended language, implementation of standard PASCAL. Supports overlay structure through additional procedure calls and the SEGMENT procedure type. Provides convenient string handling capability with the added variable type STRING. Untyped files allow memory image I/O. Requires 56K CP/M . . . $150/$20

(T) **PASCAL/Z** – Z80 native code PASCAL compiler. Produces optimized, ROMable re-entrant code. All interfacing to CP/M is through the support library. The package includes compiler, Microsoft Compatible relocating assembler and linker, and source for all library modules. Variant records, strings and direct I/O are supported. Requires 56K CP/M and Z80 CPU. . . . . $395/$25

(M) **PASCAL/MT** – Subset of standard PASCAL. Generates ROMable 8080 machine code. Symbolic debugger included. Supports interrupt procedures, CP/M file I/O and assembly language interface. Real variables can be BCD, software floating point, or AMD 9511 hardware floating point. Version 3 includes Enumeration and Record data types. Manual explains BASIC to PASCAL conversion. Source for the runtime package requires Digital Research's MAC. Requires 32K . . . . $250/$30

(M) **CBASIC-2** Disk Extended BASIC – Non-interactive BASIC with pseudo-code compiler and run-time interpreter. Supports full file control, chaining, integer and extended precision variables, etc. . . . . $110/$15

(M) **BSTAM** – Utility to link one computer to another also equipped with BSTAM. Allows file transfers at full data speed (no conversion to hex), with CRC block control check for very reliable error detection and automatic retry. We use it! It's great! Full wildcard expansion to send * COM, etc. 9600 baud with wire. 300 baud with phone connection. Both ends need one. Standard and (M) versions can talk to one another. . . . . $150/$5

(M) (T) **SELECTOR III-C2** – Data Base Processor to create and maintain multi Key data bases. Prints formatted sorted reports with numerical summaries or mailing labels. Comes with sample applications, including Sales Activity, Inventory, Payables, Receivables, Check Register, and Client/Patient Appointments, etc. Requires CBASIC-2. Supplied in source . . $349/$20

GLECTOR – General Ledger option to SELECTOR III-C2. Interactive system provides for customized COA. Unique chart of transaction types insure proper double entry bookkeeping. Generates balance sheets, P&L statements and journals. Two year record allows for statement of changes in financial position report. Supplied in source. Requires SELECTOR III-C2, CBASIC-2 and 52K system . . . . $250/$25

(M) **TEXTWRITER III** – Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Has facilities for sorted index, table of contents and footnote insertions. Ideal for contracts, manuals, etc. Now compatible with Electric Pencil* prepared files. . . . . $125/$20

ORDERS MUST SPECIFY DISK SYSTEMS AND FORMATS:

FORMATS AVAILABLE:
(A) TRS-80 Model I (M) Keys Only
(B) TRS-80 Model II
(C) TRS-80 Model III (M) Keys Only
(D) HEATHKIT H89 (M) Keys Only
(E) NORTH STAR
(F) SUPER BRAIN QD
(G) STANDARD UNIMPLEMENTED

(M) Modified version available for use with CPM as implemented on Heath and TRS-80 Model I computers.

(T) For all (T) items listed above, the recommended system configuration consists of 48K CPM 2 full dize disk drives. 24 x 80 CRT and 132 column printer.

FMG's LIBRARY:

PASCAL USER MANUAL & REPORT
(2nd) Edition by K. Jensen and N. Wirth
- Tutorial Manual and Concise Reference Report for Both Programmers and Implementors
- Includes Helpful Examples to Demonstrate the Various Features of PASCAL

The book consists of two parts: the user manual and the revised report. The manual is directed to those who have some familiarity with computer programming and who wish to get acquainted with the PASCAL language. The report defines standard PASCAL which constitutes a common base between various implementations of the language

Stock No. #B21 — Price $9.95

Microcomputer Problem Solving Using Pascal
by Kenneth L. Bowles
- A Book Designed for Both College Courses AND Individual Self-Study
- Ideal for use with UCSD Pascal
- Includes Extensions to Standard PASCAL

This book is designed both for introductory courses in computer problem solving at the freshman and sophomore college level, and for individual self-study. It includes many examples and actually executable programs. It includes information on the necessary functions and procedures for handling graphics and strings

Stock No. #B22 — Price $14.95

PROGRAMING IN PASCAL
by Peter Grogono
- An Excellent Introduction to One of the Fastest Growing Programming Languages Today
- Sections on Procedures and Files PLUS a Chapter on Dynamic Data Structures such as Trees and Linked Lists

The text is arranged as a tutorial, containing both examples and exercises to increase reader proficiency in PASCAL. Concepts are illustrated by examples, ranging from the Tower of Hanoi problem to circumscribing a circle about a triangle. PROGRAMMING IN PASCAL is sure to hold the reader's interest

Stock No. #B23 — Price $14.95

PASCAL PRIMER Problem Solving

This book has three major goals:
- To introduce all aspects of the programming and problem solving process (includes problem specification and organization, algorithms, coding, debugging, testing, documentation and maintenance);
- To teach good programming style and how to produce a high quality finished product; and
- To teach the syntax of the PASCAL programming language.

Numerous examples are employed throughout the text. PASCAL is used as a vehicle to teach various aspects of programming techniques

Stock No. #B24 — Price $18.95

BEGINNER'S MANUAL FOR UCSD PASCAL SYSTEM
- An Enlightening Introduction to UCSD PASCAL
- Demonstrates How to Use the UCSD PASCAL System and How to Program in PASCAL
- Includes Many Practical Examples of PASCAL Programs

This book is intended to be used as an introduction and reference manual for people just beginning to use the UCSD Pascal Software System. Whether you have never used a computer before or whether you are an experienced programmer who is unfamiliar with UCSD PASCAL, this book will provide a relatively easy yet thorough introduction to UCSD PASCAL.

Stock No. #B25 — Price $11.95

UCSD Reference Book
- A Reference Guide to the Complete UCSD PASCAL System
- Includes Information on Compiler Basic, Assembler and Editor
- Lists Actual P-Machine Codes

This reference book can be a valuable and time-saving guide to thorough information on the UCSD PASCAL system. The easy-to-read manual provides fast access to pertinent data.

Stock No. #B26 — Price $25.00

✓See List of Advertisers on page 306

A BASIC to FORTRAN conversion utility of limited effectiveness.

FORTRANslator

FORTRANslator
The Management
Aledo, TX
$29.95

Richard C. McGarvey
221 Hirschfield Drive
Williamsville, NY 14221

I am going to break some writing rules and start this review with a warning rather than a catchy introduction: If you are not a capable FORTRAN programmer or at least a very dedicated learner, skip this article and the FORTRANslator utility. However, if you are a dedicated learner with a good supply of FORTRAN manuals and a good printer you might be interested.

The reason for this warning is that the FORTRANslator utility, despite what you might expect, does not really translate BASIC to FORTRAN. In fact, the resulting translation is totally useless to a programmer without a good FORTRAN vocabulary.

I read about the FORTRANslator utility in the New Products section of *80 Microcomputing*, and it sounded too good to be true. The description promised that a program could be written and debugged in BASIC and then translated to FORTRAN. Most of my programs are written in BASIC, and I wanted to learn how to rewrite them in FORTRAN, but the work involved was prohibitive. It seemed as though FORTRANslator would be the answer.

The program consists of one disk with the BATRAN/CMD program and one page of documentation. I rushed to the computer, whipped out some short BASIC programs and started to work.

The Small Problems

I always expect problems with new programs and I was not disappointed this time. It appeared that the one page letter included all the instructions I needed to operate the program. It also contained a list of what the translator did and did not do. It was certainly straight-forward.

The first step is to save the BASIC program in ASCII format (SAVE"TEST/BAS",A). Then go to DOS, run BATRAN/CMD and answer the questions that the translator asks. They are: SAVE ON DISK?—If YES—ENTER FILESPEC. PRINTER ON LINE?—an answer of YES, produces a hard copy of the translation. From this point on the translator takes over and each ASCII line is supposed to be evaluated and placed into FORTRAN format. The result should be close to FORTRAN, but here is where the problems start.

```
00100 C    Convert Fahrenheit to Centigrade
00200      Integer F
00300      Write(5,5)
00400 5    Format(33H Fahrenheit Centigrade)
00500      DO 20 F=20,65,5
00600      C=5./9.*(F-32)
00700      Write(5,10)F,C
00800 10   Format(12X,I2,11X,F6.3)
00900 20   Continue
01000      End
```

Figure 1.

```
10 REM CONVERT FAHRENHEIT TO CENTIGRADE
20 DEFINT F
30 PRINTTAB(5)"FAHRENHEIT    CENTIGRADE
40 FORF=20TO65STEP5
50 C=(5/9)*(F-32)
60 PRINTTAB(8)F;"          ";C
70 NEXTF
80 END
```

Figure 2. Basic Version

The instructions indicate what kinds of problems may be encountered, however, they give no suggestions as to how to solve them. Of course, if you don't know FORTRAN already, there is no chance in hell that you will be able to solve them. If the author of the translator utility had taken the time to work out the more common problems, (such as string handling) and include the solutions as examples, the beginner might have a chance. It also would have helped if the documentation described what you could expect as a result of translation.

So, documentation is a problem which wouldn't be too bad if that was all, but it's not. There are problems with the printer output. The resulting translation requires extensive editing and a line printer is a must. There was a warning to that effect in the *80* ad as well as in the documentation. They aren't kidding. FORTRAN is longer than BASIC and much more particular about the format of instruction lines. A printout on which you can edit before you actually edit the program is indispensable.

The printout procedure has some flaws. The first part of the printout is a cross reference of BASIC calls that will have to be altered. It is helpful, but when there is no output the heading is printed and a whole page is wasted.

The second page is a printout of the arrays used, broken down by GOTO array and subroutine arrays. This is also necessary since FORTRAN does not use subroutines as BASIC does, and they must all be moved to the end of the FORTRAN program. Again, when there is a short list or no list, the page is headed and wasted.

The third part of the printout is the FORTRAN program. It is neat and clean and there is plenty of room in which to write editing notes. There is no problem here—at least, there is no problem with the printout.However, the content is another story.

Difficulties with Translation

OK! So I've been a little picky so far. It gets worse. To start, I will use the FORTRAN do loop as an example of problems that occur in the translation. The program claims to produce do loops; seeing the problems that exist with them gives a clear view of the troubles that occur with FORTRAN conventions that are not produced by the translator.

A do loop is the FORTRAN version of a BASIC for-next loop and is used in the same way. Here is an example of the do loop, the for-next loop and the do loop that is produced by the translator.

Proper Do Loop	For Next Loop	Translator Do Loop
Do 10 J = 1, 1000	ForJ = TO1000	DoJ = TO1000
Step1	Step1	Step1
Step2	Step2	Step2
.	.	.
.	.	.
.	.	.
10 Continue	NextJ	ContinueJ

In the correct do loop, do 10 means "Do the steps between the do and the continue labeled 10." The variable J will equal the values 1 to 1000, as in a for/next loop. Arrays are used between the do/continue as array(J), which will increment from J = 1 to J = 1000 as the loop progresses.

The translator do loop is not correct! It must be edited to correct format. Also, the correct continue must be labeled. In fact, all that the translator does is change the word "for" to "do", the word "next" to "continue", and add the variable name, resulting in continueJ. If the original for/next loop has a subroutine branch nested in the loop, it will be lost and the logical flow of the program must be adjusted or the FORTRAN program will not run correctly.

This occurs throughout the translation. BASIC input PRINT, LPRINT, PUT and GET are all handled by FORTRAN's READ and WRITE. Each READ or WRITE statement is followed by the name of the device to be read or written to and the label of the format line that tells FORTRAN what to READ or WRITE and how. The form is READ(d,1) where d is the assigned device number (ie., 2 for printer, 5 for keyboard and video, 6 for disk) and 1 is the label of the format line to be used. The translator does not assign the correct device number but does assign a format line label. The format line will have to be corrected for the type of data and device. This is difficult if you don't know exactly which kind of data to expect. Also, LPRINTS will be translated as LWRITE(d,1) and have to be edited. I imagine that the L was left on so the programmer knows that when editing, the destination device intended was the printer. Without the correct device number in the WRITE statement there would be no way to know which device was intended without checking program logic. Disk READ and WRITE are useless as they come out of the translator and must be completely reworked and formatted.

Another important consideration is compatibility with the FORTRAN compiler. The translator produces text in FORTRAN form. It inserts C lines (REM in BASIC) and it prints in an easy to read form with room for manual editing on the page. The disk output is the same text that is on the printer—*it will not work in the TRS-80 (Microsoft) FORTRAN compiler.*

In FORTRAN, programming is done in columns and each column or group of columns has a specific use. When a program is translated, loaded into the TRS-80 FORTRAN editor and edited into proper FORTRAN syntax, the compiler will still generate errors for every C line, every label line and every statement line. This is due to improper column arrangement in the translation and requires more editing, and spacing the lines so that columns are properly aligned.

```
100  C
200  C——————————————
300  C Name:
400  C Purpose:
500  C Calling Sequence:
600  C Input Assignments:
700  C Output Arguments:
800  C Description:
900  C Devices:
1000 C Notes:
1100 C
1200 C——————————————
1300 C
1400 C Undefined Variables:
       I,J,K,L,M = Integer. Others Real
1500 C
1600   Integer
1700 C
1800   Real
1900 C
2000   Double Precision
2100 C
2200   Logical
2300 C
2400   External
2500 C
2600   Dimension
2700 C
2800   Common
2900 C
3000   Equivilance
3100 C
3200   Data
3300 C
3400 C REM Convert Fahrenheit to Centigrade
3500        Defint F
3600 C
3700        Write(3, 110)
3800   110    Format (PC, "Fahrenheit   Centigrade)
3900 C
4000        DOF = 20T065,5
4100        C = (5/9)*(F - 32)
4200 C
4300        Write(3, 111) ";C
4400   111    Format (PC,"    )
4500 C
4600        ContinueF
4700 End
```

Figure 3. Translator Version

Note the program in Fig. 1. This is an example program from the Radio Shack (Microsoft) FORTRAN package. It is correctly columned and can be compiled with no errors. This program generates the centigrade equivalent for fahrenheit values from 20 to 65 with increments of five degrees. Fig. 2 is a BASIC program that will output the data in the FORTRAN example right down to the format of video display. Now look at Fig. 3; this is the translation of the BASIC program in Fig. 2. Note that the format of the line numbers are different from that in the FORTRAN editor. Also, the (comment) lines are not spaced the same. This makes the entire translation incompatible with the FORTRAN editor.

Not only is the format wrong but the statement lines are error-filled. Note line 4100 in Fig. 3. This is the formula line for converting fahrenheit to centigrade. The translator has duplicated the formula exactly as it appears in BASIC. FORTRAN formulae are different! Look at Fig. 1, line 600. That is correct FORTRAN! Now look at line 4400 in Fig. 3. This is the format line for the data output. Nothing there! If you look at Fig. 1, line 800 you will see correct formatting. It is easy to see that a programmer who is not familiar with FORTRAN would have no use for this translation.

There are many more problems. Text lines are cut off, write lines become doubled with text format lines, and there are still more. Even if we assume that you can get a working program out of the translator—there is still one last warning to consider. The translator does not approach efficient use of FORTRAN. Microsoft FORTRAN is about 60 times faster than Disk BASIC. This estimate is based on a program I wrote to check the time difference. The benefit of faster FORTRAN is lost if you do not use it efficiently.

Not All Bad

The translator does have some value. It does aid in reducing typing time. And it provides a FORTRAN format for the BASIC program that is a skeleton starting point. It aids in following the logic of a BASIC program that you are rewriting. Also, it is helpful if you are learning FORTRAN (have some background first or you will be lost).

One final word. I have only pointed out the major problems that occurred in the translation of short BASIC programs. You can see that the problems are extensive and would be insurmountable to a programmer without FORTRAN knowledge. I shudder to think what problems may occur in the translation of an extensive BASIC program. If you write the original BASIC program carefully, some of the translation problems I have mentioned may be eliminated. ■

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

PAYROLL

Payroll invoices many complex calculations and the production of reports and documents, many of which are required by government agencies. It is an ideal candidate for the computer. With this Payroll system in-house, you can promptly and accurately pay your employees and generate accruate documents/reports to management, employees, and appropriate government agencies concerning earnings, taxes, and other deductions. The package has been converted to the TRS-80™ and is now a well documented, op-line, interactive, micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ performs all necessary payroll tasks including:
- file maintenance, pay data entry and verification
- computation of pay and deduction amounts
- printing of reports and checks

★ can handle salaried and hourly employees

★ employees can receive:
- hourly or salary wage
- vacation pay
- holiday pay
- piecework pay
- overtime pay

★ employees can be paid using any combination of pay types (except, hourly cannot receive salary and salary cannot receive hourly)

★ special non-taxable or taxable lump sums can be paid regularly or one time (bonus, reimbursements, etc)

★ health and welfare deductions can be automatically calculated for each employee

★ earnings-to-date are accumulated and added to permanent records; taxes are computed and deducted: US income tax, Social Security tax, state income tax, other deductions (regular or one time)

★ paychecks are printed; computer checks with your company letterhead can be purchased from SBSG

★ calculations are accumulated for; employee pay history, 941A report, W-2 report, insurance report, absentee report

★ fully linked to General Ledger. Each employee's payroll information can be distributed to as many as (12) twelve different GL accounts; system automatically posts to cash account

INVENTORY CONTROL/INVOICING

★ **ISAM** (Indexed Sequential Access Method) eliminates the necessity for time consuming sort.

★ Pre-Allocated Files for IMMEDIATE update and inquiry capabilities.

★ Fast Disk storage and retrieval.

★ Inventory Master Record includes...class...SKU...Division...Retail...Cost...Beginning Balance...Period Sale Units...Period Receipts...On Order...On Hand...Minimum Reorder Point...Recommended Reorder Amount...Vendor Number...Period Sale Dollars...YTD Sale Units...YTD Sale Dollars.

★ Calculated and Displayed Formulas include...Gross Margin ($)...Gross Margin (%)...Gross Margin ROI (%)...Average Inventory Retail ($)...Average Inventory Cost ($)...Turn-Over (%).

★ Reports Generated include...Master File Listing...Class Description Listing...Transaction Audit Trail...Minimum Reorder Point by Vendor...Retail Price List...Retail & Cost Price List...Period Sales Report ...Year to Date Sales Report...Stock Status (Screen or printer output) ...Commission Report (for salesmen and buyers).

★ Transaction Types include...Sales, Vendor Receipts...Vendor Orders...Customer Returns...Vendor Returns...Transfer Stock.

GENERAL LEDGER

The General Ledger accounting system consolidates financial data from other accounting subsystems (A/R, A/P, Payroll, direct posting) in an accurate and timely manner. Major reports include the Income Statement and Balance Sheet and a "special" report designed by management. The beauty of this General Ledger system is that it is completely user formatted. You "customize" the account numbers, descriptions, and report formats to suit particular business requirements. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now a well documented, online, interactive micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ more than 200 chart of accounts can be handled

★ account number structure is user defined and controlled

★ more than 1,750 transactions may be entered via:
- direct posting; done by hand; validated against the account file before acceptance
- external posting; generated by A/R, A/P, Payroll or any other user source

★ data is maintained and reported by:
- month
- quarter
- year
- previous three quarters

★ reports (samples on back) include:
- trial balances
- income statement
- balance sheet
- special accounts reports and more.....

★ user formats reports with the following designated as you wish:
- titles
- headings
- account numbers
- descriptions
- subtotals
- totals
- skip lines
- skip pages

★ up to eight levels of totals - fully user designated

★ menu driven; easy to use; full screen prompting and cursor control

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

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

24 HOUR ORDER LINE
(914) 425-1535

master charge · VISA

PRICING	MOD-1 VERSION	MOD-II VERSION	MOD-III VERSION
ACCOUNTS RECEIVABLE	$125	$225	$199.95
ACCOUNTS PAYABLE	$125	$225	$199.95
GENERAL LEDGER	$125	$225	$199.95
PAYROLL	$125	$225	$199.95
INVENTORY	$175	$275	$199.95
INVOICING	$150	$250	$199.95
COORDINATED INVENTORY/INVOICING ACCOUNTS RECEIVABLE	$449	$749	$599.95
COORDINATED AR-AP-GL	$375	$675	$599.95
COORDINATED AR-AP-GL with PAYROLL	$495	$899	$799.95
EXTENDED COORDINATED AR-AP-GL INVOICING/INVENTORY with PAYROLL	$799	$1299	$1199.95

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

MICROSOFT BASIC COMPILER

With TRS-80™ BASIC Compiler, your Level II programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC interpreter are included in the package. Features like the WHILE...WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution. You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80™ Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation **$195.00**

1980 INCOME TAX PAC

Completely Revised - Latest Tax Tables - Fully Tested - Complete Manual and Documentation. The new version of the Income Tax Pacs are full of error catching codes making it impossible to make an error. Follow the simple Step By Step procedure that makes tax preparation simple.

INCOME TAX PAC A .... **($19.95...Cassette)**
For Level II 16K Cassette Only
Does Form 1040 and 1040A
- Schedule A itemized deductions
- Schedule B interest and dividends
- Output to video display
- Schedule TC tax computation

INCOME TAX PAC B .... **$49.95...Cassette or Diskette)**
For Level II 16K with or without printer...cassette or disk has all features of Income Tax Pac A **Plus** works with or without line printer.
- Formats Form 1040 and 1040A for standard tax forms
- Schedule C income from a personally owned business
- Form 2106 employee business expense

PROFESSIONAL INCOME TAX PAC C .... **$99.95...Diskette**
For Level II 32K with disk and printer (optional)
Has all features of Income Tax Pac B **Plus** automatic memory storage for income tax preparers.
- 22 additional schedules and forms
- Formats forms for individual or tractor feed printing

MOD II CPA VERSION **$199.95**

GUARANTEED PROFIT 91% WINS PLACES SHOWS 32% AVERAGE PROFIT AT ALL TRACKS-1978

THE HORSE SELECTOR II (FLATS) (By Dr. Hal Davis **$50.00**
New simplified version of the original Horse Selector. The first Horse Selection System to actually calculate the estimated odds of each horse.
HIGHER PROFITS (OVER 100%) POSSIBLE THROUGH SELECTIVE BETTING ON:
- Rates each horse in 10 seconds.
- Easy to follow rules.
- Can be used with any Apple II Computer.
- 100% money back guarantee (returned for any reason).
- Uses 4 factors (speed rating, track variant, distance of the present race, distance of the last race).
- Using the above factors, the Horse Selector calculates the estimated odds. BET on horses whose actual payoff (from the Tote Board or Morning Lines) is higher than payoff based on estimated odds.
- Using the above factors, the Horse Selector calculates the estimated odds. BET on any selected horse with an estimated payoff (based on Tote Board or Morning Lines) higher than calculated payoff (based on Horse Selector II).
- Source listing for the TRS-80™, TI-59, HP-67, HP-41, Apple and BASIC Computers.
- No computer or calculator necessary (although a calculator would be helpful for the simple division used to calculate estimated odds).

FREE Dutching Tables allows betting on 2 or more horses with a guaranteed profit.

NEWDOS/80

A **New** enhanced NEWDOS for TRS-80™ Model I for the 1980's

Apparat Inc., announces the most powerful Disk Operating System for the TRS-80™. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2. 1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2. 1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2. 1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:
- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disk drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOSREADY" or "READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST."
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful program chaining.
- Device hanging for routing to display and printer simultaneously.
- CDE function; simultaneous striking of the C, D and E keys will allow user to enter a mini-DOS to perform some DOS commands without disturbing the resident program.
- Upward compatible with NEWDOS 2. 1 and TRSDOS 2.3.
- Includes Superzap 3.0 and all Apparat 2.1 utilities.

.... **$149.00**

STOCK MARKET MONITOR

Galactic Software Ltd.

CASSETTE VERSION **$89.00**
DISK VERSION **$99.00**

1. The system is designed for the active "trader" not the "long term" investor, as the system is "technically" oriented.
2. For the TRS-80™ Model I, Level II, 16K or more. Available in both disk and tape versions.
3. Tracks user selected issues, in a technical system that reflects the issue's performance against the overall market.
4. Set up data is input by the user from the Standard and Poors stock guide or Value Line.
5. Daily issue data, "high", "low", "close" and "volume" are input from any newspaper containing this information.
6. Daily overall market, "volume" and "closing Dow" are also provided from a newspaper.
7. Volume and price changes of an issue, as they compare to volume an price changes of the overall market, are the basis of this system's analysis of the given issue.
8. Comparisons of the issue against itself are also done. This may allow the user to spot "unusual" activity on this issue.
9. Clear indications are given as to whether the issue is "out performing", "under performing" or "performing" with the market.
10. Complete video and printed output is provided.
11. This program is intended to be a guide to indications, and is not to be used as a sole recommendation to buy, sell or hold an issue. These decisions are the responsibility of the user and his brokerage.

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

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

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software

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

Part 1.

An Idiot's Guide To Assembly Language

Robert C. Montgomery
67 Turtle Back Rd. West
New Canaan, CT 06840

There are some excellent books in circulation on assembly language programming; magazines carry helpful articles for the novice as well as for the pro; Radio Shack offers its Editor/Assembler and T-Bug with elaborate documentation. Surely, with all that help available, even an idiot can become fluent in assembly language.

Well, not *this* idiot! I have managed to gain a little skill in this area, but I did it through trial and error. I can do some things in assembler which are useful to me, but more importantly, I've reached the point at which I can understand the books and go on to develop real proficiency.

If you're having trouble getting over the threshhold, stick with me through this two-part article. We're going to go from square one all the way to square two, step by step. Get a copy of Radio Shack's Editor/Assembler, sit down at your TRS-80, and read on.

Why Bother?

It isn't an idle question. Level II BASIC is an excellent language; you can program anything in it, and you can program faster than in assembler, with less mindbenders to deal with along the way. There is really only one reason for working in assembly language: The machine code which assembly language produces executes as much as a thousand times faster! It also requires less memory for program storage, and it puts you one up on your friends. But speed is the only real reason for going to all the trouble I'm about to describe to you.

How important is speed? It can be *very* important if we're tracking incoming hostiles, or even crunching millions of numbers for the Controller's Department. In personal computing applications, however, it's most commonly needed in games. The user can drop dead of boredom while the machine is conducting a lengthy tree-search, or putting elaborate graphics on the screen, if such things are done in BASIC.

Suppose that we want to program a football game. We'll need a playing field. We plan a scale of one horizontal tab space to two yards. We'll want two double-width horizontal lines, 50 yards apart, for the sidelines; two double-width vertical lines at each end for the end zones; and nine vertical stripes for the yard markers. We'll make the 50-yard line double-width, and the other yardlines single width, as shown in Fig. 1, and we'll place the upper left corner of the field at screen position 384, so the top of the screen can be used for a scoreboard and the game controls.

Remembering that the goal lines are in the end zones, and that the sidelines are out of bounds, we might write the BASIC program shown in Fig. 2 to produce this result. Try it. You'll be amazed to find that the TRS-80 can do all that in only 3.4 seconds. But now visualize this as a subroutine in the game you're planning. You'll want to erase the entire screen for post-touchdown fireworks and to display game statistics on demand.

The user playing games on his 80 doesn't want to sit around watching graphics get drawn; he wants to get on with the job of clobbering his opponent. Will assembly language improve this? You better believe it. Before we're done, we're going to draw a complete football field so fast that you can't see it happen.

Using EDTASM

Begin by getting familiar with the operation of the Radio Shack Editor/Assembler: Load the tape as the manual instructs, using the name EDTASM after the System command. When the asterisks stop blinking, respond /, and TRS-80 EDITOR/ASSEMBLER 1.2 will appear. You'll then see an asterisk. That is EDTASM's prompt, equivalent to the greater than sign in BASIC. Now look at the Commands sections of the manual. Here are the commanual. Here are the commands which will be important to use at this stage:

- A Assemble source currently in buffer: Try A, and see what happens. It says NO TEXT IN BUFFER. Well, that makes sense. We just turned it on. But what is "source", what is "text", and what is a "buffer"?

EDTASM is a program which permits you to write in a language called assembler, to edit and correct your work, and then to record it on tape. It also translates your assembly language program into machine language, and permits you to record that as well.

The assembly language program is also called the text or source code, and the machine language program is also called the object code. The process of translation from text to machine language is called assembly.

The buffer is a segment of TRS-80 memory which holds the text while you are entering and editing it. When you press A, the text is assembled into machine code which the TRS-80 can read through the SYSTEM command and which, with a little bit of luck, it will execute to draw our football field. Both the text and the machine code are stored in memory while EDTASM is in oper-

✓See List of Advertisers on page 306

ation.

● B Return to BASIC in ROM: Don't do it unless you mean it. If you do, you'll have to reload EDTASM again, and both versions of the program you're working on will be erased. The NEW command in BASIC is very similar.

● I Insert a source line: You have to tell the computer where to insert it. I100, for example, tells it to insert it at line 100. I also has the features of AUTO in BASIC. I100 provides lines starting at 100 with the default increment of 10. I100,50 starts at 100, with increments of 50.

● E Edit a specified command: What they mean is, edit a specified line. E100 permits you to edit line 100. The subcommands—I, H, C, and others—are nearly identical to those you use in BASIC.

● R Replace line currently in buffer: R100 wipes out the text in line 100. The line number remains, and you can proceed to enter new text from the beginning of that line.

● W,L Write (W) or Load (L) a text file to or from tape: CSAVE or CLOAD, in other words. It's the text that is loaded and saved and not the machine code.

● H,P Print a specified range of source code to line printer (H) or screen (P): LLIST and LIST, in other words. But you can't simply use P or H, except to list the current line only. You must specify which line you want, using P100:500, for example. Why a colon instead of a comma, as required in I? Who knows?

● Horizontal tab: Works just as it does in BASIC, but it's important in assembler. As you'll see, elements of a line of text are separated by their tab positions.

Unless you're a lot smarter than I, you have some unanswered questions. For example:

1. How do you stop entering lines? You use break. In fact, you almost always use break when you want to stop what you're doing and start something else.
2. How do you exit from the edit mode? Press enter. You can also use E, as in BASIC, but why bother?
3. How do you start entering lines? You use I, even though there are no other lines to insert the new one between. Use I100, for example, if you want the first line to be line 100.
4. How do you run the program you've written? With EDTASM in use, you can't. EDTASM simply creates the machine code which the computer can later execute, and the text can not be run under any circumstances. In order to run your program, you must:
   a. Record the machine code on cassette.
   b. Exit EDTASM (using B, or the off-on switch).
   c. Load the machine language tape into the TRS-80, using the System command.
   d. Respond to the *- prompt with /, as you did to run EDTASM.
5. How do you record the machine code? There's no command for that purpose. When you use A to assemble the text into machine code, the screen will display READY CASSETTE. Get a cassette ready to record, then hit any key. If you don't want to record after assembly, press break.
6. System requires a program name. How do you give your program a name? Instead of using A, use A MYPROG, or whatever you want to call it. If you use A

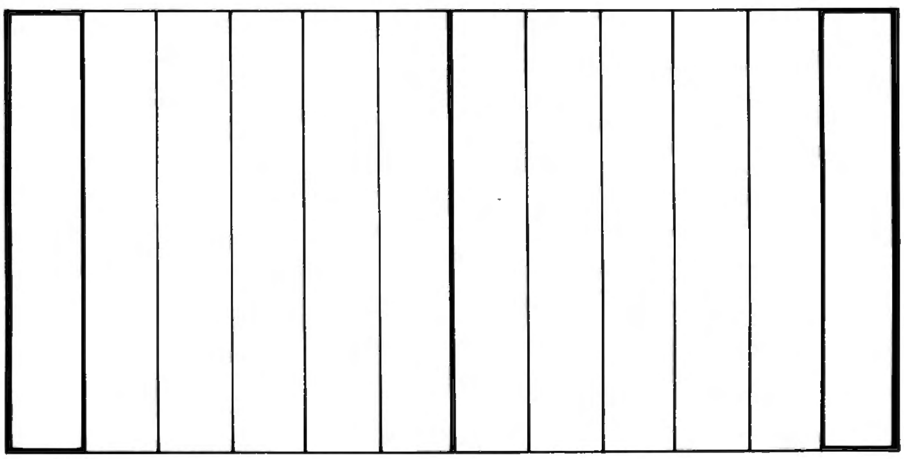


Fig. 1

✓See List of Advertisers on page 306

```
1000 P=15360 : CLS
1010 FOR I=384 TO 444 : POKE P+I,131 : NEXT
1020 FOR I=896 TO 957 : POKE P+I,176 : NEXT
1030 FOR I=384 TO 896 STEP 64 : POKE P+I,191 : POKE P+I+5,191 :
POKE P+I+56,191 : POKE P+I+61,191 : NEXT
1040 FOR I=394 TO 414 STEP 5
1050 FOR J=0 TO 512 STEP 64
1060 IF J=0 POKE P+I,171 ELSE IF J=512 POKE P+I+J,186 ELSE POKE
P+I+J,170
1070 NEXT J,I
1080 FOR I=415 TO 435 STEP 5
1090 FOR J=0 TO 512 STEP 64
1100 IF J=0 POKE P+I,151 ELSE IF J=512 POKE P+I+J,181 ELSE POKE
P+I+J,149
1110 NEXT J,I
```

Notes: Time requirement 3.4 seconds
Memory requirement 378 bytes
Speed can be improved slightly and memory significantly by compression and other optimization.

Fig. 2. Drawing a Football Field in BASIC

alone, the program will have the default name NONAME. Why a space, instead of a comma or a colon? Who knows?

7. Suppose the machine code needs correction. What do you do then? You must:
   a. Reload EDTASM.
   b. Reload the text (using L).
   c. Edit.
   d. Re-assemble and re-record text and machine code, because this is not the last time you'll have to edit.
   e. Exit EDTASM.
   f. Load the machine code, using SYSTEM.
   g. Run it, using /, discover more errors, and reload EDTASM to begin again.

8. Isn't that a lot of work? Yes, it is. It is a very awkward system. In fairness to Radio Shack, the companion T-Bug program permits you to run, edit and re-run without all of these saves and loads, although you'll have to edit the machine code directly without the help of assembly language.

Hex = 5A9E

```
5 =  5 × 4096 = 20480
A = 10 ×  256 =  2560
9 =  9 ×   16 =   144
E = 14 ×    1 =    14
                23198
```

Decimal = 23198

Fig. 3. Hex to Decimal Conversion

Decimal = 23198

```
23198 / 4096 =  5 (carry 2718)    5 = 5
 2718 /  256 = 10 (carry  158)   10 = A
  158 /   16 =  9 (carry   14)    9 = 9
   14 /    1 = 14                14 = E
```

Hex = 5A9E

Fig. 4. Decimal to Hex Conversion

Hexadecimal Numbers

Hex is a base 16 numbering system, whereas decimal is base 10, and binary is base 2. Hex and decimal numbers are the same from zero to nine. Decimal 10 is hex A, decimal 11 is hex B, and so on through decimal 15 (hex F).

We will deal only with positive integers, in the range of 0 through 32767, which is 0 through 7FFF in hex. We need to know how to convert numbers in this range from hex to decimal and back again. One way is to buy a specialized calculator, which makes these conversions automatically. Doing it by hand is merely time-consuming and annoying—not really difficult.

To convert from hex to decimal, simply multiply the right-hand digit by 1, the next digit by 16, the next by 256, and the left-hand digit by 4096. Add the results, as shown in Fig. 3. From decimal to hex, reverse the process, as shown in Fig. 4.

When you're using EDTASM, hex numbers must have the suffix H (as in 5A9EH), and hex numbers beginning with a letter (like FFF9H) must be prefixed by a zero digit (0FFF9H). I will use both these conventions.

Assembly Language Statements

Assembly language requires line numbers, and each line must contain statements of specific types, subject to exact rules of syntax. Assembler uses short (two, three or four letter) mnemonics, so called because they are supposed to suggest their own meaning. ADD succeeds admirably in this, and LDIR does not.

Take a look at the center section of the EDTASM manual. You'll see a frightening array of these mnemonics. You will have to learn only a few of them for the project in hand. You'll also see references to registers, like A, C and HL. These are temporary storage places. We can put a value in a register, add to it, subtract from it, and retrieve it for future use. In BASIC, for example, we might write: X=10. We could then use the value of X later in such ways as: Y=X+1, or X=X+5.

In BASIC, a symbol table is set up by the computer, containing the names of all the variables and the locations in memory where their values are stored. When we reference a variable named X (as in Y=X+1), for example, the computer goes to the table, finds that the value of X is stored in location 18234, and uses that value to compute the value of Y. It then sets up an entry in the table to show where Y is located, and stores the value of Y in that place.

In assembly language, we don't have the luxury of naming variables (nor do we have to pay for that luxury in slow execution). Instead, we have to tell the computer where to store each value. If we want to use it again, we will have to tell the computer where to find it.

Values can be stored in two places: in a register, such as the A register; or in a specific place in memory, such as location 18234, or 473AH. A register can also point to a place in memory at which the value is stored. 18234, for example, could be stored in a register, and the computer then told to use the value whose location is contained in that register.

To store the value of 10 (decimal) in register A, we might use the following text: "LD A,0AH", which means "Load the value AH into register A". Note the 0 prefix, (necessitated because AH starts with a letter), and the tab between LD and A,0AH. Finally, note that the name of the register comes from first and the value follows, separated by commas.

Alternatively, we could write: "LD A,(473AH)", which means "Load the value to be found in location 473AH into register A". The location is in parentheses, which tell the computer that this is the location at which the value will be found, rather than that this is the value itself.

```
I110
00110   LD   A,58H          ;"X" = 88 = 58H
00120   LD   (3CF0H),A      ;SCREEN LOCA
TION 15600 = 3CF0H
00130
*— (after pressing break)
```

Fig. 5. Screen After Program Entry

```
*A
0000 3EF8      00110   LD   A,58H       ;"X" = 88 = 58H
0002 32F03C    00120   LD   (3CF0H),A ;SCREEN LOCA
TION 15600 = 3CF0H
NO END STATEMENT
00001 TOTAL ERRORS

READY CASSETTE
*— (after pressing break)
```

Fig. 6. Screen After Assembly

Can we write: "LD (473AH),0AH", meaning "Load AH into location 473AH"?

We cannot—don't ask me why. We have to do it in two steps:

```
LD   A,0AH
LD   (473AH),A
```

It is characteristic of assembly language programming to make things harder for the programmer, and easier for the computer.

One final complexity. Could we write: "LD A,473AH", meaning "Load 473AH into register A"?

Again, we cannot. Single registers, such as A and C, can handle only one byte at a time—integers between 0 and 255. 473AH requires two bytes. Assembler gives us a break here, however. We need two registers to handle two bytes, and we can accomplish this automatically by referring to a pair of registers.

Thus, in order to put AH into location 18234, we could write:

```
LD   HL,473AH
LD   (HL),0AH
```

which means "Load 18234 into register pair HL, then load AH into the location found in register pair HL."

These simple statements are all we need to draw that football field.

PRINT X

In BASIC or assembler, to print the letter X on the screen, the character value 88 (ASC("X") = 88) is loaded into one of the TRS-80's screen memory locations (15360 through 16383). Where it prints depends upon which of these locations we select. (Let's use location 15600, which is toward the top right of the screen. That's 3CFOH, and 88 is 58H.)

Your screen has been idle for some time, and it still reads NO TEXT IN BUFFER, *_. Respond to that prompt with I110, and you see 00110 appear. EDTASM is ready for line 110 of our text. Type in the code shown in Fig. 5, using the tab arrow to line up the columns. Note the comments in the last column preceded by semicolons, which are the equivalent of REM or apostrophe in BASIC.

Check your entries against Fig. 5. Edit if necessary. Now press A and, when you see READY CASSETTE, press Break. The screen should look like Fig. 6.

What is 00001 TOTAL ERRORS? You'll be glad to know that there are 00003 total errors, and not just 00001, but EDTASM has identified only 00001. (I like the way the designer provided for 99999; it gives us a feeling for his confidence in our skill). Here are the three:

1. The left-hand column lists the locations in which the machine code is to be stored in memory, beginning with 0000. Unfortunately, that can't be done, it's the very beginning of BASIC ROM, and we can't write into that area. Pick a place in RAM which is not otherwise inhabited; it's best to pick a place which will not interfere with any BASIC program we may want to use in conjunction with the machine language program.

   MEMORY SIZE? 32610, for example, sets aside memory locations 32611 through the end of RAM (32767 in a 16K machine) for machine language programs stored through the SYSTEM command.

   To specify where the program is to reside, assembler provides the statement ORG. Type I100 to insert an additional line 100 at the beginning of the program, and then enter the following:

```
00100   ORG   7F63H   ;STORE AT 32611 = 7F63H
```

2. Level II BASIC does not require an END statement, but assembler does. Insert a new line 140:

```
00140        END
```

3. The program as it is now written will work correctly, but we'll never know it. The X will print at location 15600, but the program will be completed and the TRS-80 returned to its power up condition so fast that we'll never see the X. An endless loop will stop the program before the END statement, like "130 GOTO 130" in BASIC. Insert a new line 130:

```
00130 LOOP   JP   LOOP   ;ENDLESS LOOP
```

That's the first time we've used the tab field immediately following the line number. It's used for labels—named places in memory whose actual location EDTASM will compute and remember. JP means Jump, and the place to which the program jumps is found in the fourth column: in this case, the place in memory named LOOP.

During execution, the computer finds the label called LOOP, and makes note of its location. When it encounters JP, it looks to see where it should JP to, finds that it should JP to a location named LOOP, does so, and repeats the process endlessly.

Press A again. Your screen should look like Fig. 7. Instead of pressing Break, this time load a fresh cassette into your recorder, set it to record, and hit any key to save the machine language version of the program on cassette. Advance the tape a few clicks and remember the place. Press record again, and enter W on the keyboard. The text version of the program will be saved on the cassette beginning at the second location. You don't have to save the text, but you'll have to reenter it by hand if you don't. There's no way to reload the machine language version into EDTASM.

Exit EDTASM (using B). Rewind the cassette to the beginning. Press play. Respond to MEMORY SIZE? with 32610, and then enter SYSTEM. You'll see the prompt ?—, which means that the computer wants the name of the machine language program it is to load. Since we gave it no name, enter the default name NONAME.

After it loads, you'll see another ?— prompt. Respond /32611, which means "Ex-

```
A*
7F63                00100         ORG   7F63H       ;STORE AT 32
611 = 7F63H
7F63 3E58           00110         LD    A,58H       ;"X" = 88 = 58H
7F65 32F03C         00120         LD    (3CF0H),A   ;SCREEN LOCA
TION 15600 = 3CF0H
7F68 C3687F         00130 LOOP    JP    LOOP        ;ENDLESS LOO
P
0000                00140         END
00000TOTAL ERRORS

LOOP 7F68
READY CASSETTE
```

Fig. 7. Screen After Revised Assembly

ecute the program beginning at memory location 32611", which is the 7F63H we specified as the ORG in line 100, and, by golly, we have an X on the screen. It appears in addition to the other contents of the screen because our program did not contain a screen clearing instruction.

After you've admired that X for a sufficient length of time, you'll find something else our program did not contain: a way to stop. It continues to churn back and forth from the beginning of line 130 to the end of line 130, and back again. To exit, you'll have to press the reset button.

What does the machine code look like? One way to find out is to PEEK into the memory locations in which it is stored. Press that reset button, if you have not already done so. Enter: FOR I=32611 TO 32622 : PRINT PEEK (I); : NEXT. You get: 62 88 50 240 60 195 104 127 255 255 255 255. Translate those decimal numbers into hex, omitting the 255's (they are simply nulls remaining after the end of the machine code). You get: 3E 58 32 F0 3C C3 68 7F.

Look back at the text of the assembly language program in Fig. 7. These hex numbers are identical with those appearing in the second column. This is the machine code representation of the mnemonics, register names, values and locations we specified in the text. When you loaded NONAME into the machine, it placed these codes in locations beginning at 32611. PEEK pulled them out and translated them into decimal.

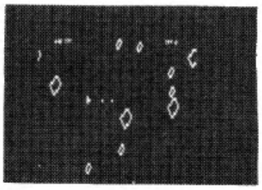
That's all there is to the machine code—no line numbers, no comments, no mnemonics, just these codes. There are only eight of them, and that means that the memory required to store this exciting program is only eight bytes.

Of course, the program didn't do much, either. But our illustrative BASIC program took 46 bytes to do the same inconsequential job, and would take 27 bytes even if optimized. We can't see much difference in speed yet—how long could it take to print an X? But we will, in Part II of this article.■

Games from BIG FIVE will turn your computer into a

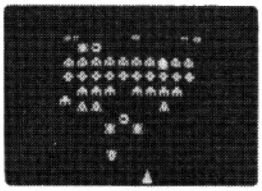
TRS-80 HOME ARCADE

SUPER NOVA©



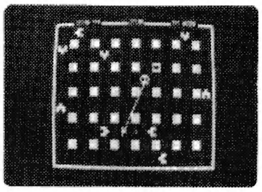
If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate!

GALAXY INVASION©



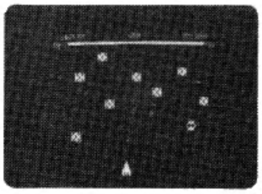
The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects!

ATTACK FORCE©



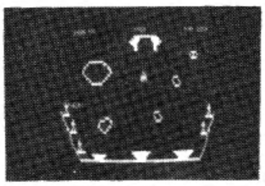
Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects!

COSMIC FIGHTER©

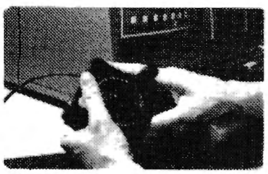


With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound!

METEOR MISSION II©



As you look down on your space viewer you can see the stranded astronauts that are crying out for you to rescue them. But first you must maneuver your shuttle down through the asteroids & meteors before you can reach them. Great! You've got one! But now can you get back to the space station to save your fellow shipmate or will you crash and kill both of you? You can fire your lasers to destroy the asteroids, but watch out, because there could be an alien FLAGSHIP lurking behind! Includes sound effects!



NEW JOYSTICKS!

For $39.95 it's now possible to have the famous ATARI joystick interfaced with your Model 1. All of our tapes are now completely compatible with the joystick. Packaged with complete instructions, you can even use it with your own programs! If your old tapes do not say "Joystick Version" on them and you wish to exchange them for new Joystick versions, enclose $2 and your old tape. (Call or write for info on Mod 3 joysticks.)

BIG FIVE SOFTWARE

~357

P.O. Box 9078-185, Van Nuys, California 91409

For MC/Visa or COD orders, call (213) 782-6861 from 9am-5pm Pacific time.

Prices per game: Level 2, 16K Mod 1/Mod 3—$15.95
Level 2, 16K Mod 1/Disk—$17.95
10% discount for 2 items, 15% for 3 or more. (Including Joystick)
Please add $1.50 postage/handling, Calif. residents add 6% tax.
All games are written in machine language and supplied on cassette.
Disk versions save high scores to your TRSDOS or NEWDOS diskette.

Cassette versions require 16K memory, disk versions require 32K.
We accept checks, money orders, and MC/Visa orders.
All games ©1980 by Bill Hogue & Jeff Konyu.
TRS-80 & TRSDOS are trademarks of Tandy Corp.
NEWDOS is a trademark of Apparat, Inc.
Dealer inquiries invited.

When white is black on the Model I.

Inverse Video

Brian D. Smith
6770 Halifax St.
Burnaby, B.C., Canada V5B 2R4

Have you ever drooled over some of the other personal computers with their inverse video cursors and their ability to display some sections of the screen in black on white and others in white on black? Although Tandy decided that we did not need or want this feature, it is possible to add it to a TRS-80.

With three ICs, one switch, and a lowercase modification, you can reverse any location or locations on the screen with a simple POKE. There *is* a "gotcha," however: You cannot have inverse characters and graphics on the screen simultaneously.

I recently found an inverse character modification in a users' group newsletter. It used the eighth bit of each character (normally used to indicate a graphics character) to identify an inverted character. I installed the mod and was quickly disappointed. The inverse field always appeared 1½ characters to the left of the location inverted. The author had a clever program that got around this problem, but I thought, there must be a better way.

A Modification Is Born

I began by studying the circuit diagrams and descriptions in the *TRS-80 Microcomputer Technical Reference Handbook.* Anyone planning to modify an 80 should have one; my explanation of the operation of the modification will assume you have a copy of the schematic from the handbook.

I decided that the best method of decoding inverse characters is with bit 7, the graphics bit. This means graphics and inverse characters will not be available at the same time. However, all alphanumeric characters can be reversed. Since the primary use of inverse characters will be with text, this seems to be the best compromise.

Bit 7 must be synchronized with the other data bits. Looking at the 80's schematic, you will see on the second sheet that bits 0 through 5 go from the video memory chips to Z28, and bits 6 and 7, along with some other signals, go to Z27. These two ICs are simply latches used to hold the data read from the video memory.

Bits 0 through 6 are then sent to both Z29 (the character generator) and Z8 (the graphics generator). Up to this point, bit 7 is still in step with the other data bits. From Z29 and Z8, the data bits go to Z10 and Z11, which are shift registers used to output the characters from the generators one dot at a time.

Bit 7 is used to select which shift register accepts data, determining whether a normal or graphics character is displayed. The problem is that bit 7 is only around long enough to load the shift register. By the time the dots are being output from the register, the values at the latches of all the data bits, including bit 7, are those for the next character to be displayed. The result is that the inverse video always appears one character-width to the left of where you intended.

Parts List

S1	DPDT switch
Z10A	74LS166
Z26A	74LS20
742A	74LS86
Misc. thin-guage wire (wire wrap is ideal)	

```
10 ' INVERSE AND LOWERCASE VIDEO DRIVER SETUP PROGRAM
20 CLEAR 100
30 CLS
40 ' GET THE CURRENT MEMORY SIZE
50 MS = PEEK(16561) + PEEK(16562) * 256
60 MS = MS - 49
70 ' SET THE BASIC MEMORY SIZE
80 POKE 16561, MS - (256 * INT(MS/256))
90 POKE 16562, INT(MS/256)
100 M = MS
110 IF M > 32767 THEN M = M - 65536
120 ' LOAD THE DRIVER INTO HIGH MEMORY
130 FOR I = M+1 TO M+49
140   READ N
150   POKE I,N
160   NEXT I
170 ' SET UP THE ADDRESS OF THE INVERSION MASK
180 MH = INT((MS+49)/256)
190 ML = (MS+49) - (256*MH)
200 POKE M+16,ML : POKE M+17,MH
210 POKE M+21,ML : POKE M+22,MH
220 POKE M+43,ML : POKE M+44,MH
230 ' LOAD THE ADDRESS OF THE DRIVER INTO THE VIDEO DEV
'ICE
240 ' CONTROL BLOCK
250 POKE 16414, (MS+1) - (256 * INT ((MS+1)/256))
260 POKE 16415, INT((MS+1)/256)
270 CLEAR
280 END
290 DATA 221, 110, 3, 221, 102, 4, 218, 154, 4, 121
300 DATA 254, 127, 32, 9, 58, 0, 0, 238, 128, 50
310 DATA 0, 0, 201, 221, 126, 5, 183, 40, 1, 119
320 DATA 121, 254, 32, 218, 6, 5, 254, 128, 210, 166
330 DATA 4, 58, 0, 0, 177, 195, 125, 4, 0
```

Program Listing 2. Cassette based inverse video driver.

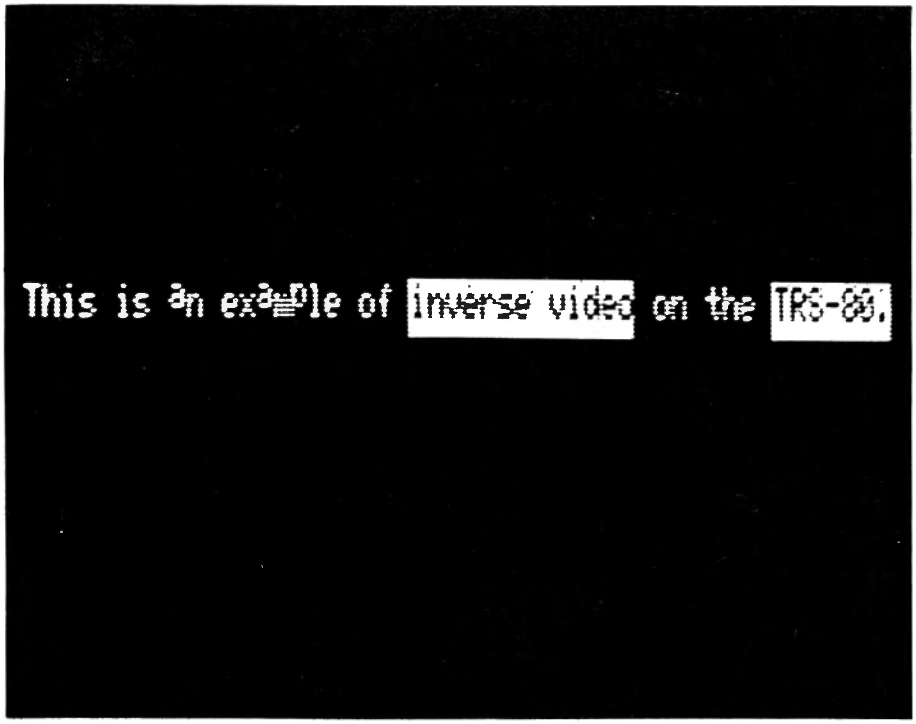


Photo 1

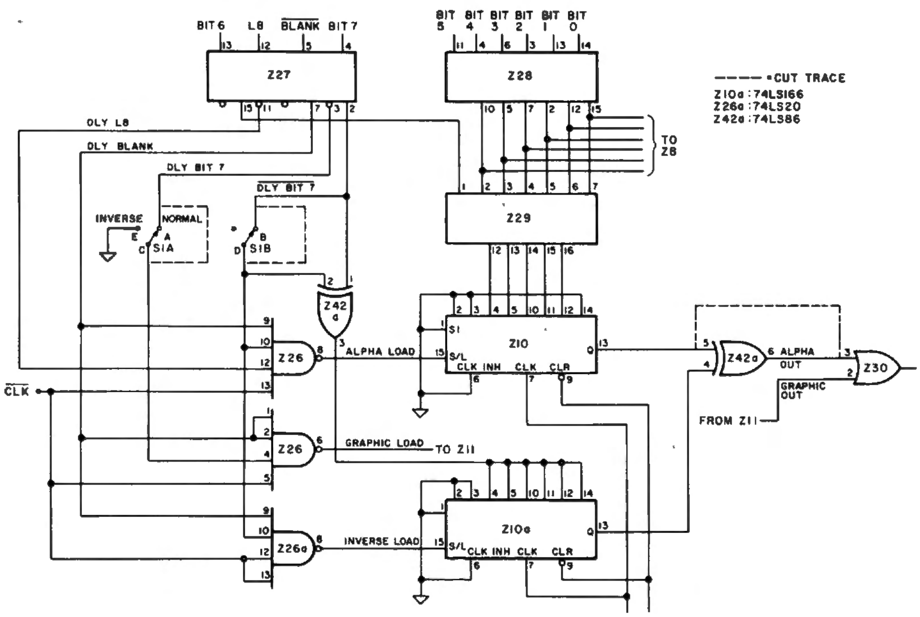


Figure 1. Inverse Video Schematic.

The solution is to install a new shift register for bit 7, the inverse character signal. This new shift register, which I call Z10A, is run in parallel with Z10, so the inverse signal keeps pace with the character output.

A complete description of how this modification works follows (see Fig. 1).

The DPDT switch is used to choose between the normal graphics mode and the inverse character mode. In the normal position, the bit 7 signals are passed through to Z26 as usual. In the inverse mode, DLY BIT 7 is forced low and $\overline{\text{DLY BIT 7}}$ is forced high, as when displaying alphanumeric characters. When bit 7 is high, in inverse mode, pin 2 of Z27 goes low. The difference between this signal and DLY BIT 7 (after the switch) is detected by Z42A, and pin 3 goes high. This signal is fed into Z10A. The high bits are then output in step with the bits from Z10. The two signals are then exclusive-ORed by Z42A. Since the output from Z10A is high, the character bits are inverted.

When the switch is in normal position (bit 7 is low), pin 3 of Z42A will be low, since both inputs are the same. This signal will be input to Z10A and stepped out as before. Since the output from Z10A is now low, when the exclusive-OR is done the character bits will be unchanged.

Z26 causes Z10 to load alphanumeric data when the CLK, DLY BIT 7, DLY BLANK, and DLY L8 signals are all low. Z26A does the same for Z10A, but does not use the DLY L8 signal, because DLY L8 is used to generate the four blank rows beneath each character. It is ignored by Z26A so that those four rows will also be inverted.

Lowercase is required to get inverse characters because when bit 7 is high in an unmodified 80, bit 6 is forced low. Therefore, all alphabetic inverse characters display as inverse control characters.

Installation

The first step when installing the inverse character modification is to find a handy place to mount the normal/inverse switch. I found the most convenient location to be on the upper half of the case, on the downward-sloping section just back of the keyboard. There is more than enough space between the case and the circuit board on which the keys are mounted for a miniature toggle switch.

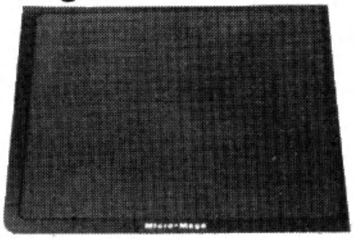
The second step is to open up the S-80.

Once the case is open, place the circuit board on a flat surface, component side up. Locate ICs Z10, Z26, Z27, Z30, and Z42. Make the modification following these steps:

1. Bend all pins of Z10A, except 1, 2, 3, 6, 7, 8, 9, and 16 into a horizontal plane.
2. Bend all pins of Z26A, except 7, 9, 10, 13, and 14 into a horizontal plane.
3. Bend all pins of Z42A, except 7 and 14, into a horizontal plane.
4. Solder Z10A into position on top of Z10, soldering the unbent pins of Z10A to the corresponding pins of Z10.
5. Similarly solder Z26A on top of Z26.
6. Similarly solder Z42A on top of Z42.
7. Cut the trace between Z27 pin 3 and Z26 pin 4.
8. Cut the trace between Z27 pin 2 and Z26 pin 10.
9. Cut the trace between Z10 pin 13 and Z30 pin 3.
10. Solder a wire from Z10A pin 4 to Z10A pin 5.
11. Solder a wire from Z10A pin 5 to Z10A pin 10.
12. Solder a wire from Z10A pin 10 to Z10A pin 11.
13. Solder a wire from Z10A pin 11 to Z10A pin 12.
14. Solder a wire from Z10A pin 12 to Z10A pin 14.
15. Solder a wire from Z10A pin 5 to Z42A pin 3.
16. Solder a wire from Z10A pin 13 to Z42A pin 4.
17. Solder a wire from Z10A pin 15 to Z26A pin 8.
18. Solder a wire from Z42A pin 1 to Z27 pin 2.
19. Solder a wire from Z42A pin 2 to Z26 pin 10.
20. Solder a wire from Z42A pin 5 to Z10 pin 23. (You may find it easier to attach

for the TRS-80 from Micro-Mega

The Original GREEN-SCREEN ✓29



The eye-pleasing Green-Screen fits over the front of your TRS-80 Video Display and gives you improved contrast with reduced glare. You get bright luminous green characters and graphics like those featured by more expensive CRT units.

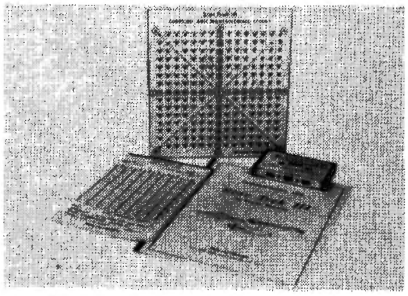
Don't confuse the Original Green-Screen with a piece of thin film stuck to the face of your video tube, such as that advertised by others. The Original Green-Screen is mounted in a full frame perfectly matched to the color and texture of the TRS-80 Video Display. It is attached with adhesive strips which do not mar your unit in any way.

The full frame design of the Original Green-Screen "squares off" the face of your video display and greatly improves the overall appearance of your system.

(Specify whether for Model I or Model III)

THE GREEN-SCREEN..*$13.95*
Add $1.00 for postage and handling.

THE ULTIMATE STAR TREK PACKAGE



Tired of trivial computer games? This complete Star Trek package will provide you with endless fascination and challenge. In addition to the program cassette, it includes comprehensive instructions, a pad of "Voyage Log" record sheets, and a free-standing "Torpedo and Maneuvering Chart."

The package is built around the latest version of Lance Micklus' incomparable Star Trek III, a 13,000 byte program with a host of subtle and imaginative features, which include numerous dynamic and spectacular graphic displays. Star Trek III puts you in command of the Enterprise cruising in a galaxy of 192 quadrants filled with uncharted hazards, including hostile Klingons, pulsars, and black holes. You have at your disposal scanners, various weapons and defense systems, on-board computers, and a loyal crew. (You will need them all to survive the Klingons.)

Your mission is to rid the region of Klingons and to locate five inhabitable planets, all within 300 stardays, before returning to Star Fleet Headquarters where your overall effectiveness as a starship commander will be scored. High scores are possible only with careful planning and effective battle tactics. The "Voyage Log" sheets will guide your strategy, and the "Torpedo and Maneuvering Chart" will give you a vital edge in combat. (When you engage three Klingon ships you can't afford to miss.)

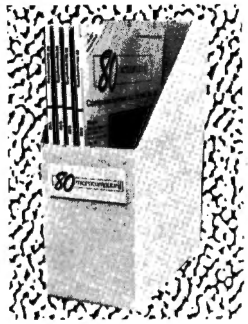
STAR TREK PACKAGE (for Level II, 16K only)..$22.95
Add $1.00 for postage and handling

Terms: Check or money order, no CODs or credit cards, please. Add amount shown for postage and handling to price of the item. All items shipped within 48 hours by first class or priority mail. Virginia residents, add 4% sales tax.

Micro-Mega · P.O. Box 6265 · Arlington, Va 22206

---

is HARD COPY STORAGE a problem?



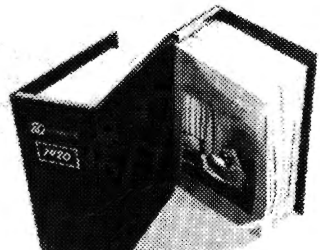
Here's the ideal way to keep your growing collection of *80 Microcomputing* in order! The *80 Microcomputing* Library Shelf Boxes. . . . sturdy, corrugated, white, dirt-resistant boxes will keep your issues of *80 Microcomputing* orderly and available for constant reference. Self-sticking labels are available for the boxes, too, not only for *80* but also for *Kilobaud/Microcomputing, 73 Magazine, CQ, QST, Ham Radio, Personal Computing, Interface Age, Byte* and *Radio Electronics*. Ask for whichever labels you want with your box order. Each box holds a full year of the above magazines. Your magazine library is your prime reference: Keep it handy and keep it neat with these strong library shelf boxes. One box (BX1000) is $2.00, 2–7 boxes (BX1001) are $1.50 each, and 8 or more boxes (BX1002) are $1.25 each. Be sure to specify which labels we should send. Shipping and handling charges are $1.00 per order. Call in your credit card orders on our toll free line 800-258-5473, or use the order card in the back of the magazine and mail to:

80 microcomputing™ Peterborough, NH 03458

Please allow 4-6 weeks for delivery. No COD orders accepted.

---

80 microcomputing™ BINDERS



order yours today

Keep your library of 80 Microcomputing safe from loss or damage in these handsomely appointed binders with rich dark green covers and gold lettering. Each binder holds 12 issues making an EXCELLENT REFERENCE HANDBOOK. Several binders form a quality library you can be proud of.

$7.50 each. . . 3 for $21.75. . . 6 for $42.00. . .

Postage paid in USA. Foreign orders please include $2.50 for postage.

Specify 1980 or 1981

Send check or money order only to:
80 MICROCOMPUTING BINDERS
P.O. Box 5120, Phila., PA 19141

Allow 6-8 weeks for delivery. Please no C.O.D. orders, no phone orders.

```
                 ;***********************************************************
                 ;* LOWER CASE AND INVERSE CHARACTER DRIVER PROGRAM        *
                 ;* (DISK VERSION)                                         *
                 ;***********************************************************
                 ;
007F             RVCODE  EQU     7FH             ; REVERSE CONTROL CODE
401E             VIDEO   EQU     401EH           ; VIDEO DCB ADDRESS
4049             HIMEM   EQU     4049H           ; DOS HIGH MEMORY POINTER
                 ;
FFB0                     ORG     0FFB0H
                 ;
FFB0 21BFFF      LOADER  LD      HL,START        ; GET DRIVER ADDRESS
FFB3 221E40              LD      (VIDEO),HL      ; AND INSERT IT INTO THE
                                                 ;   VIDEO DEVICE CONTROL
                                                 ;   BLOCK
                 ;
                 ; THE FOLLOWING TWO INSTRUCTIONS AUTOMATICALLY SET THE
                 ; DOS HIGH MEMORY POINTER.  THIS ONLY APPLIES TO NEWDOS
                 ; (2.1 OR 80) AND TRSDOS 2.2 OR LATER.
                 ; FOR OTHERS, THEY SHOULD BE DELETED.
FFB6 21BEFF              LD      HL,START-1      ; GET MEMORY SIZE
FFB9 224940              LD      (HIMEM),HL      ; AND SAVE IT
                 ;
FFBC C32D40              JP      402DH           ; RETURN TO DOS
                 ;
FFBF DD6E03      START   LD      L,(IX+3)        ; GET THE ADDRESS OF
FFC2 DD6604              LD      H,(IX+4)        ; THE CURSOR
FFC5 DA9A04              JP      C,049AH         ; IF READING FROM SCREEN,
                                                 ;   THEN RETURN TO ROM
                 ;
FFC8 79                  LD      A,C             ; GET THE CHARACTER
FFC9 FE7F                CP      RVCODE          ; IF IT'S NOT REVERSE CODE
FFCB 2009                JR      NZ,DISPLY       ;   THEN GO TO "DISPLY"
FFCD 3AEFFF              LD      A,(MASK)        ; GET THE REVERSE MASK
FFD0 EE80                XOR     80H             ; AND TOGGLE IT
FFD2 32EFFF              LD      (MASK),A        ; SAVE THE NEW MASK
FFD5 C9                  RET                     ; RETURN WITHOUT DISPLAY-
                                                 ;   ING THE CHARACTER
                 ;
FFD6 DD7E05      DISPLY  LD      A,(IX+5)        ; IF THE CURSOR
FFD9 B7                  OR      A               ; IS ON
FFDA 2801                JR      Z,GETCHR
FFDC 77                  LD      (HL),A          ;   THEN BLANK IT
                 ;
FFDD 79          GETCHR  LD      A,C             ; GET THE CHARACTER
FFDE FE20                CP      20H             ; IF IT'S A CONTROL CHAR.
FFE0 DA0605              JP      C,0506H         ;   THEN JUMP BACK TO ROM
FFE3 FE80                CP      80H             ; IF IT'S A GRAPHIC CHAR.
FFE5 D2A604              JP      NC,04A6H        ;   THEN JUMP BACK TO ROM
FFE8 3AEFFF              LD      A,(MASK)        ; GET THE INVERSION MASK
FFEB B1                  OR      C               ; AND 'OR' IT WITH CHAR.
FFEC C37D04              JP      047DH           ; JUMP BACK TO ROM
FFEF 00          MASK    DEFB    00              ; REVERSE MASK (INITIALLY
                                                 ;  SET FOR NORMAL DISPLAY)
                 ;
FFB0                     END     LOADER          ; AUTO-EXECUTE DRIVER
```

Program Listing 1. Disk based inverse video driver.

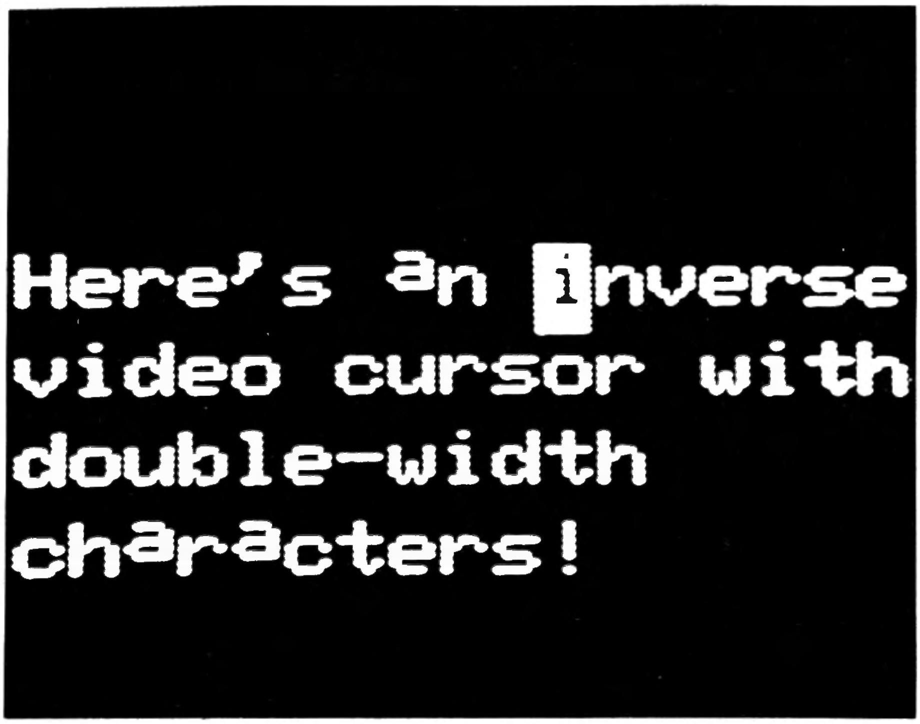


Photo 2

the wire to Z10 on the trace side).

21. Solder a wire from Z42A pin 6 to Z30 pin 3.

22. Solder a wire from Z26A pin 12 to Z26A pin 13.

23. With reference to the schematic, solder a wire from point A on the switch to Z27 pin 3.

24. Solder a wire from point B to Z27 pin 2.

25. Solder a wire from point C to Z26 pin 4.

26. Solder a wire from point D to Z42A pin 2.

27. Solder a wire from point E to Z10A pin 8.

28. Install the switch in its mounting position.

Check your wiring for incorrect connections and solder bridges and then reassemble the case.

Testing

On power up, you should see a normal display. If everything is correct so far, enable your lowercase and run the following BASIC program:

```
10 CLS
20 For I=0 TO 255
30 POKE I+15360,I
40 NEXT I
50 GOTO 50
```

You should see the normal alphanumeric character set (with control characters) displayed on the first two lines of the screen. With the switch in the normal position, you will see graphic characters on the second two lines; with the switch in the inverse position, you will see the alphanumeric characters in inverse on the second two lines. If you do not, go back and check your wiring: Something is wrong.

Program Listings 1 and 2 are versions of a machine language inverse video driver, providing lowercase and inverse display. The inverse display is turned on and off with the character code 127, the ASCII DEL character. For example, assuming the display is in normal mode to start with, the following BASIC statement will display "Hello there" in inverse video:

```
PRINT CHR$(127);"Hello there";CHR$(127)
```

The first version is designed for disk users with a disk based assembler. When the program is executed from DOS it automatically links the new video driver into the device control block and sets the DOS memory size (in NEWDOS and TRSDOS 2.2 or later). Therefore, you need not specify the memory size if you enter BASIC with the driver installed, since the DOS memory size is used as the default.

The second version is for Level II 80s and disk users without an assembler. The BASIC program POKEs the driver into high memory and links it into the control block. It also sets the BASIC memory size.

Inverse characters can be used for many purposes. One is to provide a non-destructive cursor for text editors; you can also highlight text, showing important fields in inverse video while the others are in normal video. For other applications, just use your imagination.... ■


$300
MICROTEK
BYTEWRITER-1

The World's First Under $300 80 Column Dot Matrix Printer

WHY DID IT HAPPEN?

We were approached recently by a man with an idea. He is one of the leading American manufacturers of dot matrix print mechanisms. He had observed, as we had, that in recent months certain foreign printer manufacturers were increasing their share of the low-cost printer market at an alarming rate.

He thought there MUST be a way to fight back. And he was right. We've since formed a working alliance with this manufacturer, and have brought our first joint offering to the market.

HOW DID WE MAKE IT POSSIBLE?

We set out to combine his proven low cost print mechanism with the simplest possible control electronics. Advances in single-chip microprocessor technology and price erosion of components during the last year helped to make this long awaited dream come true - a printer that can be sold for less than half the cost of the computer that drives it. A $299 printer.

But cost-effective designs and efficient manufacturing operations weren't enough. Computer retailers can make up to a $250 markup on the foreign models. Could we hold to a $299 list price and give the dealer enough incentive to sell the Bytewriter-1? No way. We had to try a more direct approach.

YOUR BUY DECISION - DEALER OR MAIL ORDER

There are some very good reasons to buy your first computer through a dealer. There is a certain amount of hand-holding required when you decide to buy a personal computer. This is one of the main functions of the retail computer store. And most of them perform this function very well.

But why would anyone want to buy add-on equipment through a dealer? If you find a product that has been designed for and tested with your particular computer, you can safely bypass the computer dealer. You can have the best of both worlds. You can save money by buying direct from the manufacturer, and you can be certain that your new device will work when you get it.

We've done extensive testing with the most popular computers - the TRS-80, the Apple II, and the Atari 400 and 800. If you own one of these computers, we guarantee you won't have any interface problems with the Bytewriter-1.

TRS-80 is a trademark of Radio Shack, Div. of Tandy Corp.
Apple II is a trademark of Apple Computer, Inc.
Atari 400 & 800 are trademarks of Atari, Inc.
Bytewriter-1 is a trademark of Microtek, Inc.

FOUR THINGS YOU SHOULD KNOW BEFORE YOU BUY THIS PRINTER

We don't want any unhappy customers. We'd like you to know the limitations of our printer, as well as its advantages. There are some differences between the Bytewriter-1 and the higher priced printers you may be looking at:

1) The Bytewriter-1 takes single sheet and roll paper only. No pin feed paper.
2) We've used a 7-wire print head. No fancy lower case descenders.
3) There aren't any software frills in the Bytewriter-1, like VFU controls. However, if your main interest is getting software listings or printing letters, you won't care. And, with a bit of ingenuity, you can provide VFU functions in your own programs.
4) You can't go into a computer store and pick up a Bytewriter-1. They're sold direct only by MICROTEK.

We realize it's unusual to point out the limitations of a product in an ad that promotes it, but we think it's important for mail order buyers to fully understand what they're buying.

The Bytewriter-1 will fill the needs of most people. People who don't see the sense in spending extra money for features they'll never use.

ORDER THE CONFIGURATION THAT'S JUST RIGHT FOR YOU

The Bytewriter-1 is available with an interface cable and complete instructions for use with three of the most popular small computers on the market today, the Apple II, the Atari 400/800, and TRS-80 Models I, II, and III. One of our divisions, MICROTEK PERIPHERALS CORP., can even provide you with the expansion card or module that your computer may require to drive a printer.

CALL OR WRITE TODAY FOR MORE INFORMATION.

We have people standing by to answer your questions.

MICROTEK inc.

Microtek, Inc. ✓362
9514 Chesapeake Drive
San Diego, CA 92123
Tel. 714-278-0633
Outside Calif. call toll free: 800-854-1081
TWX. 910-335-1269

An analysis program for geodesic dome design.

Dome Time

Daniel B. Nickell
206 Jill Lane #2
Laurel, MD 20810

In the early 1960's there was a push forward in experimentation with geodesic domes as a form of housing and shelter.

First patented in 1951 by Buckminster Fuller, the geodesic dome offers an attractive combination of low cost, high structural strength and energy efficiency. The first domes, however, suffered from some nagging practical problems which kept the concept from wide acceptance.

Today, most of the problems—mainly sealing the dome against leaks—have been solved and a number of companies are offering a full range of dome designs. For those that are interested in working up their own designs the following program may be of some use.

Dome Theory

The basic theory of geodesic domes is that a sphere, or a portion of one, is the most efficient container of a given volume and is also the strongest; that is, the shape tends to distribute forces applied to it evenly over the entire surface.

The geodesic is an attempt to provide a practical solution to the construction problem of building a sphere in an industry using straight-edge technology. By fastening together a number of triangles, Fuller illustrated a structure that resembles a sphere and can be built with the existing technology. Fuller also discovered that the greater the number of triangles involved, the closer the structure approaches spherical proportions and qualities.

This program provides the basic data required for constructing a dome called a "Three-Frequency Icosahedron." It is a very simple, yet functional geodesic.

All you need supply is the diameter of the sphere upon which the dome will be based and the scale of the model to be built. You should also know whether the dome is to be a 3/8, 1/2, or 5/8-dome. This means, what portion of the sphere is to be used as the structure: 3/8, 1/2 or 5/8. A 3/8-dome is the cheapest to build as it requires the least amount of materials. The 5/8 offers the greatest interior space and the most efficient use of that space. The 1/2-dome is a good compromise between two.

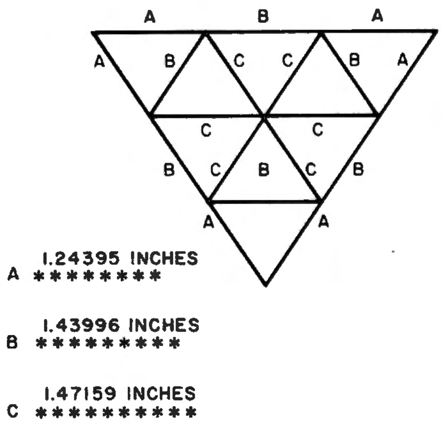


Fig. 1. Construction Triangle.

The program first asks whether you want to use a line printer for part of the output. This listing was written with the Heath H-14 line printer in mind. If you are using another, you'll have to make some adjustments, mainly in lines 1000-1060, to print the scale correctly. If you don't have a printer, delete these lines entirely.

Next, the computer will ask for data. First, enter the diameter or the square footage of the floor. If you enter the square footage first, the computer defaults to the 1/2-dome variety. If you specify a diameter, you will then be asked to select the variety of dome (3/8, 1/2 or 5/8) before the area and volume computations are made. By playing with these input variables, you can decide on the proportions best suited to your purpose.

The computer takes care of the rest. The first output displayed is the full scale dome data. Review it. If it fits your needs, then type C and INKEY$. Otherwise, type R INKEY$, and reset the input data until you are satisfied. Table 1 shows the sample output for a 4/8 (1/2) dome with a desired floor space of 640 square feet.

Continuing, the computer then asks for the type of scale you want to use. The first, A, is generally used by architects and home designers. Some minor program modifications are necessary if you want to work in metric dimensions. The second type of scale, B, is used frequently by model builders. One-half-scale is just that—the model is one-half of the acutal size. You can use metric dimensions in this scale with only two changes to the program: Adjust the labels to match, and change line 460 so SF=(100*SN)/SD instead of SF=(12*SN)

These Figures are For a 4/8ths Geodesic Dome, 28.546 Feet In Diameter.

Height Of Dome	=14.273 Feet
Surface Area Of Dome	=1280 Square Feet
Volume Of Dome	=6089.82 Cubic Feet
Area Of Floor Within	=640 Square Feet
Radius Of Floor	=14.273 Feet

C To Continue, R to Readjust Input Data

Table 1. Summary of Dome Statistics.

These Figures are For a Model Of a 4/8ths Geodesic Dome.
The Diameter Of The Model Dome is 7.1365 Inches.
Scale Used To Obtain These Data: 1"=4'

Height of the Dome Model	=3.56825 Inches
Surface Area of the Model	=80 Square Inches
Volume of Model	=95.1534 Cubic Inches
Area of Model Floor	=40 Square Inches
Radius of Model Floor	=3.56825 Inches

Hit ENTER When Ready to Proceed.?

Table 2. Model Statistics

/SN. The conversion will then be from meters to centimeters, instead of from feet to inches.

I use a scale of one inch equals four feet. Graph paper with quarter-inch line spacing is readily available and each block therefore equals one square foot. The sample output in Table 2 is based on this scale. These data refer directly to the model of the dome. You can use these in the construction of the model.

Chord Length

The next display represents the lengths of the three different chords you will need to build the geodesic frame. Assuming your TRS-80 monitor is properly aligned, the length of the chords should be accurate to about 1/16th of an inch of the calculated length. If the chord is longer than about 7.2 inches, the program will reduce the display to show the *relative* sizes of the chords, not the actual. In this instance, the display will tell you the degree of reduction and you would have to use your own measuring device to lay out the lengths.

If you are using a line printer, all of the above information will have been printed out by this time. Using the asterisk on the H-14 printer results in the same level of accuracy—about 1/16 or 1/17th of an inch. If you use a different make of printer or use a different character you will have to experiment to find the correct scaling factors in line 760. See Table 3.

The next display (Fig. 1) repeats the chord lengths and draws a basic construction triangle. You can refer to this for the proper relationship of the various chords. Typing an @ at this point will enable you to access the drafting aid subroutine. This subroutine allows you to convert a measurement from either scale to real or from real to scale. From this subroutine you can either return to the previous graphic, recompute another conversion, or terminate the program. See Table 4.

If you are handy with assembly language, try adding a USR(0) subroutine to speed up some of the graphics displays. Yet, for the most part, the existing program is fast enough. The program requires about 9500 bytes, which should leave enough room for user-designed additions or modifications to the program.

With very few exceptions, the line numbering in the program is in increments of 10. I omitted the first 89 titling lines. Lines that end with a five—except 255 and 395, which were added while preparing the article to speed up its operation and tidy up program logic—can also be omitted. I used INKEY$ routines where possible and practical to ease data entry. ■

```
Are you converting from (S)cale to real
or from (R)eal to scale? R
Enter known value? 15
The scale value of 15 feet = 3.75 inches
(G)raphics, (E)nd, or (C)onversion help E
(screen clears)
(R)epeat to see these data again, (A)nother dome
data set, or (E)nd to end program. E.
(screen clears)
Good luck with your project.
Remember that the values shown for the struts
are total length struts. If you are using hubs, be
sure to subtract the proper amount from the strut
so the overall length is not greater than shown in
this program.
Thanks for using geodesics.
Ready
```

Table 4.

Data for the construciton of a model of a geodesic dome (3-frequency alternate). Refer to 'The Domebook 2', published in 1972, by Shelter Publications.

UNIT	REAL (IN FEET)	MODEL (IN INCHES)
Height of Dome	37.5000	9.3750
Surface Area of Dome	7068.5800	441.7870
Volume of The Dome	77312.7000	1208.0100
Area of Floor	2650.7200	165.6700
Radius of Floor	29.0474	7.2618

```
Scale of Model: 1 Inch Scale = 4 Feet Real.
Chord 'A' = 2.61462 Inches.
  A****************************************
Chord 'B' = 3.02661 Inches.
  B***********************************************
Chord 'C' = 3.09309 Inches.
  C********************************************************
```

Table 3. Conversion Data Showing Chords.

Program Listing

```
90 CLEAR 1000
100 PI=3.1415926535
110 CLS:PRINT"          ******  GEODESIC MODELS  ******"
115 'INPUT DATA
120 PRINT@192,"SO YOU ARE GOING TO BUILD A GEODESIC MODEL ARE YOU?
    "
130 PRINT:PRINT"(WILL YOU NEED THE LINE PRINTER?  (Y OR N))"
140 Q$=INKEY$:IFQ$=""THEN140ELSE150
150 IF Q$="Y"THENLP=1ELSELP=0:IF LP=1THEN160ELSE180
160 LPRINT" ":LPRINT"DATA FOR THE CONSTRUCTION OF A MODEL OF A GEO
     DESIC DOME (3-FREQUENCY ALTERNATE).  REFER TO 'THE DOMEBOOK 2
     ', PUBLISHED IN 1972, BY SHELTER PUBLICATIONS."
170 LPRINT" "
180 CLS:PRINT"WELL, THIS PROGRAM WILL HELP YOU A LITTLE BIT.  FIRS
     T,
WE WILL NEED TO THE HOW LARGE THE DOME WOULD BE IN 'REAL'
     LIFE.
THEN, WE WILL NEED TO KNOW HOW MUCH SMALLER THE MODEL W
     ILL BE."
190 PRINT:PRINT"IF YOU DON'T KNOW THESE VALUES, THE COMPUTER WILL
     WAIT
WHILE YOU GET THEM.  IF YOU HAVE THEM, HIT ENTER.":INPUT
     XX
195 'THE FOLLOWING PROCEDURES ASSUME ALL MEASUREMENTS ARE IN UNITS
      OF FEET AND INCHES (DECIMAL).  TO USE METRIC, SIMPLY CHANGE

     THE LABELS IN LINES 340-380, 450, 500, 520-570, AND 700.
200 CLS:PRINT@192,"NOW THAT YOU'RE READY, GIVE ME TWO VALUES:"
210 PRINT:INPUT"FIRST, THE DIAMETER OF THE DOME OR THE
SQUARE FOOT
     AGE OF THE DESIRED FLOOR";A
220 PRINT:PRINT"NOW TELL ME WHETHER";A;"IS THE
(D)IAMETER OR THE (
     S)QUARE FOOTAGE:"
230 A$=INKEY$:IF A$=""THEN230ELSEIFA$="S"THEN255ELSE240
240 PRINT:PRINT"AS YOU KNOW, DOMES ARE CLASSED AS EITHER 3/8, 1/2
     (OR 4/8), OR 5/8 DOMES.  BY USING THE NUMERATOR (USING 4 FOR
     1/2),
INDICATE THE FORM DESIRED:";F$
250 F$=INKEY$:IFF$=""THEN250ELSE260
255 F$="4"
260 F=VAL(F$):IFF=4THENZ=2ELSEZ=1.93649
270 IF F=3ORF=4ORF=5THEN280ELSE240
280 IF A$="D"THEN290ELSE300
285 'COMPUTATION OF "REAL-LIFE" DOME STATISTICS
290 D=A:R=D/2:AB=Z*R:AF=PI*((AB/2)[2):GOTO310
300 AF=A:AB=2*(SQR(AF/PI)):R=AB/Z:D=2*R
310 DH=D*(F/8):DS=2*PI*DH*R:DV=(PI*(DH[2)*((3*R)-DH))/3
320 CLS:PRINT"SUMMARY OF DOME STATISTICS..."
330 PRINT@192,"THESE FIGURES ARE FOR A"F;"/8THS GEODESIC DOME,";D;
     "FEET IN
DIAMETER.
340 PRINT:PRINT"HEIGHT OF DOME                =  ";DH;" FEET"
350 PRINT"SURFACE AREA OF DOME          =  ";DS;" SQUARE FEET"
360 PRINT"VOLUME OF DOME                =  ";DV;" CUBIC FEET"
370 PRINT"AREA OF FLOOR WITHIN          =  ";AF;" SQUARE FEET"
380 PRINT"RADIUS OF FLOOR               =  ";AB/2;" FEET"
390 PRINT:PRINT"'C' TO CONTINUE, 'R' TO READJUST INPUT DATA";
395 Q$=INKEY$:IFQ$=""THEN395ELSEIFQ$="R"THEN200ELSE400:'ROUTINE TO
      SET SCALE OF MODEL
400 CLS
410 PRINT@192,"NOW WE NEED THE SCALE OF YOUR MODEL.   SCALES ARE E
     XPRESSED
AS:      A  --   1 INCH SCALE = X FEET REAL

     B  --  THE MODEL IS A FRACTION OF THE REAL (1/2, 1/4)
420 PRINT:PRINT"SELECT THE TYPE SELECT YOU ARE READY TO INPUT:"
430 S$=INKEY$:IF S$=""THEN430ELSE440
440 IF S$="A"THEN450ELSE460
450 CLS:PRINT@192,"1 INCH IN THE SCALE WILL EQUAL HOW MANY FEET IN
     REAL LIFE?":INPUTS:SF=1/S:GOTO470
460 CLS:PRINT@192,"WHAT IS THE NUMERATOR OF YOUR SCALE FRACTION?":
```

Program continues

:::ULTRADOS:::ULTRADOS:::ULTRADOS:::ULTRADOS:::ULTRADOS:::

:::ANOUNCING THE LATEST REVISION!:::

:::FASTER AND EVEN MORE POWERFUL THAN BEFORE!:::

A TREMENDOUS USER RESPONSE AND FEEDBACK FROM THE FIRST 2000 COPIES ISSUED HAS PROVIDED THE VALUABLE INFORMATION WE NEEDED TO DETERMINE THOSE FEATURES MOST WANTED, THEY ARE INCLUDED IN THIS REVISION.

THE SUCCESS OF ULTRADOS IN THE MARKET PLACE ALSO ALLOWS US TO REDUCE IT'S PRICE, INSTEAD OF $119.95, THE FULL BLOWN VERSION IS YOURS FOR ONLY $94.95!

OTHER USER INPUT POINTS OUT THAT SOME OF OUR CUSTOMERS DO NOT NEED OR USE ALL OF THE SOPHISTICATED PROGRAMMING UTILITIES INCLUDED IN THE SYSTEM, CONSEQUENTLY WE ARE RELEASING A SECOND VERSION WITH SOME OF THEM REMOVED, AT JUST $59.95!

:::ALSO ANNOUNCING ULTRA UTILITIES:::

BOSS 2.2 - $24.95c $29.95d Basic operated single stepper, a monitor for basic programs, a must for all basic programmers.

VFU - $19.95c $24.95d Visual file utility, single key commands will COPY or PURGE disk files, EXECUTE will load and execute both machine language and basic programs, use the 4 arrow keys to position cursor to the selected file.

UZ - $19.95c $24.95d ULTRA ZAP the super disk zap utility from ULTRADOS.

SPOOL - $19.95c $24.95d Parallel port printer spooler, define the buffer size, lines per page, and characters per line, returns use of the CPU while printing is done. Don't confuse this spooler with others that use a disk file, this one spools into RAM memory!

GEDIT - $19.95c $24.95d Global Editor, that's right folks a global editor for basic programs! Allows fast global search and replace of anything in a Basic program, even key words, will create those super fast string packed graphics!

LC - $5.95c Lower case driver, supports all known LC mods, allows typewriter keyboard function, only 68 bytes long.

EP - $5.95c Epson MX-80 printer driver, allows you to use all the features of this fine little printer including graphics.

RENUM-80 - $5.95c Renumber for tape and disk systems, super fast, fail safe operation.

:::ULTRADOS:::ULTRADOS:::ULTRADOS:::ULTRADOS:::ULTRADOS:::

:::LAST MINUTE ITEMS OF INTEREST:::

*** ALL MPI DISK DRIVE PRICES REDUCED, CHECK IT OUT!!!

*** WE ARE SHIPPING MOD III's WITH DRIVES, DO YOU HAVE YOURS?

*** USED LEVEL II's, DRIVES, PRINTERS, ETC. IN STOCK!!!

*** WE HAVE THE BEST MOD III SOFTWARE SELECTION AROUND!

*** WE TAKE HARDWARE TRADE-INS, AND OFFER HIGHEST ALLOWANCES!

*** TWO TECHNICIANS ON STAFF FOR FAST RELIABLE LOW COST SERVICE

** YOUR ONE STOP COMPUTER SHOPPING SOURCE **

LEVEL IV PRODUCTS INC. ✓14
32461 SCHOOLCRAFT RD.
LIVONIA, MICHIGAN 48150
313-525-6200 (in state)
800-521-3305 (toll free)

LEVEL IV PRODUCTS

```
INPUTSN:PRINT:INPUT"WHAT IS THE DENOMINATOR OF THE FRACTION";
SD:SF=(12*SN)/SD:S=1/SF
465 'COMPUTATION OF MODEL STATISTICS
470 RM=R*SF:DM=D*SF:HM=(F/8)*DM:SM=2*PI*HM*RM:VM=(PI*(HM[2)*((3*RM
     )-HM))/3
480 RF=(Z*RM)/2:FM=PI*(RF[2)
490 CLS:PRINT"MODEL STATISTICS"
500 PRINT@192,"THESE FIGURES ARE FOR A MODEL OF A";F;"/8THS GEODES
     IC DOME.
THE DIAMETER OF THE MODEL DOME IS";DM;"INCHES.
SCALE
     USED TO OBTAIN THESE DATA: 1''=";S;"'"
510 PRINT
520 PRINT"HEIGHT OF THE DOME MODEL       =  ";HM;" INCHES"
530 PRINT"SURFACE AREA OF THE MODEL      =  ";SM;" SQUARE INCHES"
540 PRINT"VOLUME OF THE MODEL            =  ";VM;" CUBIC INCHES"
550 PRINT"AREA OF MODEL FLOOR            =  ";FM;" SQUARE INCHES"
560 PRINT"RADIUS OF MODEL FLOOR          =  ";RF;" INCHES
570 IF LP=1 THEN580ELSE680
580 LPRINT"UNIT                          REAL (IN FEET)          MO
     DEL  (IN INCHES)
590 B$="########.####          ########.####"
600 LPRINT" "
610 LPRINT"HEIGHT OF DOME                 ";:LPRINTUSINGB$;DH;HM
620 LPRINT"SURFACE AREA OF DOME           ";:LPRINTUSINGB$;DS;SM
630 LPRINT"VOLUME OF THE DOME             ";:LPRINTUSINGB$;DV;VM
640 LPRINT"AREA OF FLOOR                  ";:LPRINTUSINGB$;AF;FM
650 LPRINT"RADIUS OF FLOOR                ";:LPRINTUSINGB$;AB/2;RF
660 LPRINT" ":LPRINT"SCALE OF MODEL:  1 INCH SCALE =";S;"FEET REAL
     .":LPRINT" "
670 GOTO690
680 PRINT:INPUT"HIT 'ENTER WHEN READY TO PROCEED.";XX
690 CLS:PRINT"THIS PROGRAM IS ONLY DESIGNED TO GIVE YOU THE NECESS
     ARY VALUES
WITH WHICH TO BUILD A '3-FREQUENCY ALTERNATE' DOME
     .  IF YOU NEEDTO BUILD A DOME OF A DIFFERENT FREQUENCY, PLEAS
     E REFER TO PAGE
109 OF THE DOME BOOK."
700 PRINT:PRINT"FOR THIS DOME TYPE YOU NEED STRUTS OF THREE DIFFER
     ENT LENGTHS.
SHOWN BELOW ARE THE THREE LENGTHS AND THE NUMERI
     CAL VALUES, IN
DECIMAL INCHES.  THE SCALE USED IS 1''=";S;"'"
710 PRINT
715 'ROUTINE TO COMPUTE LENGTH OF STRUTS
720 Z=14.69573444
730 A=.34861548:B=.40354821:C=.41241149
740 AC=A*RM:BC=B*RM:CC=C*RM
750 AI=INT((AC*32)/2)+7:BI=INT((BC*32)/2)+7:CI=INT((CC*32)/2)+7
760 AK=INT(AC*16.6666666):BK=INT(BC*16.6666666):CK=INT(CC*16.66666
     66)
770 FF=1
775 'ROUTINE TO ADJUST DISPLAY TO FIT SCREEN
780 IF AI>115 OR BI>115 OR CI>115THEN810ELSE790
790 GOSUB870
800 GOTO970
810 AI=AI/2:BI=BI/2:CI=CI/2:FF=FF*2:AK=AK/2:BK=BK/2:CK=CK/2
820 GOTO780
825 'GRAPHIC DISPLAY OF STRUTS ON MONITOR
870 FOR I=8 TO AI
880     SET (I,34)
890 NEXT I
900 FOR J=8 TO BI
910     SET (J,40)
920 NEXT J
930 FOR K=8 TO CI
940     SET (K,46)
950 NEXT K
960 RETURN
970 IF FF>1THEN980ELSE1010
980 PRINT@576,"SCREEN TOO SMALL, CHORDS SHOWN ARE 1/";FF;"OF PROPE
     R SCALE SIZE."
990 IF LP=1THEN1000ELSE1010
1000 LPRINT"PAGE WIDTH TOO NARROW, CHORDS SHOWN ARE 1/";FF"OF PROP
     ER SCALE SIZE.":LPRINT" "
```

Program continues

```
1010 PRINT@705,"A";:PRINT@646,"CHORD 'A' =";AC;"INCHES";
1020 PRINT@833,"B";:PRINT@774,"CHORD 'B' =";BC;"INCHES";
1030 PRINT@961,"C";:PRINT@902,"CHORD 'C' =";CC;"INCHES";:IF LP=1TH
     EN1040ELSEINPUT"    'ENTER' TO CONTINUE.";XX
1040 IFLP<>1THEN1080ELSELPRINT" ":LPRINT"CHORD 'A' =";AC;"INCHES."
     :LPRINT"  A  ";:LPRINTSTRING$(AK,42):LPRINT" "
1050 LPRINT" ":LPRINT"CHORD 'B' =";BC;"INCHES.":LPRINT"  B  ";:LPR
     INTSTRING$(BK,42):LPRINT" "
1060 LPRINT" ":LPRINT"CHORD 'C' =";CC;"INCHES.":LPRINT"  C  ";:LPR
     INTSTRING$(CK,42)
1070 LPRINTSTRING$(5,10)
1080 IF LP<>1THEN1090ELSEINPUT"    'ENTER' TO CONTINUE.";XX
1090 GOTO1180
1100 CLS:PRINT@512,"(R)EPEAT TO SEE THESE DATA AGAIN, (A)NOTHER DO
     ME DATA SET,
OR (E)ND TO END PROGRAM."
1110 Q$=INKEY$:IFQ$=""THEN1110ELSE1120
1120 IF Q$="R"THEN490ELSEIFQ$="A"THEN200ELSEIFQ$="E"THEN1130ELSE11
     20
1130 CLS:PRINT@256,"GOOD LUCK WITH YOUR PROJECT.

REMEMBER THAT TH
     E VALUES SHOWN FOR THE STRUTS ARE TOTAL LENGTH
STRUTS.  IF YO
     U ARE USING HUBS, BE SURE TO SUBTRACT THE PROPER
AMOUNT FROM
     THE STRUT SO THE OVERALL LENGTH IS NOT GREATER THAN"

1140 PR
     INT"SH
OWN IN THIS PROGRAM."
1150 PRINT:PRINT"THANKS FOR USING GEODESICS."
1160 END
1170 'MEMORY REQUIRED IS ABOUT 8100 BYTES
1180 CLS
1190 GOSUB870
1195 'ROUTINE TO DRAWN BASIC TRIANGLE ON MONITOR
1200 FORY=4TO43
1210    X1=INT((Y-4)*1.2307694)
1220    X2=127-X1:X3=X1+32:X4=X2-32:X5=X1+64:X6=X2-64
1230    SET(X1+31,Y):SET(X2,Y)
1240    IFX3<(X2-31)SET(X3+31,Y)
1250    IFX5<(X2-31)SET(X5+31,Y)
1260    IFX4>(X1+31)SET(X4,Y)
1270    IFX6>(X1+31)SET(X6,Y)
1280 NEXTY
1290 Y=4:FORX=32TO127:SET(X,Y):NEXTX
1300 Y=17:FORX=48TO111:SET(X,Y):NEXTX
1310 Y=30:FORX=64TO95:SET(X,Y):NEXTX
1320 PRINT@24,"A";:PRINT@40,"B";:PRINT@56,"A";:PRINT@209,"A";:PRIN
     T@218,"B";:PRINT@229,"C";:PRINT@234,"C";:PRINT@245,"B";:PRINT
     @250,"A";
1330 PRINT@483,"C";:PRINT@492,"C";:PRINT@705,"A";:PRINT@833,"B";:P
     RINT@961,"C";
1340 PRINT@416,"C";:PRINT@431,"C";:PRINT@538,"B";:PRINT@566,"B";:P
     RINT@616,"B";:PRINT@801,"A";:PRINT@813,"A";
1350 PRINT@646,AC;" INCHES";:PRINT@774,BC;" INCHES";:PRINT@902,CC;
     " INCHES";
1360 PRINT@1000,"'@' FOR DRAFTING HELP";
1370 Q$=INKEY$:IFQ$=""THEN1370
1380 CLS
1390 CLS:PRINT@192,"ARE YOU CONVERTING FROM (S)CALE TO REAL

OR FR
     OM (R)EAL TO SCALE?";
1395 'DRAFTSMAN'S AID
1400 C$=INKEY$:IFC$=""THEN1400
1410 PRINT:PRINT:INPUT"ENTER KNOWN VALUE";KV
1420 IFC$="S"THEN1480
1430 PRINT:PRINT"THE SCALE VALUE OF";KV;" FEET  =";KV*SF;" INCHES"
1440 PRINT:PRINT"(G)RAPHICS, (E)ND,   OR  (C)ONVERSION HELP"
1450 S$=INKEY$:IFS$=""THEN1450
1460 IFS$="G"THEN1180ELSE1470
1470 IFS$="E"THEN1100ELSE1390
1480 PRINT:PRINT"THE REAL VALUE OF";KV;" INCHES =";KV/SF;" FEET"
1490 GOTO1440
1500 END
```


✓See List of Advertisers on page 306

Re-ink, and long line printer ribbons!

Keep It in the Black

William D. Johnston
1808 Pomona Dr.
Las Cruces, NM 88001

Microcomputer users who own hard copy printers have always appreciated the crisp and easy-to-read print produced when a new ribbon is installed.

The ability of a fabric printer ribbon to maintain this quality is really short-lived, however, since many ribbons have a useful life of no more than 16 to 20 hours. And the print quality, of course, deteriorates even before the ink is all gone.

I was doing some work one day on a microcomputer driving an LA36 DECwriter. The printer ran continuously. To keep up acceptable print quality, it was then necessary to change the ribbon at least once a day, sometimes twice. At $5.00 per ribbon, there had to be a better way.

My old Model 15 Teletype machine runs several hours a day, printing continuous wire service news bulletins, National Weather Service broadcasts and ham radio transmissions. In the past ten years I've changed the ribbon twice. Nevertheless, it always prints nice, dark characters.

The secret is a clever little device which automatically re-inks the ribbon as it advances from one spool to the other during normal printing. All that is necessary is to replace the ink supply once every few weeks. The nylon ribbon is replaced only when the fabric is worn out.

The re-inking device on the Model 15 was manufactured especially for that machine. As far as I know, no device has ever been made for microcomputer printers, so it was worth it to see if the Model 15's re-inker could work on the DECwriter. I just designed a mounting bracket that wouldn't interfere with the DECwriter's moving parts, and the old Model 15 re-inker worked perfectly.

Mounting and Operation

Photo 1 shows a typical re-inker kit before installation. Although several different models exist, they have a bracket with a small, permanently mounted ball-bearing assembly. An inkwell snaps into the bearing. The inkwell itself has a threaded cap, and the ink feeds by capillary action through tiny holes into the felt bands that surround it. The mechanism is mounted so that the ribbon passes around the inkwell and picks up the ink from the felts whenever the printer is in operation.

An actual installation is shown in Photo 2. The printer illustrated is an LA36 DECwriter, and the re-inker was mounted without drilling holes or making modifications to the printer. It can easily be restored to its original condition.

A thin metal bar was bolted to the original re-inker bracket/bearing assembly, and the bar, in turn, was clamped to the printer chassis. Fabrication and installation took less than ten minutes. Note that this is a slightly different model of re-inker kit than the one in Photo 1, and the mounting method must be altered, depending on the model used.

By making a suitable bracket, these re-inkers can be used on any printer that employs a standard-width fabric ribbon. Common examples are the DECwriter and the Teletype printers, but there are literally scores of other brands in the same category. My own modification worked so well that I helped to modify more than 200 other printers in hobby, commercial, and university installations.

Remember, the re-inker must *not* be placed between a ribbon spool and its corresponding ribbon-reversing lever. This would alter the length of the path between these two elements, and there would be insufficient ribbon length between the reversing rivet and the spool attachment point. This could cause the ribbon to jam before the reversing lever could be actuated, resulting in damage to the printer.

Filling the Inkwell

It is easy to over-ink a ribbon in your eagerness to get nice, dark print. The best policy is to fill the inkwell just before running a program that will produce a lot of print. This will ensure that the ribbon will move back and forth several times and become evenly saturated. Some experimenting may be necessary, due to varying ribbon lengths.

If your ribbon is old to start with, you may have to fill the inkwell more than once to saturate it, initially. After that, fill it just once whenever the print isn't as dark as you'd like. Be sure to give the ribbon time to make two or three complete passages before deciding to add more ink.

To fill the well, unscrew the cap and fill the reservoir within 1/8 inch (4 mm) of the rim. Then replace the cap. This can be done without taking the inkwell out of the printer. Use only NCR (National Cash Register) type K-575-F ink. This is a non-corrosive, thick, black ink that comes in a tube, and is available at NCR repair and sales offices.

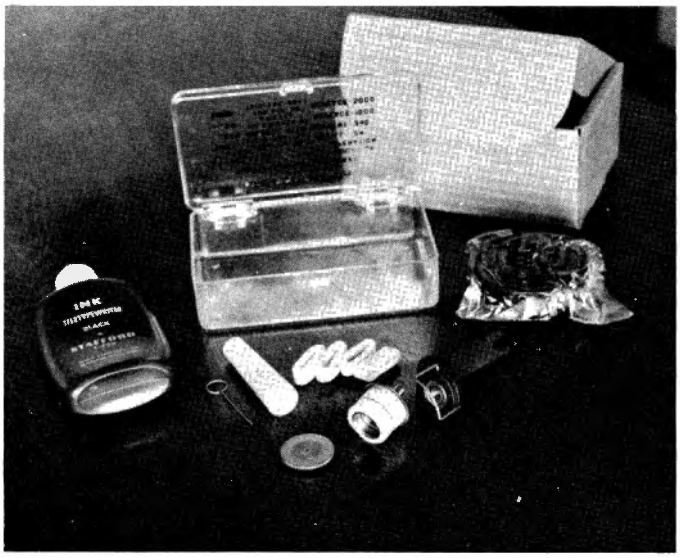


Photo 1. An automatic ribbon re-inker kit before mounting on a microcomputer printer. The main parts are the bracket/bearing assembly and the inkwell assembly. Also, extra felts, nylon ribbon, and special tools for maintenance.

One tube will last for years. This ink has the right consistency to flow properly in the re-inker, and it will not damage the machine's print head.

Where to Buy Re-Inkers

The automatic re-inker kits described are available in new, unused condition from surplus dealers and firms which handle Teletype equipment. As in Photo 1, the kit contains the bracket/bearing assembly, inkwell, extra felts, nylon ribbon (with spool), a bottle of ink, a tool for shaping the felts, and a tool for cleaning the feed holes in the inkwell. The first thing you must do is to throw out the bottle of ink. Most of these kits were manufactured more than 10 years ago, and the ink supplied is not satisfactory. Use only the NCR K-575-F ink.

If you can't locate a re-inker kit, I have a few left from the modification projects here and I will sell them on a first-come first-served basis for $18.25, which includes kit, postage and insurance. Add $1.00 for first class mail. Also include a self-addressed, stamped envelope; I will return your money if all kits have been sold.■

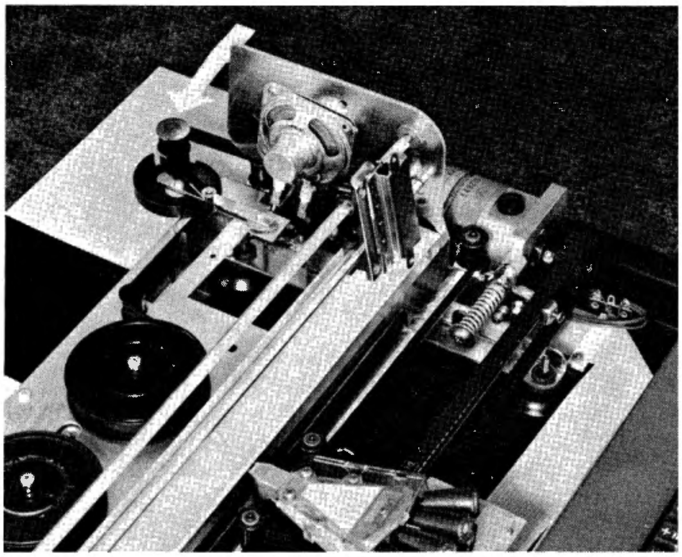


Photo 2. This shows the installation of a surplus re-inker kit in an LA36 DECwriter. Crisp, dark print is produced continuously, 24 hours a day. Just refill the inkwell once every few days.

A software tool to retrieve the irretrievable.

Direct Statement in File

D.E. Fitchhorn
3504 Piermont Dr. N.E.
Albuquerque, NM 87111

You've tried to load a BASIC program file from disk and the error message DIRECT STATEMENT IN FILE appears on the screen. When you don't have a backup copy, what do you do?

Usually, you have to start over, retyping the program from written copy. Next, you edit for the latest changes and reSAVE it. You face the same problem when you move one or more lines from one place to another in a program.

DIRECT STATEMENT IN FILE doesn't mean that the total file has been lost. It does mean that at least one line in the file doesn't have a line number. Even though a program can't be loaded under BASIC with this error, the whole program file can be listed at DOS level as it existed on disk.

If the file is intact and can be read by DOS, then the file can be salvaged by inserting a properly sequenced line number or obviating the need for a line number. The BASIC editor is of no use since it can't edit a program line that can't be loaded, and will not permit changes to be made to a line number.

Easy Repair

There is an easy way to repair these problems in a program saved as an ASCII file. A program that will read the BASIC program file as if it were a random access data file will also permit you to correct direct statement errors and change a line number.

Program DSKREP/BAS does this, reading BASIC files sector by sector like random access files, and reporting on the contents of each sector, byte by byte. It provides the option of listing the sector data in hex or ASCII form.

When a BASIC program fails to load completely because of a direct statement error, part of the file has been loaded into memory and that portion can be listed uncovering the line number of the last good statement in file. Once that line number has been located, the problem line can be located in the sector data. You will know the number of sectors in the file from the DIR data and the approximate location of the error, and can estimate the number of the sector containing the error. Observing the data printed for that sector helps pinpoint the sector containing the error. Data following the carriage return code at the end of the last good line normally contains the error.

DSKREP/BAS searches the computer sector by sector, for an ASCII string (a line number or another ASCII string clue) that you select. The program then prints the data in that sector.

The edit function accesses each byte in a sector and replaces the contents of that byte with the desired code. If a line number is changed, loading later under BASIC will rearrange the code in memory to place the line in proper position.

How it works

DSKREP/BAS first asks for the name of the file to be repaired. The program asks whether you want to locate the problem area using an ASCII string clue or by the number of the sector, for which the program then asks.

After the sector data has been printed, the program asks if you want to make a change. Y causes the program to continue the edit function, while N reverts the program to the sector number request routine.

The program next asks for the number of the byte to be changed, which should be answered with its decimal number (0 to 255). The program will print, if printable, the character in that byte. The next query asks for the ASCII character to replace the present character. This can be answered with an ASCII character or a decimal number of an ASCII character for characters not acceptable in a string variable (carriage return, etc). This operation can be repeated until all necessary changes are made.

After all changes have been made, answer the byte number query with a number greater than 255 to save the changes and return to the main program. If the sector number query is answered with a 0, the program reverts to the filespec level. Answer END to the query; FILESPEC ends the program run.■

Program Listing

```
240 ' * INITIALIZE
260 '
280 CLEAR 2000:CLS:DEFINT I-N
300 DIM H$(15),DA$(260)
320 FOR I = 0 TO 15:READ H$(I):NEXT I
340 CLS:PRINT @15,"ASCII FILE SECTOR REPAIR UTILITY "
360 '
380 ' * GET FILENAME AND OPEN FILE
400 '
420 LINEINPUT"
FILENAME: ";FS$:IF FS$ = "END" THEN END
     ELSE ON ERROR GOTO 1600  :OPEN"I",1,FS$
440 ON ERROR GOTO 0     :CLOSE:OPEN"R",1,FS$:SN = 0:S$ =
    ""
460 FOR J = 0 TO 255:FIELD 1,(J*1) AS DUMY$,1 AS DA$(J)
    :NEXT J
480 FIELD 1,255 AS DU$,1 AS DD$
500 '
520 ' * DECIDE HOW TO LOCATE SECTOR
540 '
560 SL$="":INPUT"
LOCATE SECTOR BY SEARCH STRING OR SEC
     TOR NUMBER?? (S OR N) ";SL$
580 IF SL$="E" THEN CLOSE:GOTO 340
600 IF SL$<>"S" AND SL$<>"N"GOTO560
620 IF SL$="S"GOTO 820
640 '
660 ' * LOCATE BY SECTOR NUMBER
680 '
700 S$="":INPUT"
SECTOR NUMBER (OR 'ENTER' FOR NEXT SEC
     TOR): ";S$
720 IF S$ = ""THEN SN = SN + 1 ELSE SN = VAL(S$):IF SN
    < 1 THEN CLOSE: GOTO 340
740 GOTO1020
760 '
780 ' * LOCATE BY ASCII STRING
800 '
820 SS$="":INPUT"
STRING TO SEARCH FOR?  ";SS$
840 FOR I  = 1 TO LOF(1):GET 1,I
860 Z = INSTR(DU$,SS$)
880 IF Z <> 0 GOTO 920
900 NEXT I
920 SN = I
940 IF Z = 0 THEN PRINT "STRING NOT FOUND !!":GOTO 820

960 '
980 ' * GET PROPER SECTOR
1000 '
1020 GET1,SN
1040 '
1060 ' * PRINT SECTOR INFORMATION
1080 '
1100 CLS:PRINT@64,"FILE : "FS$;TAB(35)"SECTOR: "SN
1120 L1=-4:J1=-1:J2=-1
1140 FOR I = 0 TO 3: L1 = L1+4
1160 PRINT@255,"":FOR L=0TO3:PRINT(L1+L)*16;TAB(6);:FOR
     K=1TO16:J1 =J1+1
1180 V=ASC(DA$(J1)):MN=INT(V/16):LN=V-MN*16:PRINT H$(MN
     )+H$(LN)+" ";
1200 NEXTK:PRINT
1220 PRINT TAB(7);:FOR K=1TO16:J2=J2+1:IFDA$(J2)>" "AND
     DA$(J2)<"Z"THENPRINTDA$(J2);"  ";:ELSEPRINT".  ";:
1240 NEXTK:PRINT
1260 NEXTL:PRINT @1000,"CONTINUE  ";:INPUTYN$
1280 CLS:IF I > 2 GOTO 1300 ELSE PRINT@64,"FILE :  "FS$
     ;TAB(35)"SECTOR :  "SN
1300 NEXTI:GOTO1380
1320 '
1340 ' * CHANGE DECESSIONS
1360 '
1380 YN$="":INPUT"DO YOU WANT TO MAKE A CHANGE  ?";YN$:
     IFYN$<>"Y"GOTO560
1400 INPUT"# OF THE BYTE YOU WANT TO CHANGE?";X:IFX>255
     THEN1560
1420 X$=DA$(X+1):PRINT X$:INPUT"CHANGE TO CHARACTER?";X
     $
1440 IF LEN(X$)=1 GOTO 1480
1460 X$ = CHR$(VAL(X$))
1480 LSET DA$(X+1)=X$:GOTO 1400
1500 '
1520 ' * CHANGE THE FILE
1540 '
1560 PUT1,SN:GOTO340
1580 DATA "0","1","2","3","4","5","6","7","8","9","A","
     B","C","D","E","F"
1600 PRINT:TAB(10)" - FILE NOT FOUND - ":RESUME 340
```


See List of Advertisers on page 306

If you've ever bought a piece of cassette based software for your disk system. . . read on.

Hardtimes

Joe Fettig
74-355 Buttonwood
Palm Desert, CA 92260

After owning my TRS-80 for a few months, I find it's essential to have a disk storage unit to really enjoy it. Tape cassette storage and access are just too slow and unreliable, leaving me frustrated and ready to sell my system.

I finally bought a disk drive unit and promised myself that I would never buy anything on tape cassette again. Then I ran across a utility called a variables map (variables cross-index reference).

The variables map is an invaluable tool that gives you a list of all the variables used in a program and their line locations. When dealing with long programs, it's nice to know which variables are already in use. This can eliminate using the same variable for two different functions in the same program.

The utilities package in which this debugging aid is available is The Bottom Shelf's *Basic Toolkit* by Barry Mulligan.

I went through the usual two-hour session of trying to get the tape to load, without success. A few hours later it magically loaded and ran. Then all I had to do was use my monitor to get the start and end location and the entry point into the program. It wasn't easy.

The Toolkit has a machine language program that loads and then automatically takes control of the computer, giving an error message on the monitor. The monitor is then supposed to give you the entry point. It failed! (I did notice a card with the Toolkit saying that you could send an additional $10.00 for a disk version.)

Back to Disk

At that point, I brought my project to a screeching halt and put it on disk. If it loaded into memory, it should load onto disk. Using DEBUG I determined that the program was located above memory location 46500 (decimal) in the 32K version I was using.

Using the TAPEDISK F command, I dumped the program onto disk, starting at B5A4H and ending at BFC0. The program was now on disk and would load into memory, but would not execute and I did not know the entry point of the program in memory.

Back to the original tape copy. I loaded it with the monitor again and found that the program was actually two program segments. The first segment loaded into memory, took control, and then loaded the second, main program.

Using DEBUG, I was able to determine exactly what was happening. The first program loaded, and then changed the keyboard control device location at 4016H and 4017H. This was the key to gaining entry into the program. All that was necessary was to load Toolkit when in DOS READY and then execute BASIC (be sure to protect memory).

In BASIC, execute the following program: 10 POKE &H4016, &HDF : POKE &H4017, &HB7.

This changes the keyboard device control so that it jumps to B7DFH instead of its usual location, and then looks for a shift break key. When the shift break key is pressed, the Toolkit program should execute in its normal manner.

The above will work on a 32K TRS-80. The general procedure is outlined below:

- Execute BASIC: Make sure you protect memory.
- Type CMD"T": This stops the clock.
- Type SYSTEM, and load the Toolkit program from tape. (You can test if Toolkit is working by pressing the shift and break keys.)

The next two steps are for 16K and 48K users:

- Type: Print PEEK(&H4016) <ENTER>
  * Record this value
- Type: Print PEEK(&H4017) <ENTER>
  * Record this value

- Press the reset button: This will put you in DOS READY.
- Type TAPEDISK: <ENTER> This loads the Tapedisk program.

● Use the F command listed under TAPEDISK commands. Refer to the table below for your system configuration:

MEMORY SIZE	START	END
16K	7530H	7FC0H
32K	BFA4H	BFC0H
48K	F618H	FFC0H

Use 402DH for the entry.

● After the program has been put on disk, to execute it do the following:

A With DOS READY, load the CMD file into memory.

B Type LOAD "filename", filename being what you called the program when saving it under TAPEDISK.

C Execute BASIC. Protect memory for the program with the size for your system.

● 32K users, execute the BASIC program listed above. Users of other systems (16K and 48K), see the next step.

● 32K users, go to the next step. All others: The BASIC program above will have to be changed to work with your system. The program changes the keyboard device control by POKEing two memory locations with different values. When a shift break is encountered, the Toolkit program then intercepts them. To do this, take step 4A and change the POKE statement in the BASIC program from POKE &H4016, &HDF to POKE &H4016, XXXX where XXXX = value obtained in step 4. Now change the POKE &H4017, &HB7 to POKE &H4017, YYYY where YYYY = value obtained in step 4B.

● Now execute the BASIC program for 32K users or the modified program from step 10 for 16K and 48K users.

● Press the shift and break keys and the Toolkit menu should show up.

The BASIC TOOLKIT is a worthwhile investment for disk users as well as non-disk users. The variables map is a must for the serious programmer. ■

✓See List of Advertisers on page 306

Got a touch of dance fever?
Then let your 80 show you some night moves.

Dancin! —A Disco Primer

Andrew A. Modla
5 Derby Place
Newtown, PA 18940

Rev up your floppies and let's disco! Here's a routine to use at your next party. Written for the TRS-80, 16K Level II BASIC machine, this disco dance demonstration program is easy to apply.

A number of people have asked my wife and I to teach them disco, ever since we began dance lessons two years ago. I got the computer into the act because it's accurately timed.

Since much disco dance music utilizes electronic bass and percussion instruments to establish a tempo, the beat won't vary much. The computer can follow along quite well.

Three Steps

Three dance steps—Latin Hustle, Disco Hustle and Rope Hustle—are demonstrated with footsteps that move about on the video screen in time to the music. Once synchronized with the music, the footsteps keep time. Do this by entering the rate in beats per minute, while the music is playing, and by then starting the dance on a beat.

I used a stopwatch to determine the number of beats per minute. Calculate the number of beats per minute by dividing the number of beats by the number of seconds timed by a stopwatch. Multiply by 60. Average about six samples to get an accurate number.

I wrote the time-critical part of the program, the footstep display, in machine language called from BASIC. This is the only way to maintain accurate timing of the footsteps. The rest of the program is written in Level II BASIC. The machine language subroutine is POKEd into the high order part of memory.

The program has built-in explanations and instructions, so that a separate instruction sheet is not needed. Some of the built-in features are: selections of either the woman's or man's footsteps for display, footsteps displayed one at a time, stop action on the dance steps, and restart from the first step.

The machine language subroutine for displaying the footsteps is initialized by POKEing at location 32256 the instructions that move the footstep patterns to the display RAM memory. Also initialized are the footstep patterns and control pointers. The subroutine at lines 8800 through 9660 sets up the machine language subroutine.

The text explanations and prompting are shown following initialization. These are done by subroutines starting at lines 6000, 6240, 6370 and 6560.

The dance selected is read from data statements at the end of the program. These statements are instructions about where to place the steps, how much time to delay between steps, and where to go for the next step.

The instructions consist of two parts: the first is a descriptive word indicating a pattern or command, and the second is the screen location for displaying the footstep, or just a number for the next instruction. You may recognize this as an interpretive programming technique. The instructions are converted into numeric code and POKEd into a table area to be accessed by the machine language subroutine. Reading and conversion of the data statements is performed by the subroutine from line 8000 through 8530. The table is stored starting at location 31232.

Patterns are stored starting at location 31744.

The subroutine at line 7000 computes the time delay count used in the machine language subroutine. Three counts are set up: full beat, half beat, and a quarter beat, all stored from locations 32496 to 32501.

On lines 2200, 2230, and 2240, PEEKs are used to read the keyboard RAM area because the BASIC INKEY$ command does not permit checking a key off after returning a key value.

The following variables are used in the program:

D$	Dance name
K$	Input key
I	Index, dummy variable
X	Address pointer, dummy variable
B	Beats per minute
B#	Beats per minute double precision
S	Step rate, one at a time = 1 continuous = 2
G	Address pointer
H	Address pointer
N	Dance name code: 1,2,3
C	Beats per minute conversion to time delay
C#	Beats per minute conversion to time delay
Y	Pattern address pointer
T	Display RAM address pointer
Z	256

M Woman = 1 Man = 2
R Number of END commands to skip while reading DATA statements

Easy Demonstration

The dance steps of this program were chosen so that the footsteps are oriented upward. This restriction makes it easier to program the dance step demonstrator. However, since many dances use side and turning footstep motions, a more elaborate program is necessary here. All the basic programming principles used can then be applied to more complex dances, both disco and ballroom.

Also note that only the basic steps of a dance are shown. There are many variations of the basic footworks, but here the footstep display routines are not expanded to handle complex motions (by adding extra footprint patterns). Arm and body dance motions are thus left to your imagination. ■

Program Listing

```
10 REM DISCO DANCE STEP DEMONSTRATOR
20 DEFINT A-Z
30 Z=256
40 GOSUB8810:GOSUB6000
50 GOSUB6240
55 GOSUB6370:GOSUB6560:GOTO2070
60 CLS:PRINT@25,D$:POKE32258,0:POKE32259,122
70 I=USR(0)
2000 IFS=2THEN2200
2010 K$=INKEY$:IFK$=""THEN2010
2020 IFK$=CHR$(31)THEN60
2030 IFK$=CHR$(9)THEN55
2040 IFK$=CHR$(91)THEN50
2050 IFK$=" "THEN70
2060 GOSUB6560
2070 K$=INKEY$:IFK$=""THEN2070
2075 IFK$=" "THEN60
2080 IFK$=CHR$(31)THEN60
2090 IFK$=CHR$(9)THEN55
2100 IFK$=CHR$(91)THEN50ELSE2070
2200 K$=INKEY$:X=PEEK(14400):IFX=128THEN2230
2220 IFX=2THEN2240ELSE2020
2230 X=PEEK(14400):IFX=0THEN70ELSE2230
2240 X=PEEK(14400):IFX=0THEN60ELSE2240
6000 CLS:PRINTTAB(15);"DISCO DANCE STEP DEMONSTRATOR"
6020 PRINT:PRINTTAB(15);"COPYRIGHT 1979 ANDREW A. mODLA"
6030 PRINT:PRINT
6040 PRINT"THE DISCO DANCE STEP DEMONSTRATOR SHOWS WITH DANCING FE
ET"
6050 PRINT"THE BASIC FOOTWORK FOR POPULAR DISCO DANCES. IT CUSTOMI
ZES"
6060 PRINT"THE DEMONSTRATION FOR EITHER A MAN OR WOMAN'S DANCE STE
PS."
6070 PRINT"IT ALLOWS YOU TO SET THE SPEED OF THE DANCING FEET FOR
"
6080 PRINT"SYNCHRONIZATION WITH MUSIC. IT CAN ALSO TAKE YOU THROUG
H THE"
6090 PRINT"BASIC FOOTWORK STEP BY STEP.
6140 PRINT:PRINT:PRINT@960,"PRESS ANY KEY TO CONTINUE";
6150 K$=INKEY$:IFK$=""THEN6150
6160 CLS:PRINT"THE FOLLOWING DIAGRAMS SYMBOLIZE THE FOOTSTEPS:"
6162 PRINT:PRINTTAB(19);"MAN";TAB(42);"WOMAN"
6165 PRINT
6170 PRINTTAB(14);"LEFT";TAB(23);"RIGHT";TAB(38);"LEFT";TAB(47);"R
IGHT"
6200 PRINT@704,"A MOVING FOOTSTEP INDICATES A WALKING STEP."
6203 PRINT@748,"A FLASHING FOOTSTEP"
6205 PRINT@768,"IN PLACE INDICATES A WALK IN PLACE."
6206 PRINT@804,"A FOOTSTEP WITHOUT THE HEEL"
6208 PRINT@832,"INDICATES A TAP STEP USING THE BALL OF THE FOOT."
6210 PRINT@960,"PRESS ANY KEY TO CONTINUE";
6215 POKEG,126:POKEH,208:I=USR(0)
6220 POKEH,212:I=USR(0)
6225 POKEH,216:I=USR(0)
6230 POKEH,220:I=USR(0):POKE16526,0:POKE16527,126
6235 K$=INKEY$:IFK$=""THEN6235ELSERETURN
6240 CLS:PRINT"SELECT ONE OF THE FOLLOWING DISCO DANCES:":PRINT
6250 PRINT"1  LATIN HUSTLE":PRINT"2  DISCO HUSTLE":PRINT"3  ROPE H
USTLE"
6260 PRINT:PRINT"PRESS KEY 1,2, OR 3 TO MAKE YOUR SELECTION"
6270 K$=INKEY$:IFK$=""THEN6270
6280 IFK$="1"THEN6290ELSEIFK$="2"THEN6290ELSEIFK$="3"THEN6290ELSE6
270
6290 N=VAL(K$)
6291 ONNGOTO6292,6293,6295
6292 D$="LATIN HUSTLE":GOTO6300
6293 D$="DISCO HUSTLE":GOTO6300
6295 D$="ROPE HUSTLE":GOTO6300
```

Program continues

```
6300 CLS:PRINT"THE DISPLAY WILL SHOW FOOTSTEPS FOR EITHER THE MAN
     OR WOMAN."
6310 PRINT"SELECT FOOTWORK FOR:"
6320 PRINT:PRINT"1  WOMAN":PRINT"2  MAN"
6330 PRINT:PRINT"PRESS KEY 1 0R 2 TO CONTINUE"
6340 K$=INKEY$:IFK$=""THEN6340
6350 IFK$="1"THEN6360ELSEIFK$="2"THEN6360ELSE6340
6360 M=VAL(K$)
6363 PRINT:PRINT:PRINT"LOOKING UP DANCE IN MEMORY.......";:GOSUB80
     00:RETURN
6370 CLS:PRINT"THE DANCE STEPS CAN BE DISPLAYED ONE AT A TIME AT Y
     OUR COMMAND"
6380 PRINT"OR CONTINUOUSLY WITH ANIMATION."
6390 PRINT:PRINT"1  DANCE STEPS CHANGE ONE AT A TIME"
6400 PRINT"2  DANCE STEPS CHANGE CONTINUOUSLY"
6410 PRINT:PRINT"PRESS KEY 1 OR 2 TO CONTINUE"
6420 K$=INKEY$:IFK$=""THEN6420
6430 IFK$="1"THEN6440ELSEIFK$="2"THEN6440ELSE6420
6440 S=VAL(K$)
6445 B=1:POKE32448,S-1
6450 IFS=1THEN6555
6460 CLS:PRINT"THE ANIMATED STEPS CAN BE SYNCHRONIZED WITH MUSIC T
     O HELP YOU"
6470 PRINT"FOLLOW ALONG. YOU SELECT THE RATE THE FOOTPRINTS CHANGE
      ON THE"
6480 PRINT"DISPLAY IN BEATS PER MINUTE. THE RATES RANGE FROM 85 TO
      150"
6485 PRINT"BEATS PER MINUTE."
6490 PRINT:PRINT"TYPICAL RATES FOR SELECTED MUSIC ARE:"
6500 PRINT:PRINT"112  THE HUSTLE (VAN MCCOY)"
6505 PRINT"126  LE FREAK (CHIC)"
6510 PRINT"136  MACHO MAN (VILLAGE PEOPLE)
6515 PRINT"140  SUNSET PEOPLE (DONNA SUMMER)":PRINT
6530 PRINT"HOW MANY BEATS PER MINUTE";
6540 INPUT B
6550 IFB<85THEN6530ELSEIFB>150THEN6530
6553 GOSUB7000
6555 RETURN
6560 CLS:PRINT"THESE KEYS LET YOU CONTROL THE ACTION:"
6570 PRINT:PRINT"CLEAR","START DANCE STEPS FROM BEGINNING"
6580 PRINTCHR$(94),"TIMING CHANGE"
6590 PRINT"SPACE","START/STOP/NEXT STEP"
6600 PRINTCHR$(91),"SELECT A NEW DANCE"
6610 PRINT:PRINT"ANY OTHER KEY WILL GIVE YOU THIS MENU AGAIN."
6620 PRINT:PRINT:PRINT"PRESS ONE OF THE ABOVE KEYS TO CONTINUE":RE
     TURN
7000 B#=B
7010 C#=((106445000./B#)-1941.)/20.
7020 IFC#>32767THENC#=-(65536-C#)
7030 C=C#:IFC>0THENPOKE32496,C/Z ELSEPOKE32496,INT(C/Z+256)
7040 POKE32497,C-Z*INT(C/Z)
7050 C#=((53222500./B#)-1941.)/20.
7060 IFC#>32767THENC#=-(65536-C#)
7080 C=C#:IFC>0THENPOKE32498,C/Z ELSEPOKE32498,INT(C/Z+Z)
7090 POKE32499,C-Z*INT(C/Z)
7100 C#=((26611250./B#)-1941.)/20.
7110 IF C#>32767 THEN C#=-(65536-C#)
7120 C=C#:IFC>0 THEN POKE 32500,C/Z ELSE POKE 32500,INT(C/Z+256)
7130 POKE32501,C-Z*INT(C/Z):RETURN
8000 R=2*N+M-3
8010 RESTORE
8020 IF R=0THEN8050
8030 READK$,T:IFK$<>"END"THEN8030
8040 R=R-1:GOTO8020
8050 X=31232
8060 READK$,T
8070 IFK$="MLF1"THENY=31744:GOTO8500
8080 IFK$="MRF1"THENY=31762:GOTO8500
8090 IFK$="MLT1"THENY=31780:GOTO8500
8100 IFK$="MRT1"THENY=31798:GOTO8500
8110 IFK$="WLF1"THENY=31816:GOTO8500
8115 IFK$="WRF1"THENY=31834:GOTO8500
8120 IFK$="WLT1"THENY=31852:GOTO8500
8130 IFK$="WRT1"THENY=31870:GOTO8500
8140 IFK$="ERASE1"THENY=31888:GOTO8500
8150 IFK$="DELAY1"THENY=32496:GOTO8525
8151 IFK$="DELAY2"THENY=32498:GOTO8525
8152 IFK$="DELAY4"THENY=32500:GOTO8525
8160 IFK$="SPACE"THENY=31955:GOTO8500
8170 IFK$="ONE"THENY=31906:GOTO8500
8180 IFK$="TWO"THENY=31913:GOTO8500
8190 IFK$="THREE"THENY=31920:GOTO8500
8200 IFK$="FOUR"THENY=31927:GOTO8500
8210 IFK$="FIVE"THENY=31934:GOTO8500
8220 IFK$="SIX"THENY=31941:GOTO8500
8230 IFK$="AND"THENY=31948:GOTO8500
8480 IFK$="END"THENRETURN
```

Program continues


```
8490 PRINT"ERROR IN COMMANDS 8490":END
8500 POKEX,Y/Z:X=X+1:POKEX,Y-Z*INT(Y/Z):X=X+1
8505 T=T+15360
8510 POKEX,T/Z:X=X+1:POKEX,T-Z*INT(T/Z):X=X+1:GOTO8060
8525 T=4*T+31232
8530 POKEX,Y/Z:X=X+1:POKEX,Y-Z*INT(Y/Z):X=X+1:GOTO8510
8800 REM SET UP MACHINE LANGUAGE SUBROUTINE
8810 X=32256
8820 POKEX,221:POKEX+1,33:POKEX+2,0:POKEX+3,122:POKEX+4,205:POKEX+
     5,48
8825 POKEX+6,126:X=X+7
8830 POKEX,205:POKEX+1,48:POKEX+2,126:POKEX+3,205:POKEX+4,48:POKEX
     +5,126:X=X+6
8840 POKEX,58:POKEX+1,192:POKEX+2,126:POKEX+3,254:POKEX+4,0
8845 POKEX+5,202:POKEX+6,156:POKEX+7,126:X=X+8
8850 POKEX,58:POKEX+1,255:POKEX+2,56:POKEX+3,254:POKEX+4,0
8855 POKEX+5,194:POKEX+6,156:POKEX+7,126:X=X+8
8860 POKEX,205:POKEX+1,112:POKEX+2,126:POKEX+3,58:POKEX+4,255
8863 POKEX+5,56:POKEX+6,254:POKEX+7,0:POKEX+8,194:POKEX+9,43
8866 POKEX+10,126:POKEX+11,195:POKEX+12,4:POKEX+13,126:POKEX+14,20
     1
8870 X=32300:POKEX,221:POKEX+1,33:X=32304
8880 POKEX,221:POKEX+1,70:POKEX+2,0:POKEX+3,221:POKEX+4,35
8885 POKEX+5,221:POKEX+6,78:POKEX+7,0:X=X+8
8890 POKEX,221:POKEX+1,35:POKEX+2,221:POKEX+3,102:POKEX+4,0
8895 POKEX+5,221:POKEX+6,35:POKEX+7,221:POKEX+8,110:POKEX+9,0:X=X+
     10
8900 POKEX,221:POKEX+1,35:POKEX+2,10:POKEX+3,87:POKEX+4,50:POKEX+5
     ,193
8905 POKEX+6,126:POKEX+7,3:POKEX+8,10:POKEX+9,95:POKEX+10,3:X=X+11
8910 POKEX,10:POKEX+1,119:POKEX+2,3:POKEX+3,35:POKEX+4,21
8915 POKEX+5,194:POKEX+6,77:POKEX+7,126:X=X+8
8920 POKEX,58:POKEX+1,193:POKEX+2,126:POKEX+3,87:POKEX+4,62:POKEX+
     5,60:POKEX+6,133:POKEX+7,111:POKEX+8,62:POKEX+9,0:POKEX+10,14
     0:POKEX+11,103:POKEX+12,29:POKEX+13,194:POKEX+14,77:POKEX+15,
     126:POKEX+16,201
8922 POKEX+6,133:POKEX+7,111:POKEX+8,62:POKEX+9,0:POKEX+10,140
8925 POKEX+11,103:POKEX+12,29:POKEX+13,194:POKEX+14,77:POKEX+15,12
     6
8930 POKEX+16,201:X=32368
8940 POKEX,221:POKEX+1,102:POKEX+2,0:POKEX+3,221:POKEX+4,35
8945 POKEX+5,221:POKEX+6,110:POKEX+7,0:POKEX+8,221:POKEX+9,35:X=X+
     10
8950 POKEX,70:POKEX+1,35:POKEX+2,78:POKEX+3,62:POKEX+4,0:POKEX+5,1
     1
8955 POKEX+6,184:POKEX+7,194:POKEX+8,127:POKEX+9,126:X=X+10
8960 POKEX,11:POKEX+1,185:POKEX+2,194:POKEX+3,132:POKEX+4,126:POKE
     X+5,195:POKEX+6,160:POKEX+7,126
8970 POKEX+5,195:POKEX+6,160:POKEX+7,126:X=32412
8975 POKEX,221:POKEX+1,35:POKEX+2,221:POKEX+3,35:X=X+4
8980 POKEX,221:POKEX+1,102:POKEX+2,0:POKEX+3,124:POKEX+4,50
8982 POKEX+5,3:POKEX+6,126:POKEX+7,221:POKEX+8,35:X=X+9
8985 POKEX,221:POKEX+1,110:POKEX+2,0:POKEX+3,125:POKEX+4,50:POKEX+
     5,2
8987 POKEX+6,126:X=X+7
8990 POKEX,229:POKEX+1,221:POKEX+2,225:POKEX+3,201
9000 REM SET UP POINTERS
9010 H=32302:G=32303:POKE16526,44:POKE16527,126
9020 X=32464
9030 POKEX,124:POKEX+1,0:POKEX+2,61:POKEX+3,142:POKEX+4,124
9035 POKEX+5,18:POKEX+6,61:POKEX+7,152:X=X+8
9040 POKEX,124:POKEX+1,72:POKEX+2,61:POKEX+3,166:POKEX+4,124:POKEX
     +5,90
9045 POKEX+6,61:POKEX+7,176
9050 REM SET UP FOOTSTEP PATTERNS
9060 X=31744:POKEX,4:POKEX+1,4:X=X+2:
9070 POKEX,184:POKEX+1,191:POKEX+2,191:POKEX+3,189
9080 POKEX+4,143:POKEX+5,143:POKEX+6,143:POKEX+7,135
9090 POKEX+8,176:POKEX+9,176:POKEX+10,176:POKEX+11,144
9100 POKEX+12,139:POKEX+13,143:POKEX+14,143:POKEX+15,129
9110 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9120 POKEX,190:POKEX+1,191:POKEX+2,191:POKEX+3,180
9130 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,143
9140 POKEX+8,160:POKEX+9,176:POKEX+10,176:POKEX+11,176
9150 POKEX+12,130:POKEX+13,143:POKEX+14,143:POKEX+15,135
9160 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9170 POKEX,184:POKEX+1,191:POKEX+2,191:POKEX+3,189
9180 POKEX+4,143:POKEX+5,143:POKEX+6,143:POKEX+7,135
9190 POKEX+8,32:POKEX+9,32:POKEX+10,32:POKEX+11,32
9200 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9210 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9220 POKEX,190:POKEX+1,191:POKEX+2,191:POKEX+3,180
9230 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,143
9240 POKEX+8,32:POKEX+9,32:POKEX+10,32:POKEX+11,32
9250 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9260 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9270 POKEX,160:POKEX+1,190:POKEX+2,189:POKEX+3,144
```

Program continues

```
9280 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,129
9290 POKEX+8,32:POKEX+9,176:POKEX+10,32:POKEX+11,32
9300 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9310 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9320 POKEX,184:POKEX+1,191:POKEX+2,180:POKEX+3,32
9330 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,129
9340 POKEX+8,32:POKEX+9,160:POKEX+10,144:POKEX+11,32
9350 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9360 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9370 POKEX,160:POKEX+1,190:POKEX+2,189:POKEX+3,144
9380 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,129
9390 POKEX+8,32:POKEX+9,32:POKEX+10,32:POKEX+11,32
9400 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9410 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9420 POKEX,184:POKEX+1,191:POKEX+2,180:POKEX+3,32
9430 POKEX+4,139:POKEX+5,143:POKEX+6,143:POKEX+7,129
9440 POKEX+8,32:POKEX+9,32:POKEX+10,32:POKEX+11,32
9450 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9460 X=X+16:POKEX,4:POKEX+1,4:X=X+2
9470 POKEX,32:POKEX+1,32:POKEX+2,32:POKEX+3,32
9480 POKEX+4,32:POKEX+5,32:POKEX+6,32:POKEX+7,32
9490 POKEX+8,32:POKEX+9,32:POKEX+10,32:POKEX+11,32
9500 POKEX+12,32:POKEX+13,32:POKEX+14,32:POKEX+15,32
9510 X=X+16:POKEX,5:POKEX+1,1:X=X+2
9520 POKEX,32:POKEX+1,79:POKEX+2,78:POKEX+3,69:POKEX+4,32:X=X+5
9525 POKEX,5:POKEX+1,1:X=X+2
9530 POKEX,32:POKEX+1,84:POKEX+2,87:POKEX+3,79:POKEX+4,32:X=X+5
9540 POKEX,5:POKEX+1,1:X=X+2
9550 POKEX,84:POKEX+1,72:POKEX+2,82:POKEX+3,69:POKEX+4,69:X=X+5
9560 POKEX,5:POKEX+1,1:X=X+2
9570 POKEX,32:POKEX+1,70:POKEX+2,79:POKEX+3,85:POKEX+4,82:X=X+5
9580 POKEX,5:POKEX+1,1:X=X+2
9590 POKEX,32:POKEX+1,70:POKEX+2,73:POKEX+3,86:POKEX+4,69:X=X+5
9600 POKEX,5:POKEX+1,1:X=X+2
9610 POKEX,32:POKEX+1,83:POKEX+2,73:POKEX+3,88:POKEX+4,32:X=X+5
9620 POKEX,5:POKEX+1,1:X=X+2
9630 POKEX,32:POKEX+1,65:POKEX+2,78:POKEX+3,68:POKEX+4,32:X=X+5
9640 POKEX,5:POKEX+1,1:X=X+2
9650 POKEX,32:POKEX+1,32:POKEX+2,32:POKEX+3,32:POKEX+4,32:X=X+5
9660 RETURN
11000 REM LATIN HUSTLE WOMAN
11010 DATA WLF1,218
11020 DATA WRF1,226
11030 DATA SPACE,989
11040 DATA DELAY1,4
11050 DATA ERASE1,226
11060 DATA WRT1,232
11070 DATA ONE,989
11080 DATA DELAY1,8
11090 DATA ERASE1,232
11100 DATA WRF1,226
11110 DATA TWO,989
11120 DATA DELAY1,12
11130 DATA ERASE1,218
11140 DATA WLF1,602
11150 DATA THREE,989
11160 DATA DELAY2,16
11170 DATA ERASE1,226
11180 DATA WRF1,610
11190 DATA AND,989
11200 DATA DELAY2,20
11210 DATA ERASE1,602
11220 DATA WLF1,410
11230 DATA FOUR,989
11240 DATA DELAY1,24
11250 DATA ERASE1,610
11260 DATA WRF1,226
11270 DATA FIVE,989
11280 DATA DELAY1,28
11290 DATA ERASE1,410
11300 DATA WLF1,218
11310 DATA SIX,989
11320 DATA DELAY1,4
11330 DATA END,0
12000 REM LATIN HUSTLE MAN
12010 DATA MLF1,218
12020 DATA MRF1,226
12030 DATA SPACE,989
12040 DATA DELAY1,4
12050 DATA ERASE1,218
12060 DATA MLT1,212
12070 DATA ONE,989
12080 DATA DELAY1,8
12090 DATA ERASE1,212
12100 DATA MLF1,218
12110 DATA TWO,989
12120 DATA DELAY1,12
```

Program continues

```
12130 DATA ERASE1,226
12140 DATA MRF1,610
12150 DATA THREE,989
12160 DATA DELAY2,16
12170 DATA ERASE1,218
12180 DATA MLF1,602
12190 DATA AND,989
12200 DATA DELAY2,20
12210 DATA ERASE1,610
12220 DATA MRF1,418
12230 DATA FOUR,989
12240 DATA DELAY1,24
12250 DATA ERASE1,602
12260 DATA MLF1,218
12270 DATA FIVE,989
12280 DATA DELAY1,28
12290 DATA ERASE1,418
12300 DATA MRF1,226
12310 DATA SIX,989
12320 DATA DELAY1,4
12330 DATA END,0
13000 REM DISCO HUSTLE WOMAN
13010 DATA WLF1,218
13020 DATA WRF1,226
13030 DATA SPACE,989
13040 DATA DELAY1,4
13050 DATA ERASE1,226
13060 DATA WRF1,610
13070 DATA ONE,989
13080 DATA DELAY1,8
13090 DATA ERASE1,218
13100 DATA ERASE1,218
13110 DATA TWO,989
13120 DATA DELAY2,12
13130 DATA ERASE1,218
13140 DATA WLF1,218
13150 DATA TWO,989
13160 DATA DELAY2,16
13170 DATA ERASE1,610
13180 DATA WRF1,226
13190 DATA THREE,989
13200 DATA DELAY1,20
13210 DATA ERASE1,218
13220 DATA WLF1,602
13230 DATA FOUR,989
13240 DATA DELAY1,24
13250 DATA ERASE1,226
13260 DATA ERASE1,226
13270 DATA FIVE,989
13280 DATA DELAY2,28
13290 DATA ERASE1,226
13300 DATA WRF1,226
13310 DATA FIVE,989
13320 DATA DELAY2,32
13330 DATA ERASE1,602
13335 DATA WLF1,218
13340 DATA SIX,989
13350 DATA DELAY1,4
13360 DATA END,0
14000 REM DISCO HUSTLE MAN
14010 DATA MLF1,218
14020 DATA MRF1,226
14030 DATA SPACE,989
14040 DATA DELAY1,4
14050 DATA ERASE1,218
14060 DATA MLF1,602
14070 DATA ONE,989
14080 DATA DELAY1,8
14090 DATA ERASE1,226
14100 DATA ERASE1,226
14110 DATA TWO,989
14120 DATA DELAY2,12
14130 DATA ERASE1,226
14140 DATA MRF1,226
14150 DATA TWO,989
14160 DATA DELAY2,16
14170 DATA ERASE1,602
14180 DATA MLF1,218
14190 DATA THREE,989
14200 DATA DELAY1,20
14210 DATA ERASE1,226
14220 DATA MRF1,610
14230 DATA FOUR,989
14240 DATA DELAY1,24
14250 DATA ERASE1,218
14260 DATA ERASE1,218
14270 DATA FIVE,989
14280 DATA DELAY2,28
14290 DATA ERASE1,218
14300 DATA MLF1,218
14310 DATA FIVE,989
14320 DATA DELAY2,32
14330 DATA ERASE1,610
14340 DATA MRF1,226
14350 DATA SIX,989
14360 DATA DELAY1,4
14370 DATA END,0
17000 REM ROPE HUSTLE WOMAN
17010 DATA WLF1,602
17020 DATA WRF1,610
17030 DATA SPACE,989
17040 DATA DELAY1,4
17050 DATA ERASE1,610
17060 DATA WRF1,418
17070 DATA ONE,989
17080 DATA DELAY1,8
17090 DATA ERASE1,602
17100 DATA WLF1,410
17110 DATA AND,989
17120 DATA DELAY1,12
17130 DATA ERASE1,418
17140 DATA WRF1,226
17150 DATA TWO,989
17160 DATA DELAY1,16
17170 DATA ERASE1,410
17180 DATA WLF1,218
17190 DATA AND,989
17200 DATA DELAY1,20
17210 DATA ERASE1,226
17220 DATA WRF1,418
17230 DATA THREE,989
17240 DATA DELAY1,24
17250 DATA ERASE1,218
17260 DATA WLF1,410
17270 DATA AND,989
17280 DATA DELAY1,28
17290 DATA ERASE1,418
17300 DATA WRF1,610
17310 DATA FOUR,989
17320 DATA DELAY1,32
17330 DATA ERASE1,410
17340 DATA WLF1,602
17350 DATA AND,989
17360 DATA DELAY1,4
17370 DATA END,0
18000 REM ROPE HUSTLE MAN
18010 DATA MLF1,602
18020 DATA MRF1,610
18030 DATA SPACE,989
18040 DATA DELAY1,4
18050 DATA ERASE1,602
18060 DATA MLF1,410
18070 DATA ONE,989
18080 DATA DELAY1,8
18090 DATA ERASE1,610
18100 DATA MRF1,418
18110 DATA AND,989
18120 DATA DELAY1,12
18130 DATA ERASE1,410
18140 DATA MLF1,218
18150 DATA TWO,989
18160 DATA DELAY1,16
18170 DATA ERASE1,418
18180 DATA MRF1,226
18190 DATA AND,989
18200 DATA DELAY1,20
18210 DATA ERASE1,218
18220 DATA MLF1,410
18230 DATA THREE,989
18240 DATA DELAY1,24
18250 DATA ERASE1,226
18260 DATA MRF1,418
18270 DATA AND,989
18280 DATA DELAY1,28
18290 DATA ERASE1,410
18300 DATA MLF1,602
18310 DATA FOUR,989
18320 DATA DELAY1,32
18330 DATA ERASE1,418
18340 DATA MRF1,610
18350 DATA AND,989
18360 DATA DELAY1,4
18370 DATA END,0
```

Note: Set Memory Size = 31232. Listing is configured for Level II BASIC only.

If you think you're good with a soldering pencil, try this project. It may change your mind.

Hardwire the RS-232

Don DeJarnette
1909 6th Ave. East
Tuscaloosa, AL 35401

It's midnight and your printer is busily printing out the last of those 1000 sorted inventory items. The printer stops. Ah, bed at last. But noooo!! The RS-232 board has screwed up again.

The RS-232 board is secured(?) to the 42-pin connector by two small screws. The spring clips on the connector contact the foil side of the circuit board. In theory, this connector works very well. In practice, the screws can be tightened unevenly, dust can accumulate on the contacts and the board can flex. Board flexing was my greatest problem. After a long session the CRT would heat the board, causing it to flex, resulting in poor contact with the connector.

Radio Shack has attempted to cope with this problem by cleaning the contacts, re-tinning the foil with solder and replacing the connector. This was a temporary solution in my case.

I felt the only remedy left was to hard wire the board to the connector.

Roll up your sleeves, take a shot of bourbon to calm your nerves, and pull out the smallest pencil soldering iron you can find. You will need to purchase enough ribbon cable to make 26 connections. Radio Shack has 18 inches of color-coded 16-strand ribbon cable for $3.99 (276-1976). Only about five or six inches of cable should be needed. Good soldering technique is a necessity.

Step 1—Separate all of the wires up to 1 1/2 inches from the end.

Step 2—Strip 1/16 inch insulation from each wire and lightly tin with solder.

Step 3—Trim each tinned end to 1/32 inch.

Step 4—Very carefully tin the 28 pins (Fig. 1) on the connector making sure there are no solder bridges. Note: Pins 1 and 2 are connected to each other; pins 3 and 4 are connected to each other. This accounts for the 26 wires. Tin the connector pins very quickly; this will prevent the heat from melting the plastic body of the connector.

Step 5—Solder the wires of the ribbon cable to the connector. This is the most difficult part of the process. A very short touch of the iron to the wire as it is positioned on the pin is all that is required to make a good solder connection. Alternate the solder points on the pins to each extreme end. This will help pre-

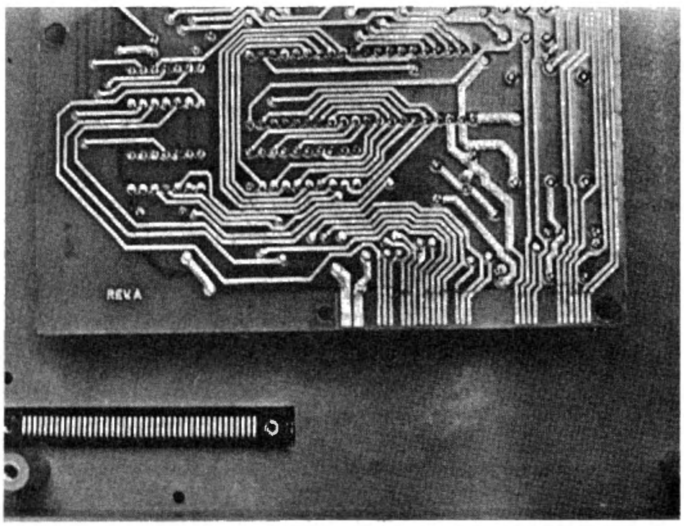


Photo 1. RS-232 Board and 42 pin connector. Note: the circuit board is pictured upside down with the foil contacts in reverse order.

Orient this end to rear of expansion interface (side that 110 volt cords exit from case)

1 ——>	CONNECT	22 ——>	CONNECT
2 ——>	CONNECT	23 ——>	NO CONNECT
3 ——>	CONNECT	24 ——>	NO CONNECT
4 ——>	CONNECT	25 ——>	NO CONNECT
5 ——>	NO CONNECT	26 ——>	NO CONNECT
6 ——>	NO CONNECT	27 ——>	NO CONNECT
7 ——>	CONNECT	28 ——>	NO CONNECT
8 ——>	CONNECT	29 ——>	NO CONNECT
9 ——>	CONNECT	30 ——>	CONNECT
10 ——>	NO CONNECT	31 ——>	CONNECT
11 ——>	CONNECT	32 ——>	CONNECT
12 ——>	CONNECT	33 ——>	CONNECT
13 ——>	CONNECT	34 ——>	NO CONNECT
14 ——>	CONNECT	35 ——>	NO CONNECT
15 ——>	CONNECT	36 ——>	NO CONNECT
16 ——>	CONNECT	37 ——>	CONNECT
17 ——>	CONNECT	38 ——>	CONNECT
18 ——>	CONNECT	39 ——>	CONNECT
19 ——>	NO CONNECT	40 ——>	CONNECT
20 ——>	CONNECT	41 ——>	CONNECT
21 ——>	CONNECT	42 ——>	CONNECT

Orient this side to front of expansion interface (side with on-off switch)

Figure 1. Pin Diagram of RS-232 Board Connector. Remember to connect pin 1 to pin 2 and pin 3 to pin 4 (see photograph).

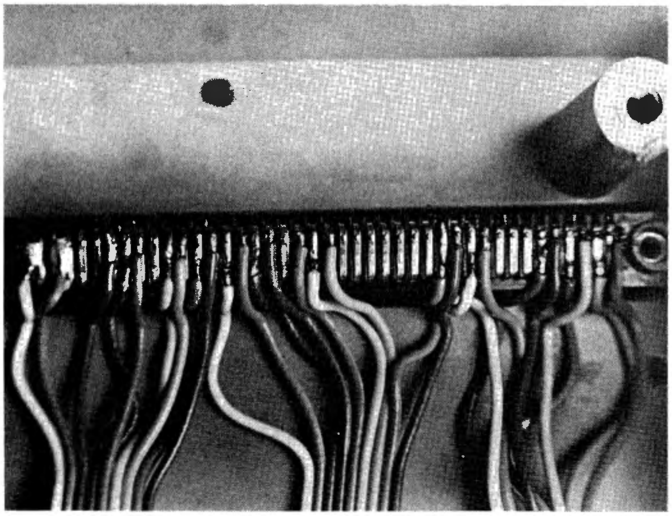


Photo 2. Wired 42 pin connector. Note the alternating solder points.

vent a solder bridge. Note: on pins 1 and 2 lay the wire between the pins and bridge the gap between them, connecting the two. Likewise, bridge the gap on pins 3 and 4.

Step 6—Check for unwanted solder bridges. I find using a magnifying glass very helpful.

Step 7—While pressing down on the top of the connector, very carefully bend and form the ribbon cable around the connector to the base of the RS-232 compartment. Secure the cable to the base with electrical or duct tape. This will prevent the pins from being pulled out of their sockets by the ribbon cable.

Step 8—Follow the color coded wires from the connector to the foil connections on the RS-232 board. Use the two wires from the bridged connections (1-2, 3-4) to key to the wide foil tabs on the board.

Step 9—Solder the cable to the RS-232 board. This is not as difficult as the connector, but watch for solder bridges and alternate solder points.

Step 10—Route the wires conveniently and position the board in the compartment. I used the foam rubber packing from my expansion interface box to hold the board in place. I also used a small amount of silicon caulking compound on top of the connector. This assures that pins will not move.

This process can be somewhat painstaking, but so can sitting through a two hour sort. In the nine months since I have installed the direct wiring of the RS-232 board, I have not had a single malfunction.■

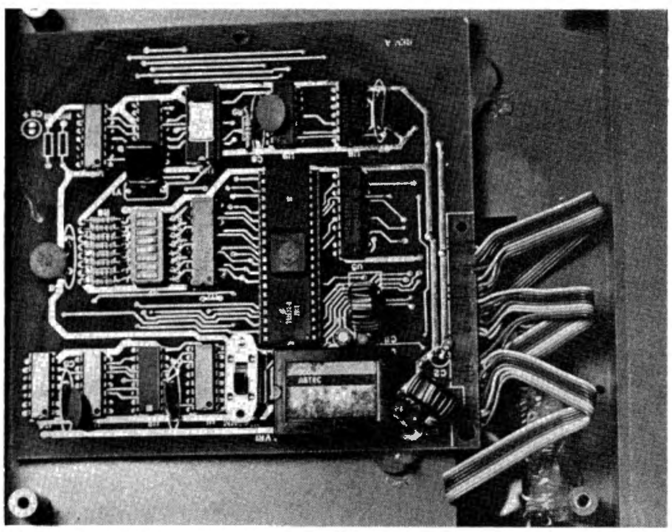


Photo 3. Completed circuit wiring and board placement.

Jazz up your displays with segmented screen scrolls.

The Flexible Scroller

Jeff Myers
210 Park Ave.
Hillsboro, OH 45133

I've often wished for a simple way to make the TRS-80 video display more stylish. If not stylish, at least more unexpected than always moving up one line.

A year ago, I devised a machine language program to treat the video display as three separate entities. Downward scrolls, upward scrolls and quick erasing were all available for each third of the screen—left, center, and right. When some of my high school students were adapting this routine to a game program they were writing, one of them goaded me into improving my program by asking whether I could make the movable regions more flexible.

After six or eight hair-pulling hours I had an answer. Yes!

The program is called Flexible Scroller (Program Listing 1). It is a BASIC program including a subroutine to POKE the machine language program into a string variable (lines 49000+), an operating routine (48000) and two examples of how it can be put to work.

When using Flexible Scroller, you do not need to protect memory (the POKE portion assumes a maximum of 16K; on larger systems minor alterations will be needed). During the use of this program you *will* need to protect the variable O and the string O$, however. You can see that in lines above 48000, seven values are POKEd directly into the machine language program so that it scrolls up or down. The variable Ø is in charge of aiming these POKEs properly.

Early in the program the subroutine at 49000+ must be called to POKE the machine language into the string variable Ø$. To make use of the scroller:

- Set S equal to the starting point (left edge) of the vertical strip to be moved (0 to 63 inclusive);
- Set W equal to the width of the strip (1 to 64 incl.), making sure that W + S does not exceed 64;
- Set the flag F$ to U or D (you get to guess what they're for !);
- GOSUB 48000.

Listing 1 illustrates two simple uses of the Flexible Scroller. One of the options shows a random column-erase which is done by calling 48000 sixteen times. To make it scroll down, just change F$.

The second option allows you to specify various combinations of W, S, and F$ and watch what happens. Be sure to try the TRS-80 feature of repeating inputs; enter one set of data, watch it move, then hit enter repeatedly.

The best way to use Flexible Scroller is to keep a copy of lines 48000 to the end. This can be entered when you begin a program in which you anticipate using it or it can be easily appended to existing programs, assuming the line numbers are safe. My tape of lines 48000+ is a weapon I'm frequently glad to have in my programming arsenal.

Invent applications of your own for the scroller: Erase the screen (in either directions or both!) one column at a time. Code and decode messages by vertically zapping columns on the screen. Leave a column or two of messages stationary while other parts of the screen scroll normally during the course of a game or practical program. (In the last case, PRINT@ with a trailing semicolon is useful.)

Listing 2 is a source code listing of this 86-byte program. The data in the BASIC program are decimal values which create this program in machine code.

Special comments have been added where up-scrolling and down-scrolling differ. This was the challenging part of writing the program: inventing a routine which would do both with a minimum of alterations.

Assembly language programmers might try to make it even more flexible by eliminating the need to alter the program at all. (The width and starting point are sent to the program as the argument of USR and register H gets the width and L gets the start. Since S doesn't exceed 63, bit 7 is available for use as an up/down flag and the machine language could alter itself.)

I have trespassed into the disk command vectors at hexadecimal 4152—4159. I do not use disks and this was one way to make the program relocatable. To use the program with a disk system, you'll have to store the variables somewhere else, CALL 000B (hex) and set up storage, PUSH and POP a lot more, or some other alteration.

Whether you want to alter the inner workings or not, Flexible Scroller should provide you with an interesting programming tool.■

```
0       'FLEXIBLE SCROLLER                          BY JEFF MYERS
1       'CALL 49000 EARLY TO POKE MACH.LANG.ROUTINE INTO
                     STRING O$...*** PRESERVE VARIABLE
    O THROUGHOUT ***
2       'AFTER SETUP, SEND W=WIDTH, S=STARTING COLUMN AN
    D                F$= 0  OR  0  TO SUBROUTINE 48000
    TO PERFORM SCROLL
3       'NOTE:MAKE SURE W + S < 64
100 CLS:CLEAR170:GOSUB49000
110 INPUT"1=COLUMN-ERASE SAMPLE     2=INPUT SAMPLE ";F
120 ON F GOTO 130,170
130 CLS:FORK=1TO100:PRINT@RND(1000),CHR$(64+RND(26));:N
    EXT
140 PRINT@0,"";:INPUT"HIT ENTER TO SEE A RANDOM COLUMN
    ERASED BY 16 'UP' CALLS";W$
150 W=RND(20):S=RND(40):F$="U":FORI=1TO16:GOSUB48000:NE
    XT
160 GOTO140
```

```
170 CLS:FORK=1TO100:PRINT@RND(1000),CHR$(128+RND(63));:
      NEXT
180 PRINT@0,"";:INPUT"WIDTH,START, U OR D";W,S,F$
190 GOSUB48000:GOTO180
48000  'SEND W,S & F$=U OR D
48010 IFF$="U"THEN48030
48020 POKEO+17,128:POKEO+18,63:POKEO+30,90:POKEO+57,66:
      POKEO+63,66:POKEO+68,0:POKEO+69,60:GOTO48060
48030 POKEO+17,64:POKEO+18,60:POKEO+30,82:POKEO+57,74:P
      OKEO+63,74:POKEO+68,192:POKEO+69,63
48060 U=USR(256*W+S):RETURN
49000 O$=STRING$(87,1):O1=VARPTR(O$):O=PEEK(O1+1)+256*P
      EEK(O1+2):POKE16526,PEEK(O1+1):POKE16527,PEEK(O1+2
      )
49010 FORM=O TO O+85:READX:POKEM,X:NEXT:RETURN
50000 DATA 205,127,10,235,33,82,65,115,35,54,0,35,114,3
      5,54,0,33,64,60,237,91,82,65,25,229,17,64,0,175,23
      7,82,235
50001 DATA 225,6,15,197,34,86,65,237,83,88,65,237,75,84
      ,65,237,176,1,64,0,175,42,88,65,237,74,235,42,86,6
      5,237,74
50002 DATA 193,16,224,33,192,63,237,91,82,65,25,237,75,
      84,65,65,54,32,35,16,251,201
```

Program Listing 2.

```
                 00100         ;SOURCE CODE FOR  =FLEXIBLE
  SCROLLER=
                 00110         ;   BY JEFF MYERS        6/1
 /80
                 00120         ;
                 00130         ;THIS LISTING IS FOR DOWNWA
  RD SCROLLING
                 00140         ;**COMMENTS DESCRIBE CHANGE
  S FOR UPWARD
0000 CD7F0A      00150         CALL    0A7FH            ;GE
  T H=WIDTH,L=START
0003 EB          00160         EX      DE,HL
0004 215241      00170         LD      HL,4152H         ;ST
  ORAGE IN DISK VECTORS
0007 73          00180         LD      (HL),E           ;ST
  UFF START AS 2-BYTE
0008 23          00190         INC     HL
0009 3600        00200         LD      (HL),0
000B 23          00210         INC     HL
000C 72          00220         LD      (HL),D           ;ST
  UFF WIDTH "  "
000D 23          00230         INC     HL
000E 3600        00240         LD      (HL),0
0010 21803F      00250         LD      HL,3F80H         ;**
  FOR UP:LD HL,3C40H
0013 ED5B5241 00260            LD      DE,(4152H)       ;GE
  T START COLUMN & ADD
0017 19          00270         ADD     HL,DE            ;
  TO LEFT EDGE ADDR
0018 E5          00280         PUSH    HL
0019 114000      00290         LD      DE,0040H
001C AF          00300         XOR     A
001D ED5A        00310         ADC     HL,DE            ;**
  FOR UP:SBC HL,DE
001F EB          00320         EX      DE,HL
0020 E1          00330         POP     HL
0021 060F        00340         LD      B,0FH            ;15
  LINES TO MOVE
0023 C5          00350 LOOP    PUSH    BC
0024 225641      00360         LD      (4156H),HL       ;SA
  VE HL=SOURCE
0027 ED535841 00370            LD      (4158H),DE       ;SA
  VE DE=DESTINATION
002B ED4B5441 00380            LD      BC,(4154H)
002F EDB0        00390         LDIR
0031 014000      00400         LD      BC,0040H
0034 AF          00410         XOR     A
0035 2A5841      00420         LD      HL,(4158H)
0038 ED42        00430         SBC     HL,BC            ;**
  FOR UP:ADC HL,BC
003A EB          00440         EX      DE,HL
003B 2A5641      00450         LD      HL,(4156H)
003E ED42        00460         SBC     HL,BC            ;**
  FOR UP:ADC HL,BC
0040 C1          00470         POP     BC
0041 10E0        00480         DJNZ    LOOP
0043 21003C      00490         LD      HL,3C00H         ;**
  FOR UP:LD HL,3FC0H
0046 ED5B5241 00500            LD      DE,(4152H)
004A 19          00510         ADD     HL,DE            ;TH
  IS LAST ROUTINE IS
004B ED4B5441 00520            LD      BC,(4154H)       ; A
  CLEAN-UP OF THE
004F 41          00530         LD      B,C              ; T
  OP (BOTTOM) EDGE
0050 3620        00540 CLEAR   LD      (HL),20H
0052 23          00550         INC     HL
0053 10FB        00560         DJNZ    CLEAR
0055 C9          00570         RET
0000             00580         END
00000 TOTAL ERRORS
```


✓See List of Advertisers on page 306

A complete assembly language development package for 16K Level II owners.

Editor/Assembler-Plus

Editor/Assembler-Plus
Microsoft
Bellevue, WA
$29.95

Dan Zuckerman
56 Hickory Place
Livingston, NJ 07039

Attention assembly language programmers! Have you ever wished you could assemble, debug, and reassemble your program *ad infinitum* without touching the cassette tape in the process? Your wish has come true! It's called Editor/Assembler-Plus and it's made by Microsoft, the same people who wrote Radio Shack BASIC and Radio Shack's Editor/Assembler. This 12.5 K bug-free machine language program includes features such as: macros, conditional assembly, and many useful editing commands. Best of all, it only requires a Level II TRS-80 with 16K of memory.

The Old Way

For the benefit of the novice, first let me explain Radio Shack's Editor/Assembler and T-BUG machine language monitor. They are used to write, assemble, and debug assembly language programs. After loading the Editor/Assembler tape, type in your source text (assembly language program). A successful editing and assembling session results in an object (machine language) tape and a source (assembly language text) tape. Next, load T-BUG and your object tape. Then, try running your program. Chances are it doesn't work. Use T-BUG to execute a few instructions at a time by using its single breakpoint (similar to a STOP statement in BASIC program debugging).

Unless you are lucky enough to have a printer (or a photographic memory) you will have to refer back to your notes on your program and a table of instructions to interpret those strange looking hexadecimal numbers that T-BUG flashes at you. Upon finding errors, reload Editor/Assembler and your source tape, correct the mistakes, and record more tape. If your program is more than a few lines long this is a very discouraging process.

As previously mentioned, Editor/Assembler-Plus was written by Microsoft. In fact, one of its authors, Mark Chamberlin, wrote the Radio Shack Editor/Assembler. Consequently, Editor/Assembler-Plus is an improved version of Radio Shack's Editor/Assembler.

A Comparison

Microsoft's Editor/Assembler-Plus package includes a 112 page instruction manual (well indexed), a 14 page folding quick reference card, and a tape with Editor/Assembler-Plus and Z-BUG (combined) on one side and Z-BUG alone on the other. (Z-BUG is a machine language monitor similar to T-BUG, RSM-2, and others for debugging machine language programs.) After successfully loading Editor/Assembler-Plus I recommend that you immediately follow any update instructions and make a new working tape.

The loading instructions are easy to follow even if you've never loaded a machine language tape before, as is the rest of the instruction manual.

The most valuable feature of Editor/Assembler-Plus is the assemble into memory option. Using this feature, the option automatically decides where to assemble your object code so that it does not conflict with your source text, symbol table, or the Editor/Assembler-Plus itself. With all these things in safe places, you can easily switch between Editor/Assembler and Z-BUG to edit, reassemble, or debug your program without manipulating tapes or losing your program.

No matter how badly your program crashes, as long as it does not disturb your source text or the Editor/Assembler-Plus, you can reenter Editor/Assembler-Plus without losing anything or touching a tape. Of course, you can also reserve high memory, assemble into memory of your own choice, or generate object tapes without assembling into memory. Larger programs can be handled by using the QUASH command to delete Z-BUG (and the assembler too, if desired). You can then load Z-BUG to debug your program.

Macro and Conditional Assembly

Macros permit you to generate commonly used sequences of instruction easily. For the benefit of BASIC programmers, a macro is best defined as a multi-line defined function or a portion of program with blanks for variables of your choice. Each time a macro is referenced, a particular sequence of code is assembled using unique parameters for that reference. This saves both time and memory space. Using macros can help you develop your own library of easily used routines, create a new language, or even simulate another computer.

The assembler also has an expanded expression evaluator (that's computerese for calculator). Radio Shack's Editor/Assembler is limited to addition, subtraction, logical and, and logical shift. The new expression evaluator can multiply, integer divide, logical or, logical exclusive or, logical not (ones complement a number), and modulo (remainder of integer division). Octal (base eight), hexadecimal (base 16), and decimal (normal base 10) can be used with Radio Shack's Editor/Assembler but Editor/Assembler-Plus also understands parentheses.

Conditional assembly is a function that allows you to assemble sections of your program only if a specified condition is true. The above feature combined with the *equal to* and *not equal to* operators make conditional assembly a useful feature.

Using conditional assembly sections of code can be assembled or ignored based on an equation or a logical arithmetic expression. These algebraic style parameters are affected by labels set in the beginning of your program. Thus, modifying one label can affect many different sections of conditionally assembled program. A conditional section of source text can contain a conditional section within it. Thus one source text can be used to create many different versions of a machine language program. Programs can be made for disk, tape, screen, printer, Level I ROM, Level II ROM, or different size memories, to mention a few possibilities.

Whenever a program is assembled, a symbol table is created. This tells which symbolic references you have used and which memory locations or data they actually represent. Editor/Assembler-Plus lists these symbols in alphabetical order along with the data they represent. The list is annotated to tell you which symbols represent macros, which change as the program is assembled, and which are involved in possible error conditions.

All error messages are fully typed out and are self-explanatory. There is no need to search through the manual to interpret error messages. You can also direct the assembler to pause when it detects an error, and await your instructions. You can stop and correct the error, continue until the next error, or abandon the wait on error feature and let assembly continue without pausing for errors.

Easy Editing

It's a pleasure to edit source text with Editor/Assembler-Plus. Text can be inserted, deleted, listed, and manipulated on tape. Lines can be copied or moved. The character oriented editor (like the Level II BASIC editor) can be invoked. The X (eXtend) subcommand of the editor can be automatically entered with one command. You can search for particular words (strings). You can even automatically replace all occurrences of a given string with another. Not only can the above functions be performed on single lines but they can all be performed on groups of lines. Thus, you can copy or move a group of lines, edit a section of text all at once, extend an area of text (to add comments), or find and/or substitute for strings within any particular area of text. The entire text can be renumbered at any time. There is never any reason to type the same thing twice or scroll through the entire text searching for a particular line.

With Radio Shack's Editor/Assembler you can specify a line number or the beginning and ending line numbers of a series. Editor/Assembler-Plus lets you specify lines such as the 13th line after line 10 (10+13) or the fifth line before the last (* −5). You can specify line number ranges such as the 23 lines following line 20 (20!23) or even the three lines after the fifth line following the first line in the text (#+5!3). It is an extremely flexible system!

If you have a printer you'll be pleased to know that Editor/Assembler-Plus has all the Radio Shack Editor/Assembler commands to send information to your printer. You can send output from the assembly or your source text with or without line numbers. If your printer requires software in RAM to run, high memory can be reserved for this and any other use you may desire.

Z-BUG

Z-BUG is a completely new machine language monitor. The feature I like best is its use of the symbol table to reference memory locations and data. For example, to run your program you can tell Z-BUG to go to the location labeled in your source text as START. Unless otherwise commanded it will refer to all memory locations in terms of the labels used in your text. The location following START is referred to as START+01. This feature can be defeated and locations displayed in octal, hexadecimal or decimal form. The contents of memory can be displayed in disassembled form (Z-80 mnemonics). All addressing (including relative addressing) can be displayed using either labels or numeric memory references. ences.

Program listings can easily be followed because instructions affecting the program flow can be used to automatically determine the next memory location shown. Memory can be displayed in bytes (8 bits) and words (2 bytes) in octal, hexadecimal, decimal, or even the printable ASCII characters. Locations can be modified in any integer base between two (binary) and 16 (hexadecimal) inclusive. One command allows an entire series of memory locations to be displayed together in any of the above forms and you may scroll up and down through memory.

The full expression evaluator of the assembler (as previously described) is available for immediate use in calculator mode. No longer is it necessary to have a hand calculator handy to debug programs. Symbolic references as well as numbers in bases two through 16 may all be used in calculations. By using symbols, a thoroughly labeled program requires very little, if any, numerical manipulation for debugging.

Z-BUG can maintain eight breakpoints at once as opposed to T-BUG's one. Breakpoints are referred to by location and assigned a number (zero through seven). After stopping at a breakpoint you can continue until the next breakpoint or any specified number of following breakpoints. Thus, by setting a breakpoint in a loop you can observe the results of going through the loop any number of times. A built-in single step function allows you to execute a program one instruction at a time while modifying the instructions and their effects on memory, screen, registers, and flags.

All of the commands are logically designed and easy to learn and use. The simple format of the commands is facilitated by a new *escape* character. When the shift key and the up arrow are both depressed a $ symbol results. This $ is different in operation than the shifted four symbol. This new *escape* character is used in many of the commands.

Editor/Assembler-Plus is made foolproof by requiring a two step command sequence for all cassette operations and for the destructive quash command. Any cassette operation command results in the CASSETTE READY prompt, after which you could hit Break to abort the command or hit any other key to confirm it. QUASH (which deletes parts of the system to allow more programming space) results in a QUASH? prompt and requires a similar response. Furthermore, the program is always checking your memory. Attempting to modify a location in fixed memory (ROM) or a defective location in user memory (RAM) results in a BAD MEMORY error message.

Almost Perfect

I can find only one fault in this giant program which fills more than three-fourths of available memory in a 16 K system. There is a short page break every 2.5 seconds when a machine language tape is generated. I find that this space in the tape makes programs slightly longer and more difficult to load. I prefer the continuous tape output of T-BUG. The monitor lacks such luxurious features as tape verify (similar to CLOAD? in BASIC), memory search and compare functions. However, Editor/Assembler-Plus' many other features make up for these functions which would utilize more memory and leave less space for programming.

One feature I felt was lacking was a symbolic listing function which is used to disassemble any supplied machine language tape or an area of memory into the assembler text buffer. By using the superb editing commands, you could easily find all labels needed for relocation and assemble or modify the new text. Disassemblers to accomplish this are available separately (Myosis Disassembler 1.2, for example). If you are willing to do some typing, this feature could be accomplished using the system as provided. You just have to display a screen full of symbolic listing in Z-BUG, jump to the Editor/Assembler, and type the information on the screen into the text buffer (same as typing in a new program). This is proof of the feasibility of implementing this feature.

Editor/Assembler-Plus is all a beginning assembly language programmer (with a 16 K Level II TRS-80) needs to have a complete assembly language development system. This powerful, easy to use system is sold for about $30—that's $15 less than buying Radio Shack Editor/Assembler and T-BUG separately. A good introductory programming book with information on Z-80 instructions and mnemonics is needed to complement the documentation provided.

Programmers without an assembler and Radio Shack Editor/Assembler owners tired of waiting for tapes load and desiring the new features of Editor/Assembler-Plus will find it well worth the money. Radio Shack Editor/Assembler owners might even consider their initial investment well worth it for the excellent documentation that is provided. ■

An observing aid for students of the species.

A Field Guide To Computerists

Chuck Doherty
32 Meadowood Drive
South Dartmouth, MA 02748

This is quite a country we live in. Where else could one find such a wide range of different people? And the computer boom of the last few years has spawned a totally new group from all the others. This group has managed, in a few short years, to carve a permanent nitch for themselves in society. This group is, of course, the computerist.

I'm not referring to the computer professional, who spends his eight hours at the helm of some IBM monster, and at five flips the switch to become Joe American, Regular Guy. I'm talking about the man who rushes home from work, pats his wife, kisses the dog and disappears until the wee hours into that mysterious little room with the phosphorus glow.

This is not to imply that all computerists are alike, quite the opposite is true. The purpose of this article is to guide you in learning the many sub-groups, and to point out the differences between them.

The Whiz Kid
(Youtharious Technicalia)

The easiest to spot of all the sub-groups, is the whiz kid who can be identified at some distance. Plaid pants and a different plaid shirt is usually the tip-off along with a plastic shirt-pocket protector containing an inordinant amount of pencils. Another sign is a leather calculator pouch hanging from the belt. If you see any of the above combined, you can be sure you've got a classic.

"The easiest to spot of all the sub-groups, is the whiz kid who can be identified at some distance."

By their thirteenth birthday, most whiz kids know more about computer hardware than most senior engineers.

Don't try to find the whiz kind at sporting events or at discos; try the local computer store or library instead.

The Computer Pioneer
(Forefrontus Electro)

If a house in your neighborhood has solar panels or a windmill, it may also have one of these. The Computer Pioneer looks like the guy in *Popular Science's* "Wordless Workshop" section. He knows that in a few years computers will control robots delivering mail at 300 m.p.h. and he's going to be ready. He also insists that his wife use a computer to figure how many pounds of hamburger to use for four people when the recipe is for only two.

All in all he's a terrific guy to have for a neighbor, especially if you ever need to borrow a benzene powered, home built, metal detector.

The Computer User
(Ineptus Bythebookia)

The computer user is a firm believer in the old addage: "If man were meant to have lowercase, it would be inside." His favorite activity is browsing the corner Radio Shack outlet and he laughs at the idea of writing his own software. This is the guy who, as a child, searched all over town for a certain brand of glue (made by the same people who made his model car kit) because the instructions said so.

He also believes that computers are the wave of the future, but doesn't hope to be their master; he only hopes to be able to talk his way into a job.

The Junior Businessman
(Financia Multipulls)

Every town has a few of these. This is the fellow who runs the shop that sharpens saws, makes 3-D magnetic signs, rubber stamps and used to sell CB radios. The Jr. Businessman has taken almost every correspondence course known to man, and views the computer as the best business opportunity to come down the pike since locking gas caps. One wall of his shop is devoted to pre-packaged software and back issues of computer magazines. He knows that riches and prosperity are right around the corner, just as soon as he gets around to cleaning up that back room and fixing the air conditioner. (He knows how to do it himself; he took a course.)

The Visionary
(Misinformia Wellintentia)

The visionary heard from a friend that computers can fix dinner, scare away burglars, keep all his records and teach the kids fluent German just by pushing a few buttons... nothing to it. Boy, is he in for a surprise.

The Computer Student (Idiot Peripheralis)

The computer student is going to make it big in the world of tomorrow. He is able, in one breath, to recite entire instruction sets of over two dozen different microprocessors. However, he cannot tie his shoes without an illustration.

The Big-Time Hustler (Richis Swiftis)

We've all heard stories of these fellows, but few have ever been met face-to-face. He knows there is money changing hands out there somewhere, and is determined to get his grimy paws on some of it. These are the fellows you see with hats over their faces as reporters try to get an interview. Most are well versed in retirement land sales and commodity options.

The Social Outcast (Dullis, Dullis, Dullis)

By far, The Social Outcast is the least interesting person you will ever meet. However, at the keyboard, he is in charge. He is a brilliant programmer, and understands both software and hardware throughly. But, outside of computer-related discussions he's at a total loss. The Wife and kids are equally as dull. Pass out the No-Doze if this man has a comment at your computer club meeting.

Of course, not every computer user falls into one of these classifications. There are also many others, some so diverse that they only have one member. Also do not forget the indirectly related groups, such as the computer widow, the computer orphan, and the infamous computer salesman.

In the future perhaps a modern Audubon will do a more in-depth study of this group, giving us a standard reference guide to the American Computerist. ■

A powerful tool with minor problems.

Microsoft's Compiler For The Model II

Basic Compiler (Model II)
Microsoft
Bellevue, WA
$400

James L. Waggoher
1854 Wedgemere Road
El Cajon, CA 92020

As a programmer who began his career on IBM 7094s that occupied entire buildings, you'd think I'd be realistic in my expectations of new products, especially compilers!

Reasons for Buying a Compiler

First, let me give you my reasons for purchasing a compiler for the Model II. Radio Shack designed the Model II so the user of a 64K machine would be able to use all 64K for programs. With the disk-based BASIC interpreter occupying 28K to 29K, the programmer is left with 35K to 36K for his programs.

My first reason, then, for buying a compiler is that I could make more memory available for BASIC programs.

The compiler saves memory by compiling statements and producing object code as opposed to ASCII bytes, and by including only those subroutines (e.g., PRINT and LPRINT) that are actually used by the programs. For example, a 9000 byte BASIC program may produce a 15,000 byte object file (/CMD file). But you save with the compiled version at 15,000 bytes because you don't need the 28,000 byte interpreter or 9000 bytes of program, making your net savings 22,000 bytes.

My second reason for buying a compiler is that programs will execute faster.

An interpreter examines each program statement and determines what to do with it when it is encountered. A compiler determines what operations are necessary to execute a statement and produces machine language instructions that are then executed when the program is run. This obviously avoids the extra steps introduced by an interpreter each time a statement is executed.

In general, the increased speed would not be noticed in print operations, screen displays, or interactive sessions, since these activities are limited by the speed of the devices (I/O bound) rather than the speed of the microprocessor. I hoped, however, that the compiler would speed up arithmetic calculations, string searches, and other internal operations.

I invested some money in phone calls to Microsoft to assure myself that the compiler would run on a single-drive Model II.

No Clicks, Hums or Displays

In typical "old hand" programmer fashion, when I received the compiler, I gave the documentation a cursory glance and popped the eight inch diskette into my single drive and hit RESET. Much to my surprise, I was not greeted by the usual clicks, hums, and displays. Instead, I received a message saying there was no system on the disk.

Not being one to give up that easily (programming requires tenacity), I booted a good disk, removed it, and inserted the disk with the compiler on it. I requested a directory that produced a list of files that agreed with what the documentation suggested should be there. This small triumph encouraged me to spend the next several hours between reading the documentation and fruitless attempts to either copy the system onto the compiler disk or copy the compiler files onto a system disk.

Even a programmer's tenacity has limits. I finally decided to get some sleep, and as I drifted off I began composing the conversation that would take place during my angry phone call to Microsoft the next day.

The conversation the following morning went something like this:

"My compiler doesn't work," I said to the customer representative. (Programmers tend to overstate their cases.)

"Did you copy the files to system disk?" he responded.

"I tried. How do I do that with one drive?"

"You can't. You have to take the disk we sent you to a Radio Shack store or a friend with two drives and make the transfer," he replied.

"You said I could run the compiler on a single-drive system."

"You can, but first you need two drives to transfer the files to a system disk."

"What if I lived in Cutbank, Montana, four hundred miles from the nearest Radio Shack computer store?" (Programmers are clever at thinking up hard questions.)

"Believe me, I get calls from those people too," he retorted.

Realizing that the conversation was leading nowhere I asked, "Are there any other surprises in store for me?"

"No," he answered.

End of conversation.

Fortunately the distance from my office to the nearest computer center is only a few miles. I sure feel sorry for that guy in Cutbank!

That evening I figured I was all set to run my BASIC programs through the compiler and produce programs that ran like greased lightning. (Programmers are also eternally optimistic.)

Documentation

Let me explain the organization of the documentation supplied by Microsoft. It's divided into several sections. The first and largest section deals with the commands available in the Microsoft version of the BASIC language. As a part of the description available with each command, a line is included stating the version for which the command is valid. This line may state that a particular command (instruction) is valid for 8K BASIC, Extended BASIC, or Disk BASIC. (Notice I did not mention the BASIC Compiler.)

This large section contains several appendices which explain differences among the various usages. There is also an appendix which describes the peculiarities of the version for the compiler. This section is followed by a section called the "BASIC Compiler User's Manual" for the CP/M usage, a section on the macro assembler and one for the linking loader.

The information dealing with the Model II version of the compiler is contained in six pages appended to the front of the manual. The appended information deals mainly with the commands (instructions) which are *not* legal in this compiler version. The user is left with the assumption that, if it isn't strictly forbidden, it must be okay.

After discovering in the first six pages that dashes "—" must be substituted for slashes "/", :TT (for CRT display), not TTY: and :LP for line printer, I typed the command line for the compile:

```
BASCOM TEST,:TT TEST=TEST-N
```

The disk drive commenced to spin furiously. Lines appeared on the CRT indicating that Microsoft's latest and greatest was up and running. I sat back triumphantly, expecting a blur of lines to scroll across the screen. After several seconds, two more lines appeared which indicated completion of the compile. Strange, I thought, that an 8000 byte program would compile without errors and produce so few lines of output on the CRT. Perhaps there's an error in the command line?

Since the completion of the compile had left me at the TRSDOS READY mode, I requested a list of the directory. Sure enough, there was a file called TEST/REL. It looked very short, so I examined it with the TRSDOS command to list it to the CRT. It looked short because it was short. In fact, it was almost non-existent.

After several more attempts using various permutations of the command line (e.g., list output to screen, list output to printer, list source only, list source and object, don't create object file, etc.), I decided I needed a beer. While sipping my beer, I glared at the CRT and vowed it would not get the better of me this night. (Beer aids tenacity.) Suddenly a thought. Perhaps the compiler doesn't accept the compressed format for BASIC. I quickly loaded the BASIC interpreter, read my program from the disk, and saved it in ASCII using the ",A" extension. At this point, I must confess that in an appendix it does state that the compiler will accept ASCII files.

000 Fatal Errors

I returned to TRSDOS and entered the BASIC compiler command with which I was now gaining a certain intimacy. The same furious disk activity began, and then the CRT was a blur of errors scrolling across the screen.

Success! Well, compared to the results so far it was a success.

After examining and eliminating several errors (bear in mind that this program ran without problems in the interpreter mode), I ran across an innocent looking error labelled "LL." Intrigued, I reached for the Microsoft manual and thumbed to the list of compiler error codes—"LL" meant "line too long."

Examining the line displayed on the CRT to which the error code apparently referred, I counted the characters. The line was only 28 bytes long (back to the documentation). After intensive investigation (and another beer), I found a reference which said, in effect, that the compiler will not accept a physical (as opposed to logical) line greater than 127 bytes.

I carefully recounted the characters displayed on the line. There were still only 28. I examined the rest of the listing. My eyes wandered down to the next line. Now this line, I thought, is definitely longer than 127 bytes. I checked several other lines with line length errors indicated and discovered a pattern. It wasn't the indicated lines that were too long, but the lines following.

Sensing victory was at hand, I quickly severed the lines at a convenient point (less than 127 bytes in length), re-typed the severed portion on a separate line and recompiled the program. After several anxious seconds, I was rewarded with that most elusive of remarks "0000 FATAL ERRORS."

The Purpose of the Loader

Flushed with victory, I plunged on to the next step in creating a loadable object program, The Linking Loader. According to the Addenda to the BASIC Compiler User's Manual, I was to type "L80 TEST-G" to load and execute my program or "L80 TEST-E" to exit and save a copy. If I wished to relocate the program at a base address other than 3000 hexadecimal, I could use the "-P";address or "-D":address switch. Since Microsoft didn't think I needed to know the difference between the two or give any examples to follow, I decided not to get that fancy on my first attempt.

I sat back and reflected upon the purpose of the loader. The linking loader processes the compiled program and searches for references to subroutines (subroutine calls). When a subroutine reference is found, the loader searches a file of subroutines, the subroutine library, and extracts the required routine for inclusion in the object file. The Microsoft subroutine library is approximately 45,000 bytes long. You could expect a program that used every conceivable variation of every command to require the inclusion of one copy of every subroutine in the library.

Conversely, limiting the variety of commands in a program would be a way of reducing the object file size. For example, if a simple program contained several instructions requiring addition and only one or two requiring multiplication and further, assumes the multiplication subroutine is 3000 bytes long, you could save 3000 bytes (more or less) by writing the multiplications as repetitive additions. (An interesting project would be to determine the size and speed of the various subroutines and select the shorter and faster routines when writing your programs.)

Well, back to the loader. The disk was working furiously and very little other activity was apparent. Searching and selecting the various subroutines from the library took several minutes. When the loader has completed its task, four hexadecimal values are displayed on the screen (two of the values are the same). As an example, say the values were 3000, 3A61, and 6AB1 (the 3000 appearing twice). These values are required for the TRSDOS DUMP routine, which saves the object file in a command file. The 3000 represents the beginning of the program.(All compiled programs will start at 3000 unless the "-P":or"-D":options are used.) The 6AB1 represents the location

FUNCTION PERFORMED	TIME (IN SECONDS) FOR 2000 EXECUTIONS COMPILED VERSION	INTERPRETER VERSION
B% = B% *(10/10)	1	15
C = C* *(10/10)	1	14
LOG(I/7)*	34	41
LOG(4.155/7.2)	31	74
SQR(I)*	30	63
I 1/3 (CUBE ROOT)*	58	129
$\frac{-B+SQR(B-4\cdot A\cdot C)}{2\cdot A}$	100	182

*I = 1 to 2000

Table 1.

where execution of the program is to begin.

I entered the TRSDOS DUMP command:

```
DUMP MYPROG/CMD
START=3000,END=6AB1,TRA=3A61
```

which dumped the object file to the disk in a few seconds.

At last, I was ready to perform the ultimate test. At the TRSDOS READY level, I typed in "MYPROG/CMD". The disk activity light came on and the first menu in my program appeared on the screen. After checking several of the features of the program, I decided that I had indeed created a compiled version of my program that actually worked just like the interpreter version.

Since I finally had the compiler figured out I decided to perform a few experiments to determine what kind of increase in processing speed I could expect. I wrote a small program in BASIC and debugged it. Once I had it running in the interpreter mode, I made the changes required for the compiler (such as deleting the CLEAR statement) and saved an ASCII version to compile. I then ran both versions of the program and, using the internal clock in the Model II, printed the time required to perform the various functions. As I expected, there was little difference in the screen display time (although the compiled version appeared slightly faster) or the printout. A significant difference in the string search times was indicated.

The string search test consisted of establishing a 2000 item array with a five-character pattern in each item. A loop was established to search the array using the INSTR$ function. I set the last item in the array to the pattern I was searching for. The interpreter version of the program took 16 seconds to find the required item. The compiler version took just four seconds, a four to one advantage in processing speed for the compiler. Upon examining the programs, I noticed I had used a single precision variable for the loop counter. I changed the variable to an integer type variable in both versions of the program, recompiled, and re-ran the test.

The interpreter version run time was reduced by a second or two (the granularity on the clock is one second), but, the compiled version now ran in only two seconds. This would indicate that programs requiring searches of large numbers of string items would run seven or eight times faster using compiled versions.

Table 1 contains data for various simple tests I ran to the execution speed between interpreter and compiler versions. Each test consisted of executing the indicated function 2000 times and displaying the start and stop times. The execution times indicated are exclusive of any interactive input or display times. The granularity of the clock was one second, and the times indicated include the loop processing time as well as the calculations. I made no attempt to optimize the routines.

Updates

Recently, I received the latest version of the Compiler and a revised addendum. Microsoft's latest version, 5.2, adds the include % function that allows a programmer to create subroutines and add them to a program without re-entering the code (a very handy function). This version also eliminates the step of dumping the object code to a command file during the load process. A new compile switch has also been added to allow compiling using one set of BASIC conventions, and execute using another. Considering all the improvements the latest version is a significant step forward.

After making another visit to the local Radio Shack computer center to transfer the latest version to a system disk, I eagerly anticipated exercising the compiler. I transferred my 29,000 byte BASIC program to the system disk.

I was expecting to encounter similar problems with the line lengths, etc., but figured I would try the largest program first just to see how it fared. To my amazement (apologies to Microsoft for my doubt), the program zipped through the compile in less than four minutes. No line length errors; the new version accepts lines greater than 127 bytes. The four indicated "fatal errors" consisted of reserved words which were imbedded in variable names, a no-no when using "-4" compiler switch.

Still dazed from my initial success, I corrected the indicated errors, re-ran the compiler and decided to attempt the loader. Not wishing to push my luck too far, I did not attempt to use the added feature which eliminates the necessity for dumping the object code. (I later verified that this function does work.)

I typed in the loader command and sat back to wait. In less than four minutes the now familiar start, end and transfer addresses appeared, indicating a successful load. I dumped the object code to a command file and not yet believing my good fortune, typed in the command file name. The first menu of the program appeared on the screen. After exercising several of the features of my program, I had to admit that everything appeared to be working. The 29,000 byte source file was now a 7,000 byte object file, a savings of 22,000 bytes.

A Different Kind of Animal

To summarize my experience with the Microsoft BASIC Compiler for the Model II, I would have to say the greatest deficiency is the documentation. Microsoft has fallen into the trap of trying to make one documentation package serve for every conceivable application.

A compiler is a fundamentally different animal than an interpreter and deserves separate instructions. Five or six pages of addendums cannot begin to provide the detail required to effectively utilize this powerful tool.■

✓See List of Advertisers on page 306

LABYRINTH!

Labyrinth is not easy. Waking a sleeping watch-dragon is the least of your troubles. Somewhere in this nightmare of tunnels lurks an evil minotaur. To escape, you must kill it. And survive.

Only the legendary light sabre of Dnev can kill the minotaur. It is secreted away in stygian darkness. To get it, you must deal with cave bears, gnomes, pits, wraiths, and much more. You must avoid the minotaur at all costs.

Alas! Once armed with your light sabre, the minotaur flees! What treasures must be found and used as bait? Why is the maiden so beautiful? What lies beyond the fog . . . .

3-D PERSPECTIVE GRAPHICS

Labyrinth features the full screen 3-D perspective graphic displays that have made Asylum and Deathmaze 5000 best sellers. You can actually see what you are doing and where you are going! The mazes and buildings are bit-coded. This allows us to store **gigantic** mazes in small amounts of memory. These programs are **not** just a series of stored pictures. Our mazes typically contain **over 600 locations.** Further, machine-language programming gives instantaneous graphics generation and game response!

LABYRINTH places you in a huge maze of tunnels inhabited by gnomes, ghosts, witches, and an evil minotaur. You must find the weapons and treasures needed to destroy the minotaur before he destroys you! There is food enough in the maze to hold out for months!
TRS-80 Level II 16K or Model III 16K $12.95

ASYLUM places you on a cot in a small room. Periodically, a janitor lobs a hand-grenade through the window of your locked door. What you do next could mean survival and escape! It could also mean permanent residence in the home for Deathmaze survivors! To leave, you will have to deal with guards, fellow survivors, doctors, the infamous Crazed Carpenter, and much, much more. Don't expect to get out any time soon! There are over 1200 locations!
TRS-80 Level II 16K or Model III 16K $14.95

DEATHMAZE 5000 places you in a gigantic five-story building. There is only one goal. **ESCAPE ALIVE!** Monsters, dogs, vampires, and other vile horrors will plague your every step as you struggle to survive one of the most challenging adventures ever written. As of December 20, only two people outside the Med Systems' staff were known to have escaped!
TRS-80 Level II 16K or Model III 16K $12.95

✓128

SATISFACTION GUARANTEED!

Asylum, Deathmaze 5000, Labyrinth, and GRBASIC are guaranteed to be the most incredible graphic packages you can buy. If for any reason you are not satisfied with these products, return your order within 14 days for a prompt and cheerful refund.

ORDERING INFORMATION

Orders are processed within five working days. We pay all postage and handling within the U.S., Canada, and U.S. territories. European orders please include $3.00 for air post.

☐ Asylum	($14.95)	$ ______
☐ Deathmaze 5000	($12.95)	$ ______
☐ Labyrinth	($12.95)	$ ______
☐ Adventures on Disk	(add $4.00)	$ ______
☐ GRBASIC Cassette	($19.95)	$ ______
☐ GRBASIC TRS-DOS Disk	($24.95)	$ ______
☐ GRBASIC NEWDOS 80 Disk	($24.95)	$ ______
	TOTAL	$ ______

Name ______

Street ______

City ______ State ______ Zip ______

Computer:
☐ TRS-80 LII 16K ☐ Model III 16K

☐ Mastercard ☐ VISA ☐ check

MC or VISA # ______

Expiration Date ______

Med Systems Software
P.O. Box 2674-Y Chapel Hill, NC 27514
(919) 933-1990

Use this subroutine to generate formatted input commands in your business software.

Dollars and Sense

Tom Andrews
WLKI Radio
Angola IN 46703

If you write much business software in BASIC, you undoubtedly appreciate such conveniences as the print using command in Microsoft BASIC.

But how about a formatted INPUT command? What a boon that would be for inputting dollars and cents amounts!

The subroutine here will provide a solution for TRS-80 users, and many other Microsoft BASIC programmers too. The subroutine can be renumbered and relocated as you like, but wherever you put it, it will provide nice, neat dollars and cents input, with the numbers marching from right to left, and with the decimal point and commas all properly placed.

To use the subroutine, set the value of variable FL to the maximum number of digits to be input. If you do not want the screen to increment to the next line after you hit the carriage return, set FF = 10. Then GOSUB to the routine. When the operator hits return after entering the last digit to be input, the subroutine will return with the input value carried in string variable IN$ which can then be transferred to whatever numeric variable you wish, using the VAL(IN$) command.

For example, if your numeric variable is X then you would write X = VAL(IN$). In this way, both single and double precision input can be handled by the same subroutine.

How It Works

The actual subroutine begins at line 50000 in Listing 1. For a quick demonstration of how it works, simply tack on the short program stub given in Listing 1. This will give you an accumulating calculator on the computer screen.

Note that the subroutine will accept entry of the minus sign only if it is the first thing entered. Attempts to enter the minus sign at any other time will be ignored, as will any effort to make a non-numeric entry. The only exception to this rule is the letter C, which will clear the routine and return the display to zero so you can start over.

In actual use, the subroutine works this way. Let's say that at line 1000 you wish to enter a dollar amount into variable A. Normally, you might write something like

```
1000 INPUT"ENTER A DOLLAR AMOUNT";A
```

and enter the desired dollar amount, including the decimal point. However, if you are entering amounts in columns, and your entries have varying numbers of digits, you will find that the columns don't line up neatly. Our subroutine solves the problem this way:

```
1000 PRINT"ENTER A DOLLAR AMOUNT";:
FL=6:GOSUB 50000:A=VAL(IN$)
```

Your display would look this way if you entered 123456:

Program Listing 1

```
50000 '------------>> INPUT USING ROUTINE <<-----------
50005 FL=FL+2
50010 T%=((FL-6)/3):PRINTSTRING$(T%+2," ");
50015 T%=0
50020 A$="":IN$="":IL$="":IS$=" "
50025 PRINTSTRING$(T%+FL-5," ");
50030 PRINT"$0.00";
50035 C!=0:T=0:I=0
50040 REM
50045 A$=INKEY$:IF A$="" THEN 50045
50050 IF A$<>"C" THEN 50065
50055 GOSUB 50150 '--ADJ FOR COMMAS
50060 PRINTSTRING$(FL+1+T%,24);STRING$(FL+1+T%," ");STR
     ING$(FL+T%,24);:GOTO50020
50065 IF ASC(A$)<>13 THEN 50085
50070 IF C!>2 THEN IN$=LEFT$(IN$,LEN(IN$)-2)+"."+RIGHT$
     (IN$,2) ELSE IF C!=2 THEN IN$="."+IN$ ELSE IF C!=1
      THEN IN$=".0"+IN$ ELSE IN$="0"
50075 IF IS$="-" THEN IN$=IS$+IN$
50080 IF FF<>10 THEN PRINT
50082 FF=0:RETURN
50085 IF C!>=FL-2 THEN 50045
50086 IF C!<>0 THEN 50095
50090 IF A$="-" THEN IS$="-":PRINT STRING$(6,24);IS$;"$
     0.00";:GOTO50040
50093 IF A$="0" THEN A$="":GOTO50040
50095 IF A$<"0" OR A$>"9" THEN 50045
50100 IN$=IN$+A$
50105 GOSUB 50150
```

Program continues

★★★

4MHZ, DOUBLE DENSITY, COLOR & B/W GRAPHICS . . THE LNW80 COMPUTER

COMPARE THE FEATURES AND PERFORMANCE

FEATURES	LNW80	PMC-80**	TRS-80* MODEL III
PROCESSOR	4.0 MHZ	1.8 MHZ	2.0 MHZ
LEVEL II BASIC INTERP.	YES	YES	LEVEL III BASIC
TRS80 MODEL 1 LEVEL II COMPATIBLE	YES	YES	NO
48K BYTES RAM	YES	YES	YES
CASSETTE BAUD RATE	500/1000	500	500/1500
FLOPPY DISK CONTROLLER	SINGLE/ DOUBLE	SINGLE	SINGLE/ DOUBLE
SERIAL RS232 PORT	YES	YES	YES
PRINTER PORT	YES	YES	YES
REAL TIME CLOCK	YES	YES	YES
24 X 80 CHARACTERS	YES	NO	NO
VIDEO MONITOR	YES	YES	YES
UPPER AND LOWER CASE	YES	OPTIONAL	YES
REVERSE VIDEO	YES	NO	NO
KEYBOARD	63 KEY	53 KEY	53 KEY
NUMERIC KEY PAD	YES	NO	YES
B/W GRAPHICS, 128 X 48	YES	YES	YES
HI-RESOLUTION B/W GRAPHICS, 480 X 192	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (NTSC), 128 X 192 IN 8 COLORS	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (RGB), 384 X 192 IN 8 COLORS	OPTIONAL	NO	NO
WARRANTY	6 MONTHS	90 DAYS	90 DAYS
TOTAL SYSTEM PRICE	$1,664.00	$1,840.00	$2,187.00
LESS MONITOR AND DISK DRIVE	$1,200.00	$1,375.00	---

When you've compared the features of an LNW80 Computer, you'll quickly understand why the LNW80 is the ultimate TRS80 software compatible system. LNW RESEARCH offers the most complete microcomputer system at an outstanding low price.
We back up our product with an unconventional 6 month warranty and a 10 days full refund policy, less shipping charges.

LNW80 Computer $1,200.00
LNW80 Computer w/B&W Monitor & one 5" Drive $1,664.00
All orders must be prepaid, CA residents please include 6% sales tax.
Shipping and handling charge of $15.00 must be included with every order.
* TRS80 Product of Tandy Corporation.
** PMC Product of Personal Microcomputer, Inc.

LNW80

- BARE PRINTED CIRCUIT BOARD & MANUAL $89.95

The LNW80 - A high-speed color computer totally compatible with the TRS-80*. The LNW80 gives you the edge in satisfying your computation needs in business, scientific and personal computation. With performance of 4 MHz, Z80A CPU, you'll achieve performance of over twice the processing speed of a TRS-80*. This means you'll get the performance that is comparable to the most expensive microcomputer with the compatibility to the world's most popular computer (TRS-80*) resulting in the widest software base.

FEATURES:
- TRS-80 Model 1 Level II Software Compatible
- High Resolution Graphics
  - RGB Output - 384 x 192 in 8 Colors
  - NTSC Video or RF MOD - 128 x 192 in 8 Colors
  - Black and White - 480 x 192
- 4 MHz CPU
- 500/1000 Baud Cassette
- Upper and Lower Case
- 16K Bytes RAM, 12K Bytes ROM
- Solder Masked and Silkscreened

LNW SYSTEM EXPANSION

- BARE PRINTED CIRCUIT BOARD AND MANUAL $69.95
  WITH GOLD CONNECTORS $84.95

The System Expansion will allow you to expand your LNW80, TRS-80*, or PMC-80** to a complete computer system that is still totally software compatible with the TRS-80* Model 1 Level II.

FEATURES:
- 32K Bytes Memory
- 5" Floppy Controller
- Serial RS232 20ma I/O
- Parallel Printer
- Real Time Clock
- Screen Printer Bus
- On Board Power Supply
- Solder Masked and Silkscreened

KEYBOARD

LNW80 KEYBOARD KIT $84.95

The Keyboard Kit contains a 63 key plus a 10 key, P.C. board, and remaining components.

LNDoubler + DOS Plus 3.2D

- Assembled and Tested $175.00

Double-density disk storage for the LNW Research's "System Expansion" or the Tandy's "Expansion Interface". The LNDoubler™ is totally software compatible with any double density software generated for the Percom's Doubler***. The LNDoubler™ provides the following outstanding features.

- Store up to 350K bytes on a single 5" disk
- Single and double density data separation
- Precision write precompensation circuit
- Software switch between single and double density
- Hardware override into single density only
- Easy plug in installation requiring no etch cuts, jumpers or soldering
- 35, 40, 77, 80 track 5" disk operation
- 120 day parts and labor Warranty

*** Doubler is a product of Percom Data Company, Inc.

DOS PLUS 3.2D

Micro Systems software's double density disk operating system. This operating system contains all the outstanding features of a well developed DOS, with ease in useability.

LNW DATA SEPERATOR

- Assembled and Tested $17.95

The LNW Data Separator provides you with a reliable and inexpensive means of solving your disk data read error problems for your 5" single density drives. Compatible with both the LNW System Expansion and Tandy's Expansion Interface. Some soldering is required.

CASE

LNW80 CASE . Add $12.00 for postage $84.95

The streamline design of this metal case will house the LNW80, LWN System Expansion, LNW80 Keyboard, power supply and fan, LNDoubler™, or LNW Data Separator. This kit includes all the hardware to mount all of the above.

PARTS AVAILABLE FROM LNW RESEARCH
4116 - 200ns RAM
6 chip set $26.00
8 chip set $33.50
16 chip set $64.00
24 chip set $94.00
32 chip set $124.00

- LNW80 "Start up parts set" LNW80-1 . . . . $82.00
- LNW80 "Video parts set" LNW80-2 . . . . $31.00
- LNW80 Transformer LNW80-3 . . . . $18.00
- LNW80 Keyboard cable LNW80-4 . . . . $16.00
- 40 Pin computer to expansion cable $15.00
- System Expansion Transformer $19.00
- Floppy Controller (FD1771) and UART (TR1602) . . . $30.00

LNW RESEARCH CORPORATION ✓53
14661-C MYFORD RD.
TUSTIN CA. 92680

VISA & MASTER CHARGE ACCEPTED, Add $3 for postage
ORDERS & INFO. NO. 714-552-8946
SERVICE NO. 714-641-8850

$0.00	Initial display
$0.01	First digit entered
$0.12	Second digit entered
$1.23	Third digit entered
$12.34	Fourth digit entered
$123.45	Fifth digit entered
$1,234.56	Sixth digit entered

The routine is an INPUT equivalent to PRINT USING.

I have used this routine for all of my business software and have found it to be fast, convenient and reliable. ■

```
50110 PRINTSTRING$(FL+1+T%,24);
50115 C!=C!+1
50120 IF C!>8 THEN IL$=LEFT$(IN$,LEN(IN$)-8)+","+MID$(I
      N$,LEN(IN$)-7,3)+","+RIGHT$(IN$,5) ELSE IF C!>5 TH
      EN IL$=LEFT$(IN$,LEN(IN$)-5)+","+RIGHT$(IN$,5) ELS
      E IL$=IN$
50125 IF C!>2PRINTSTRING$(((FL-5)-(C!-3))," ");:PRINT I
      S$;"$";LEFT$(IL$,LEN(IL$)-2);".";RIGHT$(IL$,2);:GO
      TO50140
50130 IF C!=1 THEN PRINTSTRING$(FL-5," ");IS$;"$0.0";IL
      $;
50135 IF C!=2 THEN PRINTSTRING$(FL-5," ");IS$;"$0.";IL$
      ;
50140 GOTO50040
50145 STOP
50150 T%=((LEN(IN$)-3)/3):IF T%<1 THEN T%=0: '--COMMA A
      DJ
50155 RETURN
```

Program Listing 2

```
>50000
>LIST
1  REM FORMATTED INPUT PROGRAM DEMONSTRATION DRIVER
2  REM ACCUMULATING CALCULATOR
3 REM DEMONSTRATION IS IN DOUBLE PRECISION BUT THE
SUBROUTINE AT 50000 WILL ALSO WORK IN SINGLE PRECISION
4  REM DEMONSTRATION IS A TYPICAL APPLICATION
10  CLS
20  PRINT TAB(15)"FORMATTED INPUT DEMONSTRATION"
30  CLEAR 100
40  DEFINT A-Z
50  PRINT"ENTER AN AMOUNT "::FF=10:FL=9:GOSUB 50000
60  T#=T#+VAL(IN$) :PRINT USING "$$#######,,.##";T#
70  GOTO 50
```

Explanations of Program Listing 1.

```
49999 LIST
50000 ========>>INPUT USING---HOW IT WORKS<<==========
50005 ADD 1 SPACE FOR $ SIGN, 1 SPACE FOR DECIMAL POINT
50010 T% REPRESENTS MAX # OF COMMAS
50015 CLEAR T% (AFTER THIS T% WILL REPRESENT ACTUAL # O
      F COMMAS)
50020 CLEAR STRING VARIABLES
50025 MOVE CURSOR TO BEGINNING POINT
50030 INITIAL DISPLAY (DOLLAR SIGN, ZERO, DECIMAL, AND
      TWO MORE ZEROES)
50035 IITIALIZE NUMERIC VARIABLES
50040 ENTRY POINT
50045 INKEY$ INPUT
50050 CHECK IF INKEY$ INPUT IS A  "C" --IF NOT JUMP TO
      50065
50055 PUT INTO T%, THE ACTUAL NUMBER OF COMMAS NEEDED F
      OR DISPLAY
50060 ERASE DISPLAY, RESTORE CURSOR, JUMP TO START OF E
      NTRY ROUTINE AT 50020
50065 CHECK IF CARRIAGE RETURN--A$=CHR$(13) IF NOT JUMP
      TO 50085
50070 FINAL OUTPUT ROUTINE--ESTABLISH PROPER DECIMAL PL
      ACEMENT
50075 INSERT MINUS SIGN IF NEEDED IN VARIABLE IN$
50080 DETERMINE IF SCREEN DISPLAY SHOULD BE INCREMENTE
50082 END OF FINAL OUTPUT ROUTINE----RESET SCREEN INCRE
      MENT FLAG 'FF' TO ZERO AND EXIT
50085 PREVENTS EXTNDING ENTRY BEYOND LIMIT SPECIFIED BY
      FL
50086 IGNORE NEXT TWO LINES IF COUNTER IS GREATER THAN
      ZERO
50090 ACQUIRE MINUS SIGN ONLY IF ENTERED BEFORE ANYTHIN
      G ELSE
50095 REJECT NON-NUMERIC ENTRY
50100 APPEND LAST ENTRY TO MAIN STORAGE STRING--IN$
50105 COMMA ADJUST
50110 MOVE CURSOR TO PROPER POSITION FOR DISPLAY STRING
      --IL$
50115 INCREMENT COUNTER
50120 CONFIGURE DISPLAY STRING--IL$--WITH PROPER # OF C
      OMMAS
50125 TO 50135 DISPLAY VALUE WITH PROPER DECIMAL LOCATI
      ON
50140 JUMP BACK TO INPUT POINT
50145 SAFETY STOP
50150 SUB TO DETERMINE PROPER # OF COMMAS IN CURRENT DI
      SPLAY
```

Your computer benefits from an interpreter, so why not you?

Foreign Language Translation

Dan L. Robinson
1625 Higgins Way
Pacifica, CA 94044

Can the Electric Pencil and the TRS-80 team up to translate foreign languages? They sure can!

If you must decipher a business letter from a client across the sea, research a technical paper to see how companies overseas have advanced, learn a language, or merely communicate with Grossmutter from the old country, then TRS-80 and Pencil can do it.

What makes Pencil into a foreign language translator is its string search and replacement mode. Simply type in: (space) FOREIGN WORD (space)/(space) ENGLISH WORD(space)/999 and presto! That word is changed to another language throughout the text. This means it is necessary to look up a foreign word only once.

Since most applications may be confined to a narrow field (you may be in the widget exporting business; your specialty may be laser brain surgery), a small vocabulary will probably suffice, adding only a few new words for each future translation.

Like everything else, there's a right and a wrong way to translate with Electric Pencil, and believe me, I've tried all the wrong ways.

Inputting Text

When typing in the original text, *don't* use Pencil's lowercase option. The search and replace mode won't recognize *The, the,* and *THE* as the same word, and you may have to replace the string three times unless you stick to uppercase. You can convert to lowercase later, if you wish, as you replace the foreign words with English.

Using uppercase avoids another confusing element; some languages don't follow the same conventions on capitalization that we do. For example, the Germans capitalize such words as Log, Distance and Poor Fool, even in the middle of a sentence. You might find that the important looking name with the capital letter translates to "Stuffy air" in English.

Be particularly careful with Pencil's famous dropped letters in the first word on a new line. The string search won't recognize a word with a missing letter, and probably later on neither will you.

Leave a space between words and puctuation marks. The string search won't recognize (space) WORD(space) as being the same as "WORD or WORD;.

If the foreign alphabet contains symbols not used in English, you may substitute @$% * + =, so long as you are consistent.

Make liberal use of markers such as ((((((PAGE 3)))))) so that you can easily find your place, or recognize your remarks as distinct from the text.

Building a Dictionary

If you are going to translate often, it's important to build a dictionary so you won't have to look up words more than once. As small as your widget-exporting or laser-brain-surgery vocabulary may be, it's easier if it is in alphabetical order, and being able to consult the foreign dictionary in alphabetical order is much quicker than leaping from A to Z and back to the same page.

The following program will permit you to input the foreign words at random. It will then present them in alphabetical order so that you may tack on their English counterparts. Add a routine at the end to save the data to a disk or tape file, and you'll be ready to use them again in the future:

```
 10 CLS:PRINT"DICTIONARY":CLEAR
    1500 ; DIM A$(100)
 20 PRINT:INPUT"NUMBER OF WORDS";
    C
 30 FOR D=1 TO C:PRINT"WORD";D;:
    INPUT A$(D):NEXT D
 40 FOR D=1 TO C:FOR E=1 TO C-1
 50 A$=A$(E):B$=A$(E+1):IF A$©B$
    THEN 70
 60 A$(E)=B$:A$(E+1)=A$
 70 NEXT E,D:CLS
 80 FOR D=1 TO C:PRINT A$(D);" = ";:
    INPUT Z$
 90 A$(D)=A$(D)+" = "+Z$:NEXT D
100 END
```

Making the Translation

Most languages, like our own, make larger words out of a number of smaller ones, and the meaning may have little relation to the elements. Therefore, it is important to first translate *words* and not *word elements*. For instance, changing short words might give an equivalent of "a mixture of nitrogen and oxygen" and "a tool for shaving wood" when what was meant was "airplane".

To avoid this pitfall, the string to be searched and its replacement should be placed between a pair of spaces. That way, only words and not word elements will be translated; and that is also why you input the text with a space between the words and the punctuation marks.

Making the replacement, you may now choose to have the English translation in lowercase.

In Pencil's string search mode, # is the wild card, and it can save a great deal of effort. Languages such as French, Spanish and Italian use slightly different words depending on whether the word is masculine or femine, although both would translate the same into English. Thus LOS and LAS in Spanish would both translate into THE in English using the # symbol: (space)L#S(space)/(space) THE(space)/999.

Many words have more than one meaning, and only when viewed in context is it possible to select the correct one. These words can be saved until the more obvious words have been replaced. There are two ways to deal with these multiple meaning words.

The first is a blunderbuss approach, throwing in some bell-ringing markers to get your attention as you later review the text: (space)FOREIGN WORD (space)/(space)$$$ENGLISH WORD$$$(space).

As you come across $$$ WORD$$$ in the text, you can decide if you have selected the proper meaning, and the markers can be removed once you are satisfied.

The second method is to use the search function without a replacement. Pencil will put the cursor at the beginning of the line containing the string, and you may then type in the appropriate translation. Merely type C to continue the search to the next string location.

After all the major words have been changed, it's time to deal with those tricky word elements.

In English, the first or last few letters often modify a word and give its true meaning. For example, the UN in UNable. Dealing with prefixes and suffixes is just a matter of spacing:

(space)FOREIGN PREFIX/(space) ENGLISH PREFIX/999
FOREIGN SUFFIX(space)/ENGLISH SUFFIX (space)/999
FOREIGN ELEMENT/ENGLISH ELEMENT/ 999

Translating parts of words can be hazardous, since the actual meaning isn't always the same as the literal translation. That's why word elements should be changed only after whole words have been translated.

There will probably be a few words for which you have been unable to find an English counterpart in your $1.95 paperback dictionary. By this time, you may have translated enough so that the meaning of these mystery words is made clear by the context. If not, insert * as a marker. You can use the string search to locate the tough words so that they can be listed for a later trip to the library's foreign language dictionary.

Page 1 of your first translating effort is a task, since you'll have to look up every word. As your mini-dictionary is built up, you'll discover how easy it is to use the TRS-80 and Electric Pencil to change confounding foreign letters and articles into English.■

80 Program Watch!

Converted any 80 programs lately?

80 is looking for updates to its published programs. Have you converted a program to run on a Model I? Model II? Model III? The Color Computer or the Pocket Computer?

Do your neighbors a favor and make yourself some extra

CA$H

Send them to the office in care of:
Jake Commander
80 Pine Street Peterborough, NH 03458

✓See List of Advertisers on page 306

A cassette based, double entry accounting system for the small business.

The General Ledger

R. L. Conhaim
15506 Kiamichi Rd #1
Apple Valley, CA 92307

One of the most practical applications of the microcomputer is the keeping of the financial records for small to medium-size businesses.

Among important records is the general ledger for which many packaged programs are available.

Unfortunately for the owner of a very small business, they are often too complex. Also, there is often a sizeable investment required for the hardware and software. Just keeping such small programs up to date may also require too much time. The small business, therefore, needs only those records which pare it down to provide tax information; secure loans; make mandatory reports, and provide the owners with information about the status of the business. Anything more is overkill.

Many general ledger programs available require disk memory, at least 32K of RAM, and a considerable amount of time to keep the records current. In addition, there are single entry bookkeeping systems that become unusable when the business grows large enough to need double entry accounting.

No.	Account Name	Primary Variable	Secondary Variable	Location	Type
1	Petty Cash	A1		1580	A
2	Cash in Bank	A2		1600	A
3	Savings Account	A3		1620	A
4	Parts Inventory	A4		1640	A
5	Merchandise Inventory	A5		1660	A
6	Accounts Receivable	A6		1680	A
7	Furniture & Fixtures	A7		1700	A
8	Service Equipment	A8		1720	A
9	Trucks & Autos	A9		1740	A
10	Building	B1		1760	A
11	Keogh Account	B2		1780	A
12	Miscellaneous Fixed Assets	B3		1800	A
13	FICA Withheld	L1		1820	L
14	Employees' Taxes Withheld	L2		1840	L
15	Accounts Payable	L3		1860	L
16	Notes Payable	L4		1880	L
17	Sales Tax Due	L5		1900	L
18	Mortgage Payable	L6		1920	L
19	Keogh, Employees' Contrib.	L7		1940	L
20	Jenkins Capital	L8		1960	L
21	Sales, Service Labor	I1	J1	1980	I
22	Sales, Service Parts	I2	J2	2000	I
23	Sales, Merchandise	I3	J3	2020	I
24	Interest Earned	I4	J4	2040	I
25	Miscellaneous Income	I5	J5	2060	I
26	Purchases, Parts	E1	K1	2080	E
27	Purchases, Merchandise	E2	K2	2100	E
28	Cost of Goods Sold	E3	K3	2120	E
29	Payroll	E4	K4	2140	E
30	FICA Expense	E5	K5	2160	E
31	Auto & Truck Expense	E6	K6	2180	E
32	Bad Debts	E7	K7	2200	E
33	Telephone Expense	E8	K8	2220	E
34	Postage	E9	K9	2240	E
35	Interest Paid	F1	M1	2260	E
36	Dues & Subscriptions	F2	M2	2280	E
37	Insurance	F3	M3	2300	E
38	Depreciation	F4	M4	2320	E
39	Maintenance, Building	F5	M5	2340	E
40	Utilities,Gas, Elec.,Water	F6	M6	2360	E
41	Office Expense	F7	M7	2380	E
42	Service Equipment Maintenance	F8	M8	2400	E
43	Personal Property Tax	F9	M9	2420	E
44	Workmen's Compensation	G1	N1	2440	E
45	Shipping Expense	G2	N2	2460	E
46	Service Expense	G3	N3	2480	E
47	Miscellaneous Expense	G4	N4	2500	E
48	Jenkins Withdrawals	H1	N5	2520	-
49	Profit & Loss	H2	N6	2540	-

Sample 1. Jenkins List of Accounts

The system described can provide the small business with a practical double entry general ledger (TRS-80, Level II, 16K system). It uses the standard cassette and a peripheral printer. Data is kept on cassette so there is no practical limit to the number of entries but a limit to the number of accounts, 65 being practical.

A common sense rule is—if you only use an account occasionally, you probably don't need it. Extras may be lumped into a miscellaneous category.

The most common form of bookkeeping is called double entry. For every transaction, two or more entries are made. These entries are either positive or debit, or negative credit. For every transaction, the debits must equal the credits. Assets and expenses normally have debit balances, while liability, income, and capital accounts have credit balances. Accounts are identified by name and number.

Daily entries are normally made in a journal, of which there are several styles, while account totals are accumulated in a ledger. The general ledger is the principal source of information. Subsidiary ledgers can handle accounts receivable and accounts payable. These divide totals from the general ledger into customer and supplier names and amounts.

In a double entry system, the amounts between debits and credits must correspond and balance; if they don't, a mistake has been made. Checking the balance is called trial balancing in which all ledger accounts are added, the debits in one column and the credits in another. The totals of each must then match. The columns may be added together algebraically, looking for zero balance.

The advantage of double entry is its ability to point out errors quickly, and distribute amounts to accounts efficiently. Of course, since everything is written twice, the chance for making errors is there, but it may be minimized with a well-designed computer system.

The system here has been developing for three years, and was begun on a programmable TI-59 calculator and later adapted to the TRS-80. This provided more account memory with greater flexibility.

It has been through the throws of correction, adaptation, modification and expansion to the point where it satisfies several types of businesses. Two features include: a manually-maintained general ledger which can be updated whenever convenient. The journal serves as a check register, since many businesses receive income and make dispersals by check.

The system may be used with cash or accrual bookkeeping. The accrual method is a necessity for income tax purposes, where merchandise inventories are maintained. In this method, sales are considered made when the merchandise or service is delivered, and purchases or expenses are accrued when the purchased items are received.

The Journal

The journal in this system has a number of columns. There are three for cash—debit, credit and balance. The balance is maintained manually as a check on the computer balance's accuracy. It is one of many checks to insure that information is correctly entered. Most of the other columns are single-columns in which distributions of transactions are made. The accounts most often used have individual columns. A three-column catch-all, labeled account number, debit and credit takes care of less often used accounts.

To make inputting to the computer faster, a worksheet is used. It takes its data from the journal and lists the account numbers and amounts for debits, and account numbers and amounts for credits for each transaction. The sheet may seem unnecessary and increase the error possibilities, but working at the computer with a large, clumsy journal is indeed difficult. The use of a worksheet avoids errors.

The Computer

The computer part of the system uses three cassettes: one contains the program; a second, contains monthly data (and is used over and over, month after month), a third, accumulates year-to-date data

The operator enters the date, and indicates as to whether the data is monthly or accumulated annually. A menu then appears on the screen providing a choice of the following: making entries, running a trial balance, printing a profit and loss statement or a balance sheet, and performing the end of month procedure. The last choice provides a means of adding the monthly data to the annual accumulated data. When the end of the month's procedure is completed, the monthly data tape retains only the assets and liabilities accounts. Expense and income accounts have been zeroed so the tape is ready for the new month. The year-to-date tape retains all data for accounts cumulatively for the year.

The Workings of the System

The workings of this system are best illustrated by us creating a hypothetical business and setting up accounts for it. We'll call our business the Jenkins Service Company. It is a single proprietor establishment owned by George Jenkins. The business of the company is repairing and selling TV and radio sets. The repair business is the largest part of the company. Jenkins employs three people and owns two service trucks. There are 49 accounts in this business, but these are not necessarily typical of other businesses, nor are the amounts used.

The account breakdown in as follows: six current asset accounts, six fixed asset accounts, five current liability accounts, two fixed liability accounts, one capital (net worth) account, five income accounts, one cost of goods sold account, two purchases accounts and 21 expense accounts.

Any business may use the program—changing only the names of the accounts, along with making a few minor program changes, if needed. As we shall see, accounts may be added, for a total of perhaps 65.

A Closer Look

The program is fully interactive with the operator, providing instructions throughout for making entries and securing various reports. The program takes about 10K. (Note in line 30 we are using double precision for the various dollar amounts.) Early in the development of this program, we discovered that a combination of double precision and two decimal place print using statements got rid of those one-cent errors plaguing bookkeeping programs. Lines 60 and 70 define the print using formats.

When the entries portion of the program are selected, the screen asks for account number, amount, and whether the amount is a credit. The operator presses ENTER if the amount is a debit, but must enter C if it is a credit amount. If the amount is the same as a previous entry just hit ENTER in answer to the enter amount input. The previous entry is saved by the program in lines 260 through 290. This avoids possible errors which could be caused by re-entry of the same amount. In many transactions, there is only one debit and one credit account and the amount is the same for each.

Entries are printed out, and in case of an entry error, traced easily. Line 330 accumulates algebraically both debit and credit amounts. The result of this addition is printed as an entry balance after all entries are made. This is done in line 410 where we seek a 0.00 answer. This technique shows entry errors immediately, and you don't have to wait for a trail balance.

During the entry of data, ON

A/C #1 PETTY CASH	108.39
A/C #2 CASH IN BANK	5,965.20
A/C #3 SAVINGS ACCOUNT	2,715.42
A/C #4 PARTS INVENTORY	1,694.85
A/C #5 MERCHANDISE INVENTORY	3,040.40
A/C #6 ACCOUNTS RECEIVABLE	848.69
A/C #7 FURN. & FIXTURES, NET	2,493.70
A/C #8 SERVICE EQUIPMENT	5,630.21
A/C #9 TRUCKS AND AUTOS	8,424.98
A/C #10 BUILDING	113,443.51
A/C #11 KEOGH RETIREMENT	7,432.00
A/C #12 MISC. FIXED ASSETS	111.22
A/C #13 FICA WITHHELD	−84.84
A/C #14 EMPLOYEE TAXES WITHHELD	−449.25
A/C #15 ACCOUNTS PAYABLE	−521.10
A/C #16 NOTES PAYABLE	−1,500.00
A/C #17 SALES TAX DUE	−336.14
A/C #18 MORTGAGE PAYABLE	−89,966.73
A/C #19 KEOGH, EMPLOYEES CONTRIB.	−3,918.68
A/C #20 JENKINS CAPITAL	−55,307.83
A/C #21 SALES SERVICE LABOR	−7,591.43
A/C #22 SALES SERVICE PARTS	−2,483.50
A/C #23 SALES, MERCHANDISE	−4,118.73
A/C #24 INTEREST EARNED	−34.59
A/C #25 MISCL. INCOME	−10.56
A/C #26 PURCHASES, PARTS	1,533.26
A/C #27 PURCHASES, MERCHANDISE	2,101.01
A/C #29 PAYROLL	3,026.13
A/C #30 FICA EXPENSE	69.77
A/C #31 AUTO & TRUCK EXPENSE	142.33
A/C #32 BAD DEBTS	91.18
A/C #33 TELEPHONE EXPENSE	56.25
A/C #34 POSTAGE	15.00
A/C #35 INTEREST PAID	529.50
A/C #36 DUES & SUBSCRIPTIONS	17.00
A/C #37 INSURANCE	201.00
A/C #39 MAINTENANCE, BUILDING	76.34
A/C #40 UTILITIES, GAS, ELEC., WATER	207.60
A/C #41 OFFICE EXPENSE	31.04
A/C #42 SERV. EQUIP. MAINTENANCE	18.95
A/C #45 SHIPPING EXPENSE	1.45
A/C #46 SERVICE EXPENSE	51.07
A/C #47 MISCL. EXPENSE	4.14
A/C #48 JENKINS WITHDRAWALS	2,300.00
A/C #49 PROFIT AND LOSS	3,765.77
DEBIT/CREDIT DIFFERENCE =	$0.00

Sample 2. Typical Trial Balance from Annual Cumulative Data

ASSETS		
CURRENT		
A/C #1 PETTY CASH	108.39	
A/C #2 CASH IN BANK	5,965.20	
A/C #3 SAVINGS ACCOUNT	2,715.42	
A/C #4 PARTS INVENTORY	1,694.85	
A/C #5 MERCHANDISE INVENTORY	3,040.40	
A/C #6 ACCOUNTS RECEIVABLE	848.69	
TOTAL CURRENT		$14,372.96
FIXED		
A/C #7 FURN. & FIXTURES, NET	2,493.70	
A/C #8 SERVICE EQUIPMENT	5,830.21	
A/C #9 TRUCKS AND AUTOS	8,424.98	
A/C #10 BUILDING	113,443.51	
A/C #11 KEOGH RETIREMENT	7,432.00	
A/C #12 MISC. FIXED ASSETS	111.22	
TOTAL FIXED		$137,735.62
TOTAL ASSETS		$152,108.57
LIABILITIES		
CURRENT		
A/C #13 FICA WITHHELD	−84.84	
A/C #14 EMPLOYEE TAXES WITHHELD	−449.25	
A/C #15 ACCOUNTS PAYABLE	−521.10	
A/C #16 NOTES PAYABLE	−1,500.00	
A/C #17 SALES TAX DUE	−338.14	
TOTAL CURRENT		−$2,893.33
FIXED & NET WORTH		
A/C #18 MORTGAGE PAYABLE	−89,988.73	
A/C #19 KEOGH, EMPLOYEES CONTRIB.	−3,918.68	
A/C #20 JENKINS CAPITAL	−55,307.83	
TOTAL FIXED & NET WORTH		−$149,215.24
TOTAL LIABILITIES & NET WORTH		−$152,108.57

Sample 3. Typical Balance Sheet

GOSUB commands are used. Because of the number of accounts, you can't get all GOSUB line numbers on one line. Lines 350 through 390 show how two lines are used.

The Trial Balance

The trial balance prints out all accounts one after the other, with their positive (debit balances) or negative (credit balances). Accounts not used have zero balances and are not printed. The amounts are accumulated algebraically in line 580, and the results printed in lines 600 and 610. Here again we are looking for a zero balance. The recording of updated information is accomplished in lines 620 through 650. The CHR$(12) PRINT command in line 660 advances the paper in the author's printer to the next sheet. The trial balance may be used with monthly or year-to-date data, and is customarily used with both.

The profit and loss statement for year-to-date data is the conventional income and expense statement. The cost of goods sold, in the Jenkins report, prints out with a zero balance. This is a reminder to the reader that the account has not been journalized. It is normally figured from opening and closing inventories and purchases. In a small business, the taking of inventory is usually done once a year. It can be estimated, however it wasn't done in the Jenkins P & L. This assumes purchases account for all merchandising and parts sold.

Note that net profit is figured by deducting expenses and withdrawals from the gross profit amount. For income tax purposes, withdrawals cannot be called an expense and the profit amount calculated in lines 880 and 890 as income is used. Since income accounts have a credit balance, the sign is changed in line 880 by dividing by −1 to get a correct profit calculation.

There is another way to calculate net profit. This method is used in the end-of-month procedure as a check on the accuracy of the amount in the P & L statement.

Balance Sheet

The balance sheet is the most important indicator of the health of a business. The method used here is a straightforward presentation of assets, liabilities and net worth. All the dollar amounts are taken from their respective accounts; none is calculated in preparing the statement. Assets should balance liabilities and net worth. In addition, not all businesses require fixed and current assets or liabilities, but they are provided in this case to show programming possibilities.

End of Month Procedure

At the end of each month or accounting period, it is necessary to add the month's figures to the year-to-date or annual totals. Since the year-to-date and monthly accounts use the same variables, however, we can't read the monthly tape without destroying the annual one. To stop this, move the data from the annual to other variable storage registers. Provide variables for expense and income accounts. Then add the prime variables from the monthly tape to the annual to give an updated total for the year. Note: we do not need this technique for assets and liabilities, since these are correct each month. We merely destroy outdated annual information in the process.

Change overs are incorporated in lines 1360 to 1560. There are a couple of items that may need clarifying, however. There is another way to determine profit and loss. It is the difference between Jenkins' capital account for the previous period and the capital account for the present. First compute the capital for the present by adding asset and liability accounts algebraically (line 1460). The net remainder is the amount of capital. Using the current capital amount, profit and loss is calculated in line 1470. The capital account is corrected in line 1480. We can now write the updated annual tape.

To upgrade the monthly tape for use in the next month, we want to preserve only the asset and liability accounts, and zero the income and expense accounts. The zeroing is done in line 1520, after which the monthly tape is written, ready for the

INCOME		
A/C #21 SALES SERVICE LABOR	−7,591.43	
A/C #22 SALES SERVICE PARTS	−2,483.50	
A/C #23 SALES, MERCHANDISE	−4,118.73	
A/C #24 INTEREST EARNED	−34.59	
A/C #25 MISCL. INCOME	−10.56	
TOTAL INCOME		−$14,238.81
COST OF GOODS SOLD		$0.00
GROSS PROFIT		−$14,238.81
EXPENSES		
A/C #26 PURCHASES, PARTS	1,533.28	
A/C #27 PURCHASES, MERCHANDISE	2,101.01	
A/C #29 PAYROLL	3,026.13	
A/C #30 FICA EXPENSE	69.77	
A/C #31 AUTO & TRUCK EXPENSE	142.33	
A/C #32 BAD DEBTS	91.18	
A/C #33 TELEPHONE EXPENSE	56.25	
A/C #34 POSTAGE	15.00	
A/C #35 INTEREST PAID	529.50	
A/C #36 DUES & SUBSCRIPTIONS	17.00	
A/C #37 INSURANCE	201.00	
A/C #39 MAINTENANCE, BUILDING	76.34	
A/C #40 UTILITIES, GAS, ELEC., WATER	207.60	
A/C #41 OFFICE EXPENSE	31.04	
A/C #42 SERV. EQUIP. MAINTENANCE	18.96	
A/C #45 SHIPPING EXPENSE	1.45	
A/C #46 SERVICE EXPENSE	51.07	
A/C #47 MISCL. EXPENSE	4.14	
TOTAL EXPENSES		$8,173.04
PROFIT		$6,065.77
LESS JENKINS WITHDRAWALS		$2,300.00
NET PROFIT		$3,765.77

Sample 4. Typical Profit and Loss Statement

A/C #1 PETTY CASH	108.39
A/C #2 CASH IN BANK	5,965.20
A/C #3 SAVINGS ACCOUNT	2,715.42
A/C #4 PARTS INVENTORY	1,694.85
A/C #5 MERCHANDISE INVENTORY	3,040.40
A/C #6 ACCOUNTS RECEIVABLE	848.69
A/C #7 FURN. & FIXTURES, NET	2,493.70
A/C #8 SERVICE EQUIPMENT	5,830.21
A/C #9 TRUCKS AND AUTOS	8,424.98
A/C #10 BUILDING	113,443.51
A/C #11 KEOGH RETIREMENT	7,432.00
A/C #12 MISC. FIXED ASSETS	111.22
A/C #13 FICA WITHHELD	−84.84
A/C #14 EMPLOYEE TAXES WITHHELD	−449.25
A/C #15 ACCOUNTS PAYABLE	−521.10
A/C #16 NOTES PAYABLE	−1,500.00
A/C #17 SALES TAX DUE	−338.14
A/C #18 MORTGAGE PAYABLE	−89,988.73
A/C #19 KEOGH, EMPLOYEES CONTRIB.	−3,918.68
A/C #20 JENKINS CAPITAL	−55,307.83
DEBIT/CREDIT DIFFERENCE =	−$0.00

Sample 5. Monthly Data After Clearing

next month's entries. Although the end of the period procedure is a little complex, the operator sees only the read and write requests in lines 1370, 1420, 1500 and 1540.

If you stay in the account breakdown listed, you should have no trouble adapting the program to your needs. If the Jenkins case uses more accounts than you, just show zero balances for the unused accounts.

The account numbering system shown here is required for the several ON GOSUB functions. You could use your own account numbers followed by dash numbers. Entries would be made by dash number.

Adding more expense accounts is fairly simple. Suppose, for example, that you also need an advertising account. Try this:

```
Delete 2560
2560 G5 = G5 + C5; If G5 = 0 Return
2580 LPRINT "A/C #50 Advertising Expense";Tab(40);Using A$; G5:Return
2600 END
390 (Add by Editing ',2560' after 2500)
550 (Add by Editing ',2560' After 2500)
570 (Change '49' to '50')
580 (Add ' + G5' at end of line)
840 (Add',2560' at end of line)
860 (Add ' + G5' at end of line)
3030 (Add ',G5' at end of line)
4030 (Add ',G5' at end of line)
```

This system, as exemplified by the Jenkins example has proved to be a practical time saver. ■

```
1Ø 'JENKINS SERVICE COMPANY GENERAL LEDGER
2Ø DEFSTR Q
3Ø DEFDBL A-P
4Ø CLS
6Ø LET A$ = "###,###.##"
7Ø LET D$ = "$$###,###.##"
8Ø C1 = Ø
9Ø INPUT "LOAD DATA TAPE. PRESS ENTER";C1
1ØØ IF C1 = Ø GOSUB 3ØØØ ELSE GOTO 8Ø
11Ø INPUT "ENTER DATE AS MO/DA/YR";Q
12Ø INPUT "IF MONTHLY DATA ENTER 1. IF ANNUAL CUMULATIV
    E DATA ENTER 2";C2
13Ø CLS
14Ø PRINT TAB(2Ø); "MENU": PRINT
15Ø PRINT "1 FOR ENTRIES"
16Ø PRINT "2 FOR TRIAL BALANCE"
17Ø PRINT "3 FOR PROFIT AND LOSS STATEMENT"
18Ø PRINT "4 FOR BALANCE SHEET"
19Ø PRINT "5 FOR END OF MONTH PROCEDURE":PRINT
2ØØ INPUT "ENTER SELECTION NUMBER";C3
21Ø IF C3 <> 1 GOTO 43Ø
22Ø LPRINT "JENKINS SERVICE ENTRIES AS OF ";Q
23Ø IF C2 = 1 LPRINT "MONTHLY DATA" ELSE LPRINT "ANNUAL
    CUMULATIVE DATA"
24Ø INPUT "ENTER ACCOUNT NUMBER. IF NONE ENTER Ø";C4
25Ø IF C4 = Ø GOTO 41Ø
26Ø INPUT "ENTER AMOUNT. IF SAME PRESS ENTER";C5
27Ø IF C5 = Ø GOTO 29Ø
28Ø C6 = C5: GOTO 3ØØ
29Ø C5 = C6
3ØØ C$ = "Ø"
31Ø INPUT "ENTER C IF CREDIT AMOUNT";C$
32Ø IF C$ = "C" THEN C5 = -C5
33Ø C7 = C7 + C5
34Ø LPRINT TAB(3Ø) "#"; C4; TAB(4Ø) C5
35Ø IF C4 >45 GOTO 38Ø
36Ø ON C4 GOSUB 158Ø,16ØØ,162Ø,164Ø,166Ø,168Ø,17ØØ,172Ø
    ,174Ø,176Ø,178Ø,18ØØ,182Ø,184Ø,186Ø,188Ø,19ØØ,192Ø
    ,194Ø,196Ø,198Ø,2ØØØ,2Ø2Ø,2Ø4Ø,2Ø6Ø,2Ø8Ø,21ØØ,212Ø
    ,214Ø,216Ø,218Ø,22ØØ,222Ø,224Ø,226Ø,228Ø,23ØØ,232Ø
    ,234Ø,236Ø,238Ø,24ØØ,242Ø,244Ø,246Ø
37Ø GOTO 4ØØ
38Ø P9 = C4-45
39Ø ON P9 GOSUB 248Ø ,25ØØ ,252Ø ,254Ø
4ØØ C5 = Ø:GOTO 24Ø
41Ø CLS:LPRINT :LPRINT "ENTRY BALANCE";USING A$;C7
42Ø INPUT "FOR THE MENU, TYPE 1";X1:IF X1 = 1 GOTO 13Ø

43Ø IF C3 = 2 GOTO 47Ø
44Ø IF C3 = 3 GOTO 68Ø
45Ø IF C3 = 4 GOTO 94Ø
46Ø IF C3 = 5 GOTO 136Ø
47Ø LPRINT "JENKINS SERVICE AS OF ";Q
48Ø IF C2 = 1 LPRINT "MONTHLY DATA" ELSE LPRINT "ANNUAL
    CUMULATIVE DATA"
49Ø LPRINT TAB(4) STRING$(15,"-");"TRIAL BALANCE";STRIN
    G$(15,"-"):LPRINT
5ØØ C8 = 1
51Ø IF C8 >45 GOTO 54Ø
52Ø ON C8 GOSUB 158Ø,16ØØ,162Ø,164Ø,166Ø,168Ø,17ØØ,172Ø
    ,174Ø,176Ø,178Ø,18ØØ,182Ø,184Ø,186Ø,188Ø,19ØØ,192Ø
    ,194Ø,196Ø,198Ø,2ØØØ,2Ø2Ø,2Ø4Ø,2Ø6Ø,2Ø8Ø,21ØØ,212Ø
    ,214Ø,216Ø,218Ø,22ØØ,222Ø,224Ø,226Ø,228Ø,23ØØ,232Ø
    ,234Ø,236Ø,238Ø,24ØØ,242Ø,244Ø,246Ø
53Ø GOTO 56Ø
54Ø C9 = C8-45
55Ø ON C9 GOSUB 248Ø ,25ØØ ,252Ø ,254Ø
56Ø C8 = C8 + 1
57Ø IF C8>49 GOTO 58Ø  ELSE 51Ø
58Ø X3 = A1+A2+A3+A4+A5+A6+A7+A8+A9+B1+B2+B3+L1+L2+L3+L
    4+L5+L6+L7+L8+I1+I2+I3+I4+I5+E1+E2+E3+E4+E5+E6+E7+
    E8+E9+F1+F2+F3+F4+F5+F6+F7+F8+F9+G1+G2+G3+G4+H1+H2
59Ø LPRINT
6ØØ LPRINT TAB(15) "DEBIT/CREDIT DIFFERENCE = ";
61Ø LPRINT USING D$;X3
62Ø D2 = Ø:INPUT "TO RECORD TRIAL BALANCE PREPARE DATA
    CASSETTE FOR RECORDING AND PRESS ENTER";D2
63Ø IF D2<> Ø GOTO 13Ø
64Ø PRINT "TRIAL BALANCE BEING RECORDED"
65Ø GOSUB 4ØØØ
66Ø LPRINT CHR$(12)
67Ø GOTO 13Ø
68Ø LPRINT "JENKINS SERVICE CO. AS OF ";Q
69Ø IF C2 = 1 LPRINT "MONTHLY DATA" ELSE LPRINT "ANNUAL
    CUMULATIVE DATA"
7ØØ LPRINT TAB(1Ø) STRING$(15,"-");"PROFIT & LOSS STATE
    MENT";STRING$(15,"-")
71Ø LPRINT
72Ø LPRINT "INCOME"
73Ø GOSUB 198Ø
74Ø GOSUB 2ØØØ
75Ø GOSUB 2Ø2Ø
76Ø GOSUB 2Ø4Ø
77Ø GOSUB 2Ø6Ø
78Ø D3 = I1+I2+I3+I4+I5
79Ø LPRINT TAB(1Ø)"TOTAL INCOME";TAB(5Ø) USING D$;D3
8ØØ LPRINT TAB(1Ø)"COST OF GOODS SOLD";TAB(5Ø)USING D$;
    E3
81Ø LPRINT TAB(2Ø)"GROSS PROFIT";TAB(5Ø)USING D$;D3-E3
82Ø LPRINT:LPRINT "EXPENSES"
83Ø D4=1
84Ø ON D4 GOSUB 2Ø8Ø ,21ØØ ,214Ø ,216Ø ,218Ø ,22ØØ ,222
    Ø ,224Ø ,226Ø ,228Ø ,23ØØ ,232Ø ,234Ø ,236Ø ,238Ø
    ,24ØØ ,242Ø ,244Ø ,246Ø ,248Ø ,25ØØ
85Ø D4 = D4+1:IF D4<>22 GOTO 84Ø
86Ø D5=E1+E2+E4+E5+E6+E7+E8+E9+F1+F2+F3+F4+F5+F6+F7+F8+
    F9+G1+G2+G3+G4
87Ø LPRINT TAB(1Ø) "TOTAL EXPENSES";TAB(5Ø)USING D$;D5
88Ø D6 = ((D3-E3)/-1)-D5
89Ø LPRINT TAB(1Ø)"PROFIT";TAB(5Ø)USING D$;D6
9ØØ LPRINT TAB(1Ø)"LESS JENKINS WITHDRAWALS";TAB(5Ø)USI
    NG D$;H1
91Ø LPRINT:LPRINT TAB(1Ø)"NET PROFIT";TAB(5Ø)USING D$;D
    6-H1
92Ø LPRINT CHR$(12)
93Ø INPUT "FOR THE MENU, TYPE 1";X2: IF X2 = 1 GOTO 13Ø

94Ø LPRINT "JENKINS SERVICE CO. AS OF ";Q
95Ø IF C2 = 1 LPRINT "MONTHLY DATA" ELSE LPRINT "ANNUAL
    CUMULATIVE DATA"
96Ø LPRINT TAB(1Ø) STRING$(15,"-");"BALANCE SHEET";STRI
    NG$(15,"-"):LPRINT
97Ø LPRINT "ASSETS"
98Ø LPRINT TAB(1Ø)"CURRENT"
99Ø GOSUB 158Ø
1ØØØ GOSUB 16ØØ
1Ø1Ø GOSUB 162Ø
1Ø2Ø GOSUB 164Ø
1Ø3Ø GOSUB 166Ø
1Ø4Ø GOSUB 168Ø
1Ø5Ø D7 = A1+A2+A3+A4+A5+A6
1Ø6Ø LPRINT TAB(1Ø)"TOTAL CURRENT";TAB(5Ø)USING D$;D7
1Ø7Ø LPRINT:LPRINT TAB(1Ø)"FIXED"
1Ø8Ø GOSUB 17ØØ
1Ø9Ø GOSUB 172Ø
11ØØ GOSUB 174Ø
111Ø GOSUB 176Ø
112Ø GOSUB 178Ø
113Ø GOSUB 18ØØ
114Ø D8 =A7+A8+A9+B1+B2+B3
115Ø LPRINT TAB(1Ø)"TOTAL FIXED";TAB(5Ø)USING D$;D8:LPR
     INT
116Ø LPRINT TAB(2Ø)"TOTAL ASSETS";TAB(5Ø)USING D$;D7+D8
     :LPRINT
117Ø LPRINT "LIABILITIES"
118Ø LPRINT TAB(1Ø)"CURRENT"
119Ø GOSUB 182Ø
12ØØ GOSUB 184Ø
121Ø GOSUB 186Ø
122Ø GOSUB 188Ø
123Ø GOSUB 19ØØ
124Ø D9 = L1+L2+L3+L4+L5
125Ø LPRINT TAB(1Ø)"TOTAL CURRENT";TAB(5Ø)USING D$;D9:L
126Ø LPRINT TAB(1Ø)"FIXED & NET WORTH"
127Ø GOSUB 192Ø
128Ø GOSUB 194Ø
129Ø GOSUB 196Ø
13ØØ P1 = L6+L7+L8
131Ø LPRINT TAB(1Ø) "TOTAL FIXED & NET WORTH";TAB(5Ø)US
     ING D$;P1
132Ø LPRINT:LPRINT TAB(2Ø) "TOTAL LIABILITIES & NET WOR
     TH";TAB(5Ø)USING D$;D9+P1
```

Program continues

EAST COAST

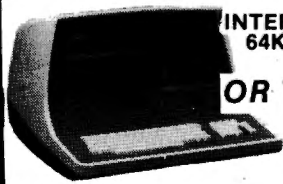
ΩMEGA Sales Co.
12 Meeting St.
Cumberland, RI 02864
1-800-556-7586
1-401-722-1027

ΩMEGA SALES CO.

WEST COAST

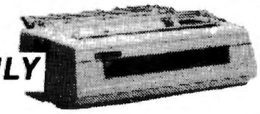
ΩMEGA Sales Co.
3353 Old Conejo Rd. #102
Newbury Park, CA 91320
1-800-235-3581
1-805-499-3678

SPECIAL OF THE MONTH!!

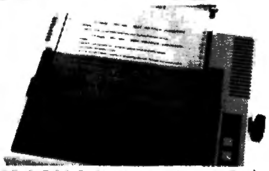


INTERTEC SUPERBRAIN 64K RAM $2849

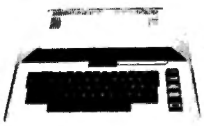
OR TOGETHER FOR ONLY $5195



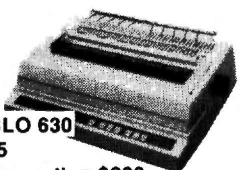
NEC 5510 SPINWRITER $2395



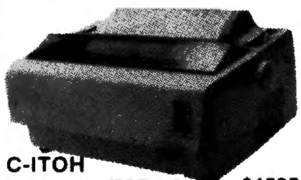
EPSON MX-80 PRINTER $479
INTERFACES: IEEE $55, TRS•80 $35, APPLE INTERFACE + CABLE $90, RS•232 $70



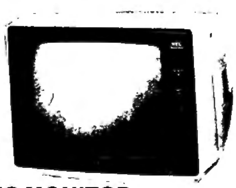
ATARI 800 $759



DIABLO 630 $1995
Tractor option $200



C-ITOH STARWRITER 25 $1595



NEC MONITOR $219

OKIDATA MICROLINE-83	$949
OKIDATA MICROLINE-80	$469
APPLE II PLUS 48K	$1150
APPLE DISK w/ 3.3 DOS Controller	$545
APPLE DISK w/o Controller	$435
BASE II PRINTER	$649
HAZELTINE 1420	$799
NORTHSTAR HORIZON II 32K QD	$2975
ATARI 400 8K	$399
ATARI 400 16K	$499
RADIO SHACK 16K Level II Model 3	$875
RADIO SHACK 64K Model 2	$3449
ANADEX DP-9500	$1249
TELEVIDEO 912C	$669
TELEVIDEO 920C	$729
TELEVIDEO 950	$1049
ATARI 825 PRINTER	$729
ATARI 850 INTERFACE	$169
ATARI 810 DISK	$449
(Call for price list of ATARI Software)	

- *NO SURCHARGE FOR CREDIT CARDS*
- *WE ACCEPT C.O.D.'S*
- *ALL EQUIPMENT FACTORY FRESH W/MANUFACTURER'S WARRANTY*
- *STOCK SHIPMENTS USUALLY SAME DAY OR NEXT DAY*
- *NO HIDDEN CHARGES - WE LIVE BY OUR PUBLISHED PRICES*

VISA **EAST COAST / WEST COAST** master charge

ΩMEGA SALES COMPANY

✓389

✓See List of Advertisers on page 306

```
1330 LPRINT CHR$(12)
1340 INPUT "FOR ANOTHER CHOICE TYPE 1";X:IF X = 1 GOTO
1350 'END OF PERIOD PROCEDURE STARTS HERE
1360 P2 = 0
1370 INPUT "READ ANNUAL TAPE.WHEN READY PRESS ENTER";P2
1380 IF P2 = 0 GOSUB 3000 ELSE 1360
1390 J1=I1:J2=I2:J3=I3:J4=I4:J5=I5:K1=E1:K2=E2:K3=E3:K4
     =E4:K5=E5:K6=E6:K7=E7:K8=E8:K9=E9:M1=F1:M2=F2:M3=F
     3:M4=F4:M5=F5:M6=F6:M7=F7:M8=F8:M9=F9:N1=G1:N2=G2:
     N3=G3:N4=G4:N5=H1:N6=H2
1400 P3 = L8/-1
1410 P4 = 0
1420 INPUT "READ MONTHLY TAPE. WHEN READY PRESS ENTER";
     P4
1430 IF P4 =0 GOSUB 3000 ELSE 1410
1440 I1=I1+J1:I2=I2+J2:I3=I3+J3:I4=I4+J4:I5=I5+J5:E1=E1
     +K1:E2=E2+K2:E3=E3+K3:E4=E4+K4:E5=E5+K5:E6=E6+K6:E
     7=E7+K7:E8=E8+K8:E9=E9+K9:F1=F1+M1:F2=F2+M2:F3=F3+
     M3:F4=F4+M4:F5=F5+M5:F6=F6+M6:F7=F7+M7:F8=F8+M8:F9
     =F9+M9:G1=G1+N1:G2=G2+N2:G3=G3+N3
1450 G4=G4+N4:H1=H1+N5:H2=H2+N6
1460 P5=A1+A2+A3+A4+A5+A6+A7+A8+A9+B1+B2+B3+L1+L2+L3+L4
     +L5+L6+L7
1470 H2 = N6+(L8-(P5/-1))
1480 L8=P5/-1
1490 P7=0
1500 INPUT"WRITE ANNUAL TAPE. WHEN READY PRESS ENTER";P
     7
1510 IF P7<>0 GOTO 1490 ELSE GOSUB 4000
1520 I1=0:I2=0:I3=0:I4=0:I5=0:E1=0:E2=0:E3=0:E4=0:E5=0:
     E6=0:E7=0:E8=0:E9=0:F1=0:F2=0:F3=0:F4=0:F5=0:F6=0:
     F7=0:F8=0:F9=0:G1=0:G2=0:G3=0:G4=0:H1=0:H2=0
1530 P8=0
1540 INPUT "WRITE MONTHLY TAPE.WHEN READY PRESS ENTER";
     P8
1550 IF P8<>0 GOTO 1530 ELSE GOSUB 4000
1560 GOTO 130
1570 'ACCOUNT I.D. STARTS HERE
1580 A1=A1+C5:IF A1 = 0 RETURN
1590 LPRINT "A/C #1 PETTY CASH";TAB(40)USING A$;A1:RETU
     RN
1600 A2=A2+C5:IF A2 = 0 RETURN
1610 LPRINT "A/C #2 CASH IN BANK";TAB(40)USINGA$;A2:RET
     URN
1620 A3=A3+C5:IF A3 = 0 RETURN
1630 LPRINT "A/C #3 SAVINGS ACCOUNT";TAB(40)USING A$;A3
     :RETURN
1640 A4=A4+C5:IF A4 = 0 RETURN
1650 LPRINT "A/C #4 PARTS INVENTORY";TAB(40)USING A$;A4
     :RETURN
1660 A5=A5+C5: IF A5 = 0 RETURN
1670 LPRINT "A/C #5 MERCHANDISE INVENTORY";TAB(40)USING
      A$;A5:RETURN
1680 A6=A6+C5:IF A6 = 0 RETURN
1690 LPRINT "A/C #6 ACCOUNTS RECEIVABLE";TAB(40)USING A
     $;A6:RETURN
1700 A7=A7+C5:IF A7 = 0 RETURN
1710 LPRINT "A/C #7 FURN. & FIXTURES,NET";TAB(40)USING
     A$;A7:RETURN
1720 A8=A8+C5: IF A8 = 0 RETURN
1730 LPRINT "A/C #8 SERVICE EQUIPMENT";TAB(40)USING A$;
     A8:RETURN
1740 A9=A9+C5: IF A9 = 0 RETURN
1750 LPRINT "A/C #9 TRUCKS AND AUTOS";TAB(40)USING A$;A
     9:RETURN
1760 B1=B1+C5:IF B1 = 0 RETURN
1770 LPRINT "A/C #10 BUILDING";TAB(40)USING A$;B1:RETUR
     N
1780 B2=B2+C5:IF B2 = 0 RETURN
1790 LPRINT "A/C #11 KEOGH RETIREMENT";TAB(40)USING A$;
     B2:RETURN
1800 B3=B3+C5: IF B3 = 0 RETURN
1810 LPRINT "A/C #12 MISC. FIXED ASSETS";TAB(40)USING A
     $;B3:RETURN
1820 L1=L1+C5: IF L1 = 0 RETURN
1830 LPRINT "A/C #13 FICA WITHHELD";TAB(40)USING A$;L1:
     RETURN
1840 L2=L2+C5: IF L2 = 0 RETURN
1850 LPRINT "A/C #14 EMPLOYEE TAXES WITHHELD";TAB(40)US
     ING A$;L2:RETURN
1860 L3=L3+C5: IF L3 = 0 RETURN
1870 LPRINT "A/C #15 ACCOUNTS PAYABLE";TAB(40)USING A$;
     L3:RETURN
1880 L4=L4+C5: IF L4 = 0 RETURN
1890 LPRINT "A/C #16 NOTES PAYABLE";TAB(40)USING A$;L4:
     RETURN
1900 L5=L5+C5: IF L5 = 0 RETURN
1910 LPRINT "A/C #17 SALES TAX DUE";TAB(40)USING A$;L5:
     RETURN
1920 L6=L6+C5: IF L6 = 0 RETURN
1930 LPRINT "A/C #18 MORTGAGE PAYABLE";TAB(40)USING A$;
     L6:RETURN
1940 L7=L7+C5: IF L7 = 0 RETURN
1950 LPRINT "A/C #19 KEOGH,EMPLOYEES CONTRIB.";TAB(40)U
     SING A$;L7:RETURN
1960 L8=L8+C5: IF L8 = 0 RETURN
```

Program continues

```
1970 LPRINT "A/C #20 JENKINS CAPITAL";TAB(40)USING A$;L
     8: RETURN
1980 I1=I1+C5:IF I1 = 0 RETURN
1990 LPRINT "A/C #21 SALES SERVICE LABOR";TAB(40)USING
     A$;I1:RETURN
2000 I2=I2+C5: IF I2 = 0 RETURN
2010 LPRINT "A/C #22 SALES SERVICE PARTS";TAB(40)USING
     A$;I2:RETURN
2020 I3=I3+C5:IF I3 = 0 RETURN
2030 LPRINT "A/C #23 SALES, MERCHANDISE";TAB(40)USING A
     $;I3:RETURN
2040 I4=I4+C5:IF I4 = 0 RETURN
2050 LPRINT "A/C #24 INTEREST EARNED";TAB(40)USING A$;I
     4:RETURN
2060 I5=I5+C5: IF I5 = 0 RETURN
2070 LPRINT "A/C #25 MISCL. INCOME";TAB(40)USING A$;I5:
     RETURN
2080 E1=E1+C5: IF E1 = 0 RETURN
2090 LPRINT "A/C #26 PURCHASES,PARTS";TAB(40)USING A$;E
     1:RETURN
2100 E2=E2+C5:IF E2 = 0 RETURN
2110 LPRINT "A/C #27 PURCHASES, MERCHANDISE";TAB(40)USI
     NG A$;E2:RETURN
2120 E3=E3+C5: IF E3 = 0 RETURN
2130 LPRINT "A/C #28 COST OF GOODS SOLD";TAB(40)USING A
     $;E3:RETURN
2140 E4=E4+C5: IF E4 = 0 RETURN
2150 LPRINT "A/C #29 PAYROLL";TAB(40)USING A$;E4:RETURN
2160 E5=E5+C5: IF E5 = 0 RETURN
2170 LPRINT "A/C #30 FICA EXPENSE";TAB(40)USING A$;E5:R
     ETURN
2180 E6=E6+C5: IF E6 = 0 RETURN
2190 LPRINT "A/C #31 AUTO & TRUCK EXPENSE";TAB(40)USING
      A$;E6:RETURN
2200 E7=E7+C5: IF E7 = 0 RETURN
2210 LPRINT "A/C #33 TELEPHONE EXPENSE";TAB(40)USING A$
     RN
2220 E8=E8+C5: IF E8 = 0 RETURN
2230 LPRINT "A/C #33 TELEPHONE EXPENSE"?TAB(40)USING A$
     ;E8:RETURN
2240 E9=E9+C5: IF E9 = 0 RETURN
2250 LPRINT "A/C #34 POSTAGE";TAB(40)USING A$;E9:RETURN
2260 F1=F1+C5: IF F1 = 0 RETURN
2270 LPRINT "A/C #35 INTEREST PAID";TAB(40)USING A$;F1:
     RETURN
2280 F2=F2+C5: IF F2 = 0 RETURN
2290 LPRINT "A/C #36 DUES & SUBSCRIPTIONS";TAB(40)USING
      A$;F2:RETURN
2300 F3=F3+C5: IF F3 = 0 RETURN
2310 LPRINT "A/C #37 INSURANCE";TAB(40)USING A$;F3:RETU
     RN
2320 F4=F4+C5: IF F4 = 0 RETURN
2330 LPRINT "A/C #38 DEPRECIATION";TAB(40)USING A$;F4:R
     ETURN
2340 F5=F5+C5: IF F5 = 0 RETURN
2350 LPRINT "A/C #39 MAINTENANCE,BUILDING";TAB(40)USING
      A$;F5:RETURN
2360 F6=F6+C5: IF F6 = 0 RETURN
2370 LPRINT "A/C #40 UTILITIES,GAS,ELEC.,WATER";TAB(40)
     USING A$;F6:RETURN
2380 F7=F7+C5: IF F7 = 0 RETURN
2390 LPRINT "A/C #41 OFFICE EXPENSE";TAB(40)USING A$;F7
     :RETURN
2400 F8=F8+C5: IF F8 = 0 RETURN
2410 LPRINT "A/C #42 SERV. EQUIP. MAINTENANCE";TAB(40)U
     SING A$;F8:RETURN
2420 F9=F9+C5: IF F9 = 0 RETURN
2430 LPRINT "A/C #43 PERSONAL PROPERTY TAX";TAB(40) USI
     NG A$;F9:RETURN
2440 G1=G1+C5: IF G1 = 0 RETURN
2450 LPRINT "A/C #44 WORKMEN'S COMPENSATION";TAB(40)USI
     NG A$;G1:RETURN
2460 G2=G2+C5: IF G2 = 0 RETURN
2470 LPRINT "A/C #45 SHIPPING EXPENSE";TAB(40)USING A$;
     G2:RETURN
2480 G3=G3+C5: IF G3 = 0 RETURN
2490 LPRINT "A/C #46 SERVICE EXPENSE";TAB(40)USING A$;G
     3:RETURN
2500 G4=G4+C5: IF G4 = 0 RETURN
2510 LPRINT "A/C #47 MISCL. EXPENSE"; TAB(40)USING A$;G
     4:RETURN
2520 H1=H1+C5: IF H1 = 0 RETURN
2530 LPRINT "A/C #48 JENKINS WITHDRAWALS";TAB(40)USING
     A$;H1:RETURN
2540 H2=H2+C5: IF H2 = 0 RETURN
2550 LPRINT "A/C #49 PROFIT AND LOSS";TAB(40)USING A$;H
     2:RETURN
2560 END
3000 INPUT #-1,A1,A2,A3,A4,A5,A6,A7,A8,A9,B1,B2,B3
3010 INPUT #-1,L1,L2,L3,L4,L5,L6,L7,L8,I1,I2,I3,I4,I5
3020 INPUT #-1,E1,E2,E3,E4,E5,E6,E7,E8,E9,F1,F2,F3
3030 INPUT #-1,F4,F5,F6,F7,F8,F9,G1,G2,G3,G4,H1,H2
3040 RETURN
4000 PRINT #-1,A1,A2,A3,A4,A5,A6,A7,A8,A9,B1,B2,B3
4010 PRINT #-1,L1,L2,L3,L4,L5,L6,L7,L8,I1,I2,I3,I4,I5
4020 PRINT #-1,E1,E2,E3,E4,E5,E6,E7,E8,E9,F1,F2,F3
4030 PRINT #-1,F4,F5,F6,F7,F8,F9,G1,G2,G3,G4,H1,H2
4040 RETURN
```

Shake hands with this $59.95 alternative to an E/I.

LNW Research Expansion Board

Ian Hodgson
296 Malcolm Circle
Dorval, Quebec,
Canada H9S 1T7

While the Level II, 16K TRS-80 is a lot of machine for not much money, it becomes evident that for serious work more memory and disk drives are essential.

The first equipment I investigated was the RS expansion interface. At $837 for the interface with 32K RAM, $160 for the RS232 board, and $45 for the communications software it looked like a grand total of $1042 just to connect the drives and add a bit of memory. There had to be a cheaper way.

I then saw the ad from LNW Research for their TRS-80 expansion board. I ordered one for $59.95: it is a 9½ × 11½ inch, double-sided, soldermasked, glass epoxy board.

The board has provisions for:

- up to 32K extra memory
- floppy disk controller
- parallel printer port
- dual cassette port
- real time clock
- RS232 port (can also be used for 20mA loop)
- onboard power supply
- bus for expansion

All features are optional. The manual includes itemized parts, lists for each option, enabling you to build only the sections you want.

I chose everything except the power supply and the dual cassette port, so the first job was to look for the necessary parts. Although the manual does mention parts substitutions, they don't cover the UART, a Western Digital TR1602B. I have used both an AY-5-1013A and an AY-3-1014A with fine results. One word of warning: If you haven't bought any 74LS IC's lately, be prepared for a shock when you see the current prices. I kept the bill to less than $280, including memory, so the entire expansion was built for about $370 Canadian—a lot less than $1042 for the RS interface.

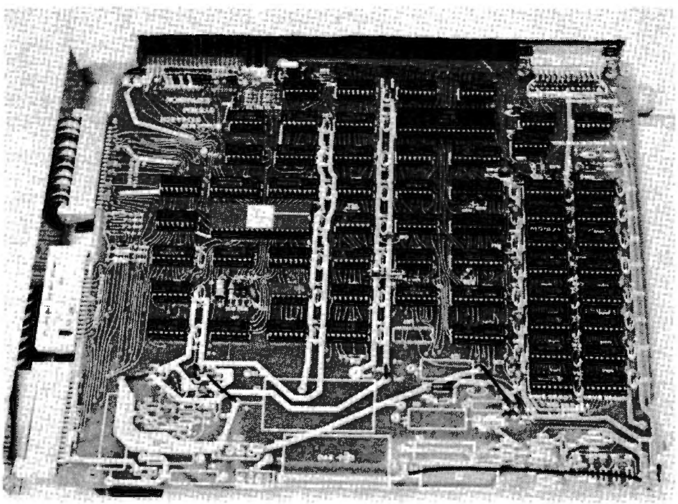
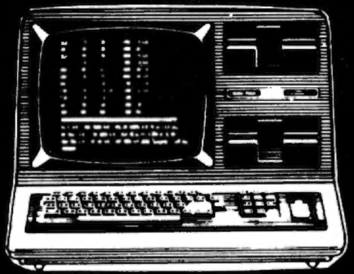


Photo 1. The expansion board installed behind the TRS-80. At left, from front to back, are the disk drive connector, parallel printer port, and keyboard connector. At rear, left, is the expansion connector, and at right, the RS232 connector. The two 40-pin chips are the UART and the disk controller. The vacant area in front is for the power supply.

The RS232 port is well provided with handshaking, and Request to Send (RTS), Clear to Send (CTS), Data Terminal Ready (DTR), Data Set Ready (DSR), Carrier Detect (CD), and Ring Indicator (RI) are all implemented, as well as provision for generating a true BREAK signal. In addition, the serial data appears at one bit (IN OE8H, bit 1) and the port may be wired for 20mA operation if desired. Although the TRS-80 may be set up either as a data terminal (DTE) or data communication device (DCE), the handshaking lines are fully implemented only for DTE. There is no provision for



the UART to generate interrupts, nor is the interrupt line easily available to those who have time-critical applications. Interrupts are definitely usable on the TRS-80 (from your own machine language programs), since the vectors are stored in RAM. You must be careful, though, because many of the ROM routines depend on the interrupt vectors, and you may have to write your own keyboard and screen drivers depending on what interrupt mode you use.

Although a power supply is built in on the board (and it is quite a sophisticated one, too, using an additional Radio Shack TRS-80 transformer) I used one I already had, to save money and because my RS store refused to order the transformer for me. The completed board works with an external supply. It requires +5 volts at about 1A, +12V at a few mA, and −12V also at a few mA.

The manual is well written, and details are given for construction, configuring the system and theory of operation. Data sheets for the UART and the disk controller (1771) are included with software for using the 80 as a terminal. Software is also provided to drive a serial printer through the RS232 port, or through the parallel port, with certain board modifications. I only found one error in the chart of UART control addresses; the bits for Request to Send and for Data Terminal Ready are interchanged.

Construction went fairly well, but there were a few bugs that took several hours to find. Amazingly, these were all traced to a batch of new 100 ohm resistors, of which three were open.

One feature I didn't like was the vertically-mounted terminating resistors at the expansion bus.

Finally, everything worked, including the TRSDOS 2.3 memory test (TEST1) and stress test (TEST2).

Words of Warning

A few words of warning are in order. The bus expansion provided should be used with due caution. It is a simple, unbuffered extension of the TRS bus, and any expansion will require more buffering with high impedance input buffers (not 7400 IC's). Remove the terminating resistors from the LNW board and terminate at the end of your expansion, as well. There are two edge connectors on the board. Either should work with the keyboard, and the other may be used as the bus expansion. However, when I used the connector next to the terminating resistors for the keyboard, I had unusual problems like random rebooting, and incorrect operation of a few programs. Using the other connector cured the problem. I also suggest that you use the shortest cable possible. Incidentally, the cable does not come with the board, but you can make one with 40-conductor ribbon cable. The identical connectors on each end are 3M #3464-000. They are installed using a small vice.

Using disks, machine language programs can be patched to correct errors, and reassembled with ease. Data files are child's play, and of course, long programs load in seconds. My system has a KIM-1 converting from parallel to serial on the parallel printer port allowing me to use the KIM to format my output (paging, line length, margins, etc.). What is more, it has 4K of memory available to act as a buffer. When I LLIST a program less than 4K long, the READY returns in about one second!

I am using a Terminet 300 printer (upper and lowercase) at 300 baud. I use the RS232 port for packet radio transmission and reception at 2400 baud.

To summarize: the LNW Research board is well laid out and inexpensive. The documentation is satisfactory. With disk drives, 48K of memory and Electric Pencil with lowercase modification, it makes a good machine. ■

Verbatim Disks for TRS 80** at Lowest Prices with Free Hub Rings & Plastic Case Too!

Compare this offer on mini disks ... then buy now and save!

(10 Pack — 5¼" Disks)

	Generic Disks	Others	Verbatim MD 525-01
Disks	$21.95	$24.95	$27.95
Hub Ring	.60	.60	Free
Case	3.95	3.95	Free
	$26.50	$29.50	$27.95

The quality facts about Verbatim Disks

- Certified 100% ERROR FREE
- It's the diskette preferred by *Apple and only one of two Apple-approved
- Factory-installed hub rings means ... no loss of data due to damaged hub holes or loss of coating when hub hits disk

*A trademark of Apple Computer Inc

**A trademark of Tandy Corporation

LT-DATA INC. ✓59

Here's our Verbatim Offer —

MD 525-01 5¼" soft sectored MINI DISKS (includes factory-installed hub ring and a plastic storage case)

10 DISK Pack $27.95 ea.

Bonus — order 9 and get 10th for 99¢!

FD 34-8000 8" soft sectored DISKS (includes factory-installed hub ring in a hard box)

10 DISK Pack $43.95 ea.

Bonus — order 9 and get 10th for $14.95!

Add $2.50 per 10 Disk Pack for shipping and handling

— NOTE: Any excess shipping will be refunded

We accept check, money order, Visa or Master Charge. (With credit card orders, include card number and expiration date.)

C.O.D. shipments, $2.00 additional. Direct mail orders to LT-DATA INC. at either address below.

24 Hour Phone Order Line 1-303-575-8518

P.O. Box 1567, Wheat Ridge, Colorado 80033
Colorado Residents add 3% Sales Tax
P.O. Box 1017, Medina, Ohio 44258
Ohio Residents add 5½% Sales Tax

Final Opportunity . . . limited quantity of TDK cassettes still available (D-C60 & D-C90) See our April ad.

Push and pop with blocks of BASIC code.

The Encoder

Ron Cain
Route 1, Box 190
Chancellor, AL 36316

You just finished writing a short asssembly language program for that flashy graphics display you want in your latest game program. Now, you have to convert all those machine language instructions in hex to decimal, and build them into data statements. You can then let your BASIC program poke the machine code into memory when it is run. By doing this, you eliminate the need to load both the machine language code and BASIC program in two separate steps. Good idea. But, what a job!

The Easy Way

Here is an easier way. Let the computer convert the hex machine instructions into a block of BASIC code which will POKE the hex codes back into memory at the proper addresses.

Program Encoder is written in BASIC, requires a minimum 16K, one disk system, and will generate the BASIC code (instructions and data statements) to POKE your machine code into memory.

Here's how it works. First, load your machine code to its proper location in reserved high memory. Then run Encoder. You will first be asked for the starting and ending addresses in memory where it can find the machine code.

These addresses can be in decimal (with or without a trailing D), or in hexadecimal (which must include the trailing H). Encoder then asks for the starting line number and increment to use in the BASIC code it produces.

Finally, you are asked for a filespec to store the code produced. The program then proceeds to create and write to the disk file a block of BASIC code to reproduce the machine instructions. Note that the code is stored on disk as if it had been SAVEd using the ASCII option, so that it can be merged with other BASIC programs.

Also, the code generated is in the form of a subroutine to the program, which is merged (since it contains the RETURN instruction). Take out the RETURN, and you can execute the routine in line.

So, next time you want to build a machine language program into your BASIC program, let Encoder do the work while you just do the thinking.■

```
1 '******************************************************
     ******
2 '*    PROGRAM NAME - ENCODER         VERSION 1.0
     *
3 '*      WRITTEN BY - R. L. CAIN
     *
4 '*                   BOOTSTRAP COMPUTER SERVICES
     *
5 '*                   CHANCELLOR,  ALABAMA  36316
     *
6 '*                   (205) 347-5206
     *
7 '******************************************************
     ******
100 CLS: CLEAR 500: DEFINT A-Z
110 PRINT @13,"MACHINE LANGUAGE TO BASIC ENCODER": PRIN
     T: PRINT
120 '*
130 '* GET THE NECESSARY INPUT FOR THIS RUN *
140 '*--------------------------------------*
150 INPUT "STARTING ADDRESS"; A$
160 GOSUB 30050
170 IF ER=1 THEN 150
180 SA=A: SA!=A!
190 '*
200 INPUT "  ENDING ADDRESS"; A$
210 GOSUB 30050
220 IF ER=1 THEN 200
230 EA=A: EA!=A!
240 '*
250 IF EA!>SA! THEN 290
260 PRINT :PRINT "* END ADDRESS NOT > START ADDRESS *":
      PRINT
```

Program continues

```
270 GOTO 150
280 '*
290 PRINT: INPUT "STARTING LINE NUMBER"; SL
300 INPUT "      LINE INCREMENT"; LI
310 '*
320 PRINT: LINE INPUT "FILESPEC FOR OUTPUT? "; FS$
1000 '*
1010 '*  OPEN THE FILE FOR OUTPUT, AND WRITE THE CODING TO POKE
1020 '*  THE MACHINE INSTRUCTIONS BACK INTO MEMORY.
1030 '*-------------------------------------------------------*
1040 OPEN "O",1,FS$
1050 GOSUB 40050
1060 LN$=LN$ + "ZA% = " + STR$(SA)
1070 GOSUB 50040: GOSUB 40050
1080 LN$=LN$ + "FOR ZI! = " + STR$(SA!) + " TO " + STR$(EA!)
1090 GOSUB 50040: GOSUB 40050
1100 LN$=LN$ + "READ ZI%: POKE ZA%,ZI%"
1110 LN$=LN$+": IF ZA%=32767 THEN ZA%=-32768 ELSE ZA%=ZA%+1"
1120 GOSUB 50040: GOSUB 40050
1130 LN$=LN$ + "NEXT ZI!: RETURN"
1140 GOSUB 50040: GOSUB 40050
2000 '*
2010 '*  WE SET UP A FOR/NEXT LOOP TO PEEK EACH BYTE FROM
2020 '*  MEMORY AND BUILD THEM AS DECIMAL INTEGERS INTO
2030 '*  DATA STATEMENTS.
2040 '*---------------------------------------------------*
2050 LN$=LN$+"DATA "
2060 FOR SA!=SA! TO EA!
2070 IF LEN(LN$) < 60 THEN 2110
2080 LN$=LEFT$(LN$,LEN(LN$)-1)
2090 GOSUB 50040: GOSUB 40050
2100 LN$=LN$+"DATA "
2110 A$=STR$(PEEK(SA)): A$=RIGHT$(A$,LEN(A$)-1)
2120 LN$=LN$+A$+","
2130 IF SA=32767 THEN SA=-32768 ELSE SA=SA+1
2140 NEXT SA!
2150 LN$=LEFT$(LN$,LEN(LN$)-1)
2160 GOSUB 50040
2170 CLOSE
2180 END
30000 '*
30010 '* THIS SUBROUTINE CONVERTS A NUMBER IN A$ TO A DECIMAL
30020 '* VALUE. A$ CAN CONTAIN EITHER A HEX NUMBER (ENDS IN H)
30030 '* OR DECIMAL NUMBER (ENDS IN D OR A NUMBER).
30040 '*-------------------------------------------------------*
30050 ER=0
30060 IF LEN(A$)=0 THEN ER=1: RETURN
30070 IF RIGHT$(A$,1)<>"D" THEN 30090
30080 A$=LEFT$(A$,LEN(A$)-1): GOTO 30100
30090 IF RIGHT$(A$,1)="H" THEN 30160
30100 FOR I=1 TO LEN(A$)
30110 K=ASC(MID$(A$,I,1))
30120 IF K<48 OR K>57 THEN ER=1: RETURN
30130 NEXT I
30140 A!=VAL(A$): IF A!>32767 THEN A=A!-65536 ELSE A=A!
30150 RETURN
30160 J!=1: A!=0
30170 FOR I=LEN(A$)-1 TO 1 STEP -1
30180 K = ASC(MID$(A$,I,1))
30190 IF K>47 AND K<58 THEN K=K-48: GOTO 30220
30200 IF K>64 AND K<71 THEN K=K-55: GOTO 30220
30210 ER=1: RETURN
30220 A!=A!+(J!*K)
30230 J!=J!*16
30240 NEXT I
30250 IF A!>32767 THEN A=A!-65536 ELSE A=A!
30260 RETURN
40000 '*
40010 '*  THIS SUBROUTINE INITIALIZES EACH LINE OF CODE TO BE
40020 '*  OUTPUT WITH ITS LINE NUMBER MAKING IT READY FOR THE
40030 '*  ADDITION OF THE BASIC INSTRUCTIONS OR DATA.
40040 '*-------------------------------------------------------*
40050 PRINT "CREATING LINE";SL,
40060 LN$=RIGHT$(STR$(SL),LEN(STR$(SL))-1) + " "
40070 RETURN
50000 '*
50010 '*  THIS SUBROUTINE WRITES EACH LINE OF CODE CREATED TO
50020 '*  THE DISK FILE, AND THEN INCREMENTS THE LINE NUMBER.
50030 '*-------------------------------------------------------*
50040 PRINT "WRITING LINE";SL: PRINT#1, LN$
50050 SL=SL+LI
50060 RETURN
```

Program Listing for Encoder

Ride herd on your cassette voltage levels and interface with the world.

The Pulse Jockey

Larry Suter
1643 Warsaw Ave.
Livermore, CA 94550

How would you like to have a cheap and simple way to hook your TRS-80 to the outside world? A way that uses software and trickery to side-step complicated hardware? One that doesn't involve ribbon cables, expansion connections, p.c. boards or expensive chips?

For $5 and an evening's effort you can take this hardware/software route to interfacing.

We'll use serial I/O through the cassette port and some simple hardware for our interface. Because the hardware is so simple, we'll have to augment it with short algorithms written in Level II BASIC. For faster I/O we'll use machine language algorithms. But this can be done painlessly, and I'll show you how.

Essential Point

The key to this approach is that you can get two independent sources of pulses from your cassette output plug. Let's first see how the cassette output actually produces three software selectable output voltages. Then we'll understand how these three voltage levels can be used to generate the two independent channels of pulses.

Let's experiment. Hook a voltmeter across the cassette output plug (the grey one which goes into AUX). Run the program:

```
10 INPUT N:OUT 255, N:GOTO 10
```

When you input N as O you'll measure about 0.45-V output. What you've done with this program is to send a 0 to port 255. (255 is the cassette port's i.d. number.) Sending a 0 to port 255 results in 0.45-V output.

Next, input N as 1. You'll find that sending one to port 255 results in 0.9V output.

Finally, input a two. Sending two to port 255 results in 0V output. You'll see that your TRS-80 actually puts out three voltage levels—0.45V, 0.9V or 0V, depending on whether you send a 0, 1 or 2 port 255. These three software selectable voltage levels provide the key to interfacing through the cassette port. They can be used to generate two channels of output pulses.

Fig. 1 shows how you get two output channels from three voltage levels. Normally the cassette output level is 0.45V. When you issue the command: OUT 225, 1:OUT 255, 0, the output rises to 0.9V and drops back to 0.45V. You can generate a logic compatible pulse by connecting the cassette output plug to a comparitor, which produces a one when its input is greater than 0.7V. We'll call the output of this comparitor Pulse 1 since it's actuated by sending a one to port 255.

If you send out an OUT 255,2:OUT 255,0 you won't get PULSE 1 since the voltage is always less than the comparitor's threshold. But, you will get a pulse from a comparitor which has a threshold of 0.35V. We'll call this second output Pulse 2. That's how you get two channels out of one plug. Three voltage levels and two comparitors.

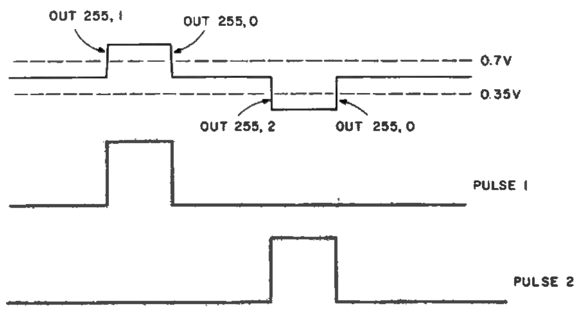


Fig. 1. Your cassette output plug produces three software controllable voltage levels. Two comparitors will convert these levels into two independent sources of logic pulses, Pulse 1 and Pulse 2.

1- 10K resistor
2- 5K resistors
2- 1K resistors
3- 1N34A Germanium diodes
1- 0.01 microfarad capacitor
2- 0.047 microfarad capacitors
1- 10. microfarad electrolytic capacitor
1- 339 quad comparitor
1- 74154 1:16 multiplexer
1- 7493 counter
1- 74151 8:1 demultiplexer
2- miniature phone jacks
1- bread board

In addition to these parts you'll want chips such as 74157 quad latches, more 7493 counters and 7404 inverters, to let you make parallel output ports.

Parts List

By happenstance, the TRS-80 cassette plug can actually give two channels of pulses. What are the benefits of two pulse sources? Two pulse sources allow you to use one pulse to route the other pulse wherever you want it.

Fig. 2 shows how this is done. Pulse 1 goes into a counter. The output of the counter is connected to the address pins of a multiplexer chip. What this chip does is to route its one input signal (Pulse 2!) to any of its output pins. Which output is hooked up the Pulse 2 is determined by the four address pins connected to the counter.

For example, let's say that you want to send out 10 pulses to pin 7. First you issue seven Pulse 1s which cause the counter to put out a 0111 bit pattern. This pattern causes output pin 7 of the multiplexer to be connected to Pulse 2. Now send out 10 Pulse 2s and they will end up where you want them.

What if you want parallel output? Fig. 2 also indicates how you can do this with a separate counter. One of the multiplexer's outputs is connected to the reset pin on the counter, and another to its clock input. By first clearing the counter and then sending out N pulses you can get any output. You can get many parallel outputs by connecting latches to the counter's output and using other pulses from the multiplexer to latch the data.

Is it possible to do half as much with just one pulse source? Possibly, but it would require pulse length modulation or some other tricks. It's a lot easier to do it this way.

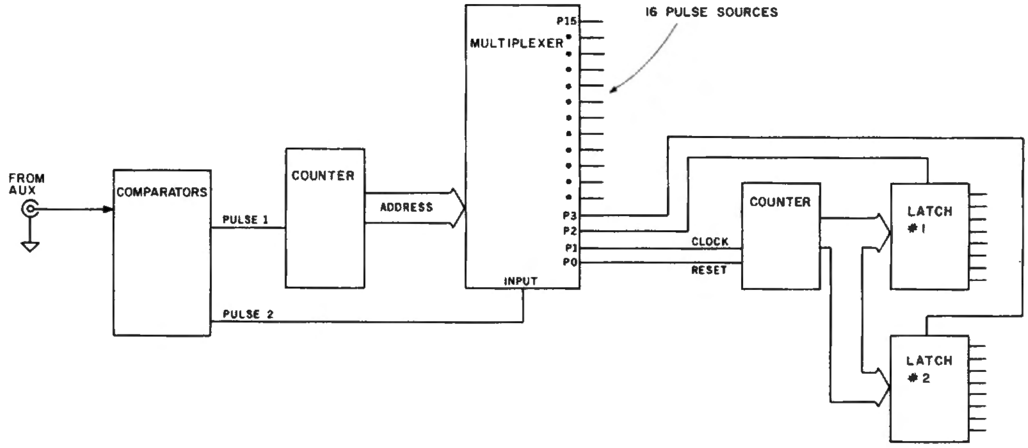


Fig. 2. Pulse 1 and Pulse 2 can generate as many independent pulses as you need. The idea is that Pulse 1 routes Pulse 2 wherever you want. You get steady, parallel outputs by using the pulses to drive counters and latches.

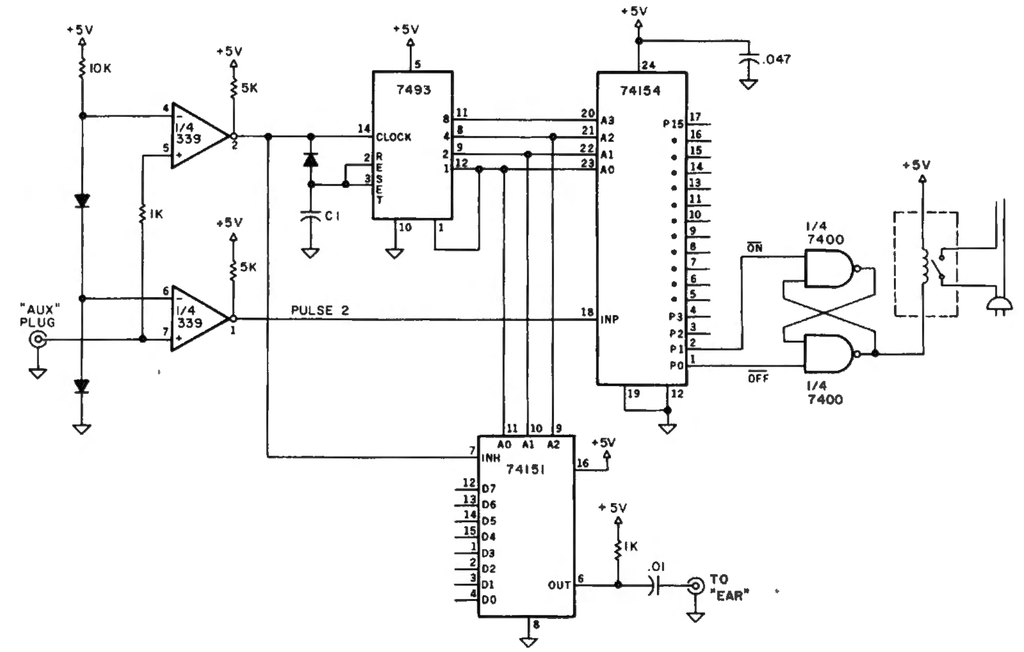


Fig. 3. Circuit for the I/O port. The upper three chips form the output port. Adding the 74151 will give you input as well. The port is shown driving a relay via a flip-flop.

Hardware

The circuit I use to do this kind of pulse jockeying is shown in Fig. 3. The 339 comparitors, which are the heart of this scheme, must be properly biased. One at 0.35V and the other at 0.7V. Use germanium diodes to do this since they provide a voltage drop of 0.3 to 0.5V, depending on their current. The way the inputs are hooked up, Pulse 1 (the router) is normally low. Pulse 2 (the routee) is normally high.

Since the 7493 counter is hooked up to Pulse 1, it gets incremented every time you send out an OUT 255,1:OUT 255,0. By sending out the right number of pulses you can hook Pulse 2 up to any one of the multiplexer's outputs.

If you want to hook up pin 7 to Pulse 2, then you can do it by sending out seven Pulse 1s—if the counter was zero when you started. To make sure you've started at zero, the circuit of Fig.

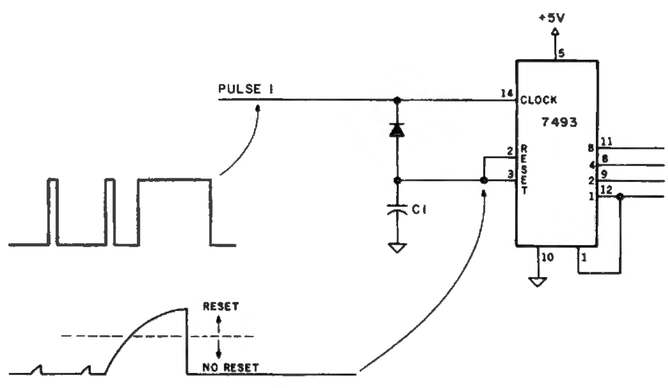


Fig. 4. The diode and capacitor allow you to zero the 7493 counter with a long Pulse 1s and increase it with short Pulse 1s.

✓See List of Advertisers on page 306

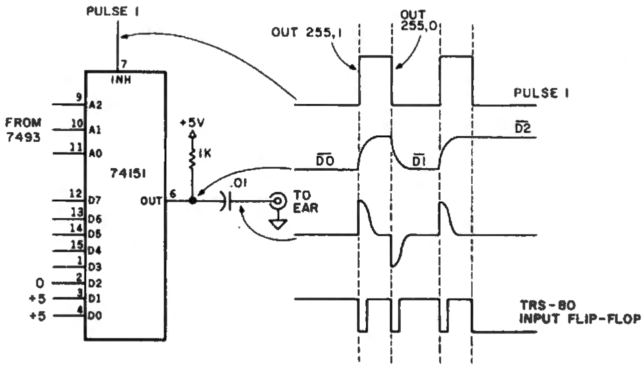


Fig. 5. Every Pulse 1 initiates a chain of events which loads a data bit into your TRS-80's input flip-flop.

```
              CALL OA7F         M of USR(M)9HL
              LD A,O1H
              OUT FFH           Clear 7498
              XOR A             with long
              LD B,A            PULSE1
              DJNZ, FE
              OUT FF

              BIT 7H
                                IF HL<O GOTO INPUT
              JR NZ, INPUT

              LD DO1H
              LD E,H            Send out H PULSE1's
              CALL PULSE

              LD D,O2
              LD E,L            Send out L PULSE2's
              CALL PULSE

              RET

PULSE         INC E
LOOP          DEC E
              RET Z             Send pulses to cassette port
              LD A,D            D=type (1 or 2)
              OUT FFH           E=how many
              XOR A
              OUT FF
              JR, LOOP

INPUT         LD C,A            O9C
              LD B,O8H          B=# of bits to input
LOOP2         CALL DELAY
              SRL C             C=2*C
              IN FF
              AND 80H           IF INP(225)>127 0 THEN C=C+128
              ADD C             ELSE C=C+O
              LD C, A
              LD A,01H
              OUT FFH
              XOR A             Send out PULSE1
              CALL DELAY
              OUT FFH
              DJNZ, LOOP2
              LD, L, C
                                C9HL
              LD H,OOH
                                HL9USR(N)
              JP OA9AH
DELAY RET
```

Table 1. Assembly language listing of the machine language I/O program you load with Program Listing 3.

3 will let you zero the counter by sending a long Pulse 1. Before looking at Fig. 4 which explains how this zeroing is done, you'll

need one essential fact. That fact is that when a TTL input is low it has about 1ma of current trickling out of it.

To zero this counter with a long Pulse 1 you can use the instruction:

OUT 244, 1:FOR I = 1TO20:NEXT:OUT 255,0

Program Listing 1 is the BASIC listing of a subroutine for sending out N pulses to pin P. First we zero the counter. Then we send P Pulse 1s followed by N Pulse 2s. That's all there is to it!

C1 of Fig. 3 & 4 must be 10 microfarads when using this BASIC algorithm.

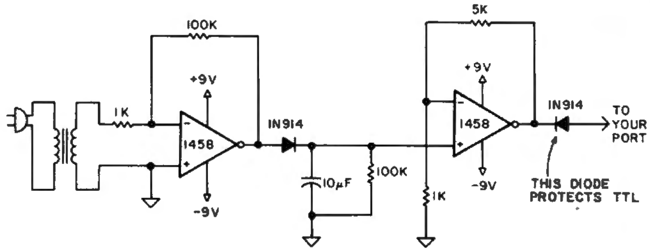


Fig. 6. This handy transducer tells whether an appliance is on or off. AC current will induce a signal in the pick-up loop which gets amplified and rectified to a TTL compatible level. The output is connected to one of the input pins of your port.

```
10 FOR I=0 TO 73:READ A:POKE 20224+I,A:NEXT
20 POKE 16526,0:POKE 16527,79
25 DATA 205,127,10,62,1,211,255,175,71,16,254,211,255,2
     03,124,32,24,22,1,92,205,30,79,22,2,93,205,30,79,2
     01,28,29,200,122,211,255,175,211,255,24,246,0079,6
     ,8,205,73,79
26 DATA 203,57,219,255,230,128,129,0079,62,1,211,255,20
     5,73,79,175,211,255,16,233,105,38,0,195,154,10,201
```

Program Listing 1. This BASIC algorithm sends output pulses to pin P of the 74154 of Fig. 3. C1 of Figure 3 must be 10 microfarads when using this algorithm.

```
2990 REM CLEAR 7493 WITH A LONG PULSE1
3000 OUT 255,1:FOR I=1 TO 30:NEXT:OUT 255,0
3005 REM SELECT OUTPUT PIN WITH P PULSE1'S
3010 IF P>0 THEN FOR I=1 TO P:OUT 255,1:OUT 255,0:NEXT
3015 REM SEND N PULSES TO SELECTED OUTPUT PIN
3020 IF N>O THEN FOR I=1 TO N:OUT 255,2:OUT 255,0:NEXT
3030 RETURN
```

Program Listing 2. When you call this BASIC subroutine, you'll get back C, the decimal equivalent of the input to the 74151 chip.

```
1990 REM CLEAR 7493 WITH A LONG PULSE1
2000 OUT 255,1:FOR I=1 TO 30:NEXT:OUT8255,0
2005 REM READ IN D0-D7. CONVERT THIS BIT PATTERN TO DEC
     IMAL              NUMBER C.
2010 C=0:FOR I=0 TO 7:IF INP(255)>127 THEN C=C+2[I
2020 OUT 255,1:OUT 255,0:NEXT:RETURN
```

Program Listing 3.

Machine Language Algorithm

Program Listing 3 is a small Level II BASIC routine which will load a machine language form of the output, add input algorithims into memory. It will also set up the appropriate TRS-80 pointers so you jump to the routine whenever you issue an X = USR(N) command. The machine language program is contained in the DATA statement so be sure to get it right.

Table 1 is an annotated assembly language listing of the machine language I/O algorithm. One thing needs explaining. The calls to subroutine DELAY when doing input are needed to chew up about 15 microseconds. Apparently the TRS-80 cassette input amplifier isn't fast enough to respond to the very quick pulses which would result without the delay.

Great for Beginners

If you have little or no experience with digital electronics then this is the project for you. You can put the circuit of Fig. 3 together one chip at a time. Every chip is a milestone. An individual accomplishment.

First get the comparitor working to generate Pulse 1 and Pulse 2. Use a voltmeter to diagnose the circuit.

Then hook Pulse 1 up to the 7493 counter and practice incrementing and zeroing it. You can also check this action with your voltmeter.

Next comes the 74154 multiplexer. But don't stop your digital education here. Hook up counters, inverters and latches to the output pulses. Once you have parallel outputs your computer and output port form the basis of a digital lab. You can see this system to study the operation of just about any digital chip. And they're not hard to set up on a breadboard.

Where can you get more information about digital electronics? Well one good source is Lancaster's TTL Cookbook. It's full of information about chips and techniques. And is presented in a way that allows you to skip around. ■

✓See List of Advertisers on page 306

REVIEW

Photo 1

Shows you how to use two free video pins.

>22MHz For $99

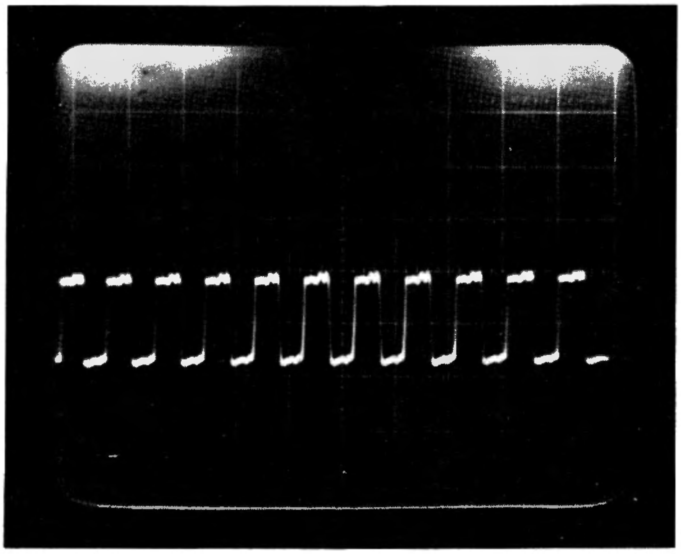


Photo 2

Fredi Egger
Grundstrasse 21
CH-8500 Frauenfeld
Switzerland

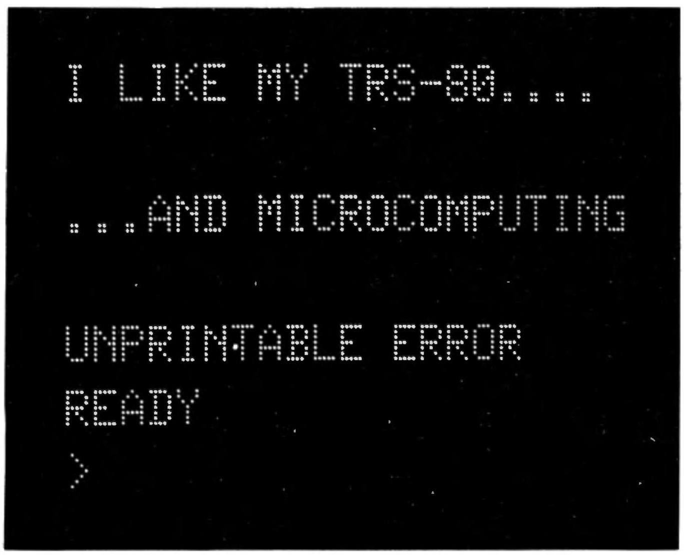


Photo 3

Everyone owning a TRS-80 seems to agree that the cassette recorder is the weak point of the whole line. A lot of articles have been published in order to help people to get rid of these problems. I personally consider the Data Dubber, published in Jan. and Feb., 1980 of *80 Microcomputing*, to be the most reliable cure. There is nothing to be added—it just works perfectly.

In my opinion the TRS-80 monitor takes second place for being problematic. I tried hard to improve the display. I followed the hints in *Kilobaud Microcomputing* (Dec., 1979, "Peak Your TRS-80 Display") and *80 Microcomputing* (Feb., 1980, "Video Tune-Up"). But I was not quite satisfied.

Finally I decided to go for another monitor—without spending much money. I looked through the relevant magazines and discovered an ad announcing a monitor with a bandwidth of less than 22 MHz for only $99 from Electravalue Industrial.

The ASL-monitor came in perfect condition and was accompanied by fairly good documentation. They even enclosed an extra sheet with suggestions on how to connect it to the TRS-80. Since the monitor does not have a mixed video input one has to bring the horizontal and the vertical sync signals out of the TRS-80. TRS-80 designers left two of the five pins of the video-DIN connector free.

Following the instructions supplied with the monitor, the horizontal sync pulses are taken from Z6, pin 6, and brought to the DIN connector, pin 2. Similarly, the vertical syncs from Z57, pin 8, go to DIN, pin 3 in Fig. 1.

There is one problem: Due to a phase jitter between the horizontal and vertical sync pulses produced by the TRS-80, the displayed dots jitter randomly.

The jitter disappears by adjusting the horizontal position potentiometer (R20 in Fig. 1) to move the display to the right side of the screen. But you will be able to use only the right half of the screen.

Fortunately, there is a simple way to fix this. Instead of pro-

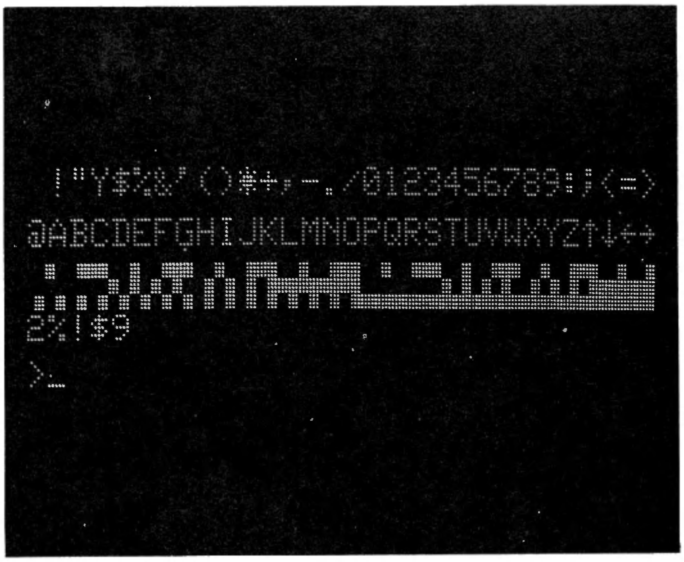


Photo 4

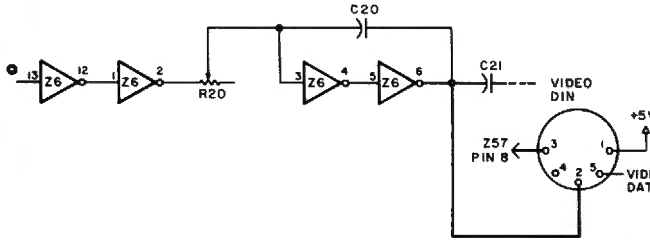


Fig. 1

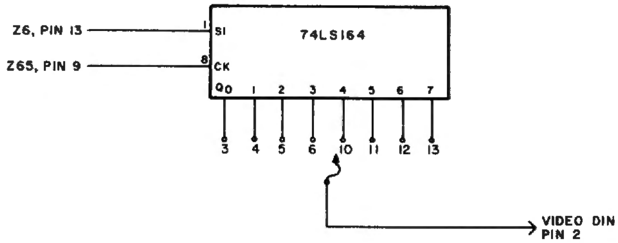


Fig. 2

ducing the horizontal sync delay with a CR network (C20 and R20 in Fig. 1), use a shift register as shown in Fig. 2. The 74LS164 is piggybacked onto Z7 (taking +5 volts from pin 14 and ground from pin 7). All the other pins are bent outward. Connect the serial input to Z6, pin 13, and the clock input to Z65, pin 9. At this point the frequency is 443.52 KHz which gives us a clock width of 2.25 microseconds.

Then digitally adjust the horizontal position at the beginning of the character lines following Table 1. I found pin 10 the most convenient. All other pins of the 164 are unused.

A bandwidth of 22 MHz gives us a chance for more improvements. Consider the letter Z in the MEMORY SIZE question. You will notice that the top and bottom bar appear much brighter on the screen than the single dots forming the diagonal line. When there is more than one dot in a row of the shift registers (Z10 and Z11), the display will be brighter than the single dots. Look at the E and Z in Fig. 3.

There is an easy way to get rid of that. In the TRS-80 the parallel-serial-conversion of character and graphics slices takes place in two shift registers (Z10 and Z11). The serial outputs of these ICs are NORed in Z30 (pins 2 and 3). At the output of this gate (point A in Fig. 3) a logical 0 means a white spot on the video screen. Correspondingly, a 1 creates a screen blank.

If we now insert a gate (one of the spare 74LS32 gates in the TRS-80) between Z30, pin 1, Z41, pins 6 and 7 (as shown in Fig. 3), we can control the video output by applying an appropriate signal to the second input of this gate. At Z9, pin 8, we find the clock signal for the two shift registers in Photo 2. Connect this to the inserted gate, and you get the display shown in Photo 3.

In fact, I used one-half of a 74LS123-monoflop. It seemed to give better results than a simple OR gate. The 123 was piggybacked onto Z10 and wired up as shown in Fig. 4. In this application we are using the two bottom lines of the monoflop's truth table. The monoflop is initially fired by the low-going edge of the signal coming from the shift registers. About 50 nanoseconds later, it is reset by the low-going clock signal. If the shift register signal stays low, the astable will be fired again by the high-going edge of the clock signal (another 50 nanoseconds later).

We can paint 73720 (64 characters * 6 dots * 16 rows * 12 scanlines) distinct points in a 384*192 matrix on our screen. If we only had the hardware (memory) and the software to control them. . . But don't you think it is a promising start to high resolution graphics?■

Hor Sync Output	Line Locations
pin 3	about center of screen
4	↓
5	moving toward the left
6	in half-inch steps
	(2.25 microseconds/step)
10	↓
11	
12	
13	leftmost position

Table 1. If the horizontal sync output is connected to a pin designated in the left column, the position beginning the line is indicated in the right column.

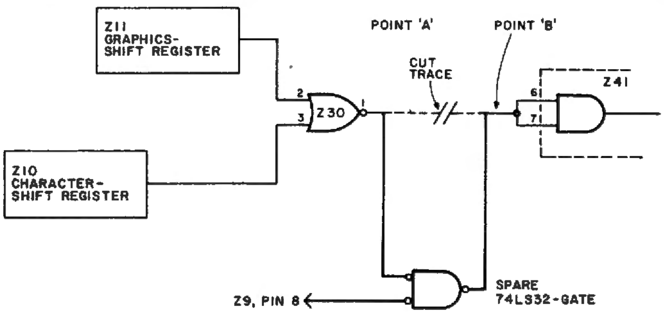


Fig. 3

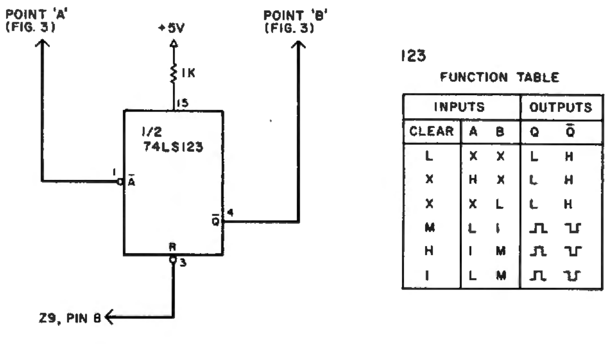


Fig. 4

A calculator based utility that takes the hex off number base conversions.

Hex Converter

Richard H. Malone
82 Garden Drive
Albertson, NY 11507

One of the most difficult prerequisites of assembly language programming is learning the hexadecimal number system. If you're like me, accustomed to BASIC where all references to memory locations and constants are in decimal, your first exposure to machine language programming can be a nightmare of hex to decimal/decimal to hex conversion tables and reams of scribbled notes.

Of course many conversion programs are available to you, but when you're deep into a machine language problem, your computer is tied up and you must resort to good old paper and pencil.

A better solution is to perform the necessary calculations on your hand calculator or a programmable one.

Number Systems

Let's review the relationship between the decimal and hexadecimal number systems.

The basis of the hexadecimal system is easy to understand. Each character position in a number has a value determined by two factors:

- the value of the character occupying the position;
- the value of the particular position.

These two factors are multiplied to determine the value for that position and the value for all positions is added to determine the value of the entire number.

In the decimal system there are ten possible characters to occupy each position (0–9). In the hexadecimal system there are sixteen possible characters to occupy each position (0–9 and A–F).

In any number system the value of a particular position is equal to the base of the system raised to a power equal to the number of positions to the left of the decimal point, minus one. Thus in the decimal system (base 10), the value of each position to the left of the decimal point is:

10^4	10^3	10^2	10^1	10^0
10000	1000	100	10	1

To determine the value of a number we multiply the value of the character times the value of the position it occupies. Thus 234 decimal is $2\times10^2 + 3\times10^1 + 4\times10^0$. Remember, any number raised to the zero power is equal to one.

The hexadecimal system is similar. Each position to the left of the decimal point (or, more correctly, the hexadecimal point) has a value of sixteen (the base) raised to a power equal to the number of positions to the left of the decimal point minus one.

16^4	16^3	16^2	16^1	16^0
65536	4096	256	16	1

To determine the decimal value of a hexadecimal number, multiply the character value times the value of the position it occupies and add all the values. For example, 234 hex is equal to $2\times16^2 + 3\times16^1 + 4\times16^0$ which adds up to 564 decimal. Position 3C00 hex in the TRS-80 memory (position one in video memory) would be $3\times16^3 + 12\times16^2 + 0\times16^1 + 0\times16^0 = 15360$ decimal. Pretty laborious!

To go from decimal to hexadecimal is even worse. The decimal number must be divided by the largest power of sixteen that results in a whole number. This value times the power of sixteen is then subtracted from the original and the process repeated. Example 1 shows you how to convert 15360 decimal to hexadecimal.

The Programmable Calculator

Your programmable calculator can perform these multiple operations for you. The only real difficulty is that the hexadecimal system has sixteen characters per position while your decimal calculator has only ten.

You can get around this problem by entering all hexadecimal characters greater than nine as their decimal equivalents. Thus, a hexadecimal A is entered as 10, a B as 11, a C as 12, a D as 13, an E as 14 and an F as 15.

With this procedure in mind, program your calculator to perform the following operations for hexadecimal to decimal conversion.

$$\underline{n} \times 4096 + (2\times256) + (\underline{n} \times 16) + (\underline{n}) =$$

<u>N</u> is a hexadecimal character expressed as zero to fifteen. It is important to note that you must enter four hexadecimal characters. If you are converting hex FF to decimal you must enter 0, 0, 15, 15. The answer will be dis-

```
Label1
n
Store0 ÷ 4096 = Int Store1 × 1000000 = Store5
Recall0 − (Recall2 × 4096) = Store6 ÷ 256 = Int
Store2 = 10000 = Mem+5
Recall6 − (Recall2 × 256) = Store7 ÷ 16 Int
Store3 × 100 = Mem+5
Recall7 − (Recall3 × 16) = Mem+5
Recall5
```

Program Listing 1.

played on your calculator. Hexadecimal numbers larger than FFFF cannot be converted by this routine.

The procedure for converting decimal to hexadecimal is a little more involved. By making liberal use of the calculator's storage facilities I managed to fit both the previous hex to decimal and the following decimal to hex conversion routines in the calculator's 128 program steps (Program Listing 1). By attaching a label to this routine I am able to select it directly by using the GOTO function on the keyboard.

N is the decimal number you wish to convert. The number must be in the range of 0 to 65535. The result, from memory five, will be displayed in eight adjacent display positions; two digits per hex character. Thus 65000 decimal would be converted and displayed as 15, 13, 14, 08.

As before, you must make the mental conversion to FDE8. This conversion, however, is well within the capabilities of your bionic computer. If you convert small numbers, say 256 decimal, the result will be displayed as 15, 15 (FF hex) on the display. The high order digits, being all zeros, are suppressed.

There you have it. It has saved me many hours of laborious pencil scratching and table fumbling. And it has put my old programmable calculator back into active use. ■

```
         3             12         0        0    =  3C00 hex.
4096√15360      256√3072     16√0     1√0
     12288           256
      3072           512
                     512
                       0
```

Example 1. Decimal to Hex Conversion of 15,360.

✓See List of Advertisers on page 306

Generate random sequences according to Poisson, exponential or normal distributions.

Not-So-Random Numbers

Timothy R. Zeigler
USAF
309 N. 49th St.
Grand Forks, ND 58201

The random number generator (RND) available in TRS-80 BASIC is easy to use and serves its purpose very well. It returns random numbers from a uniform distribution. You specify the parameters of this distribution by the argument you use when calling the function.

There is, however, more to random numbers than those returned by RND. Many applications, especially those in the area of simulation, require the user to be able to generate numbers selected at random from a specified distribution with specified parameters. This cannot be done by use of the RND function alone.

The Program Listing contains three subroutines which can be used to generate random numbers from a theoretical distribution of stated parameters. Random numbers can be generated from the Poisson, exponential or normal distributions, depending on which subroutine is called. The functions available in TRS-80 BASIC provide the necessary ingredients to generate these random numbers. A quick review of the three distributions is in order before discussing the subroutines.

Three Common Distributions

The Poisson distribution, Fig. 1a, is a discrete distribution specified by one parameter, Lambda. The mean and the variance of a Poisson distribution are both equal to Lambda. Lambda need not be an integer, but it must be positive. Customer arrivals, in many queuing situations, follow a Poisson distribution with a mean of Lambda.

The Poisson and exponential distributions are closely related. If customer arrivals follow a Poisson distribution, the time between arrivals follows an exponential distribution. The exponential distribution, Fig. 1b, is a continuous distribution specified by the parameter Theta. Theta is the mean of the distribution and must also be positive. Many phenomena follow an exponential distribution. The time between failures of the ICs in your micro is an excellent example.

The normal distribution, Fig. 1c, is by far the most noted of distributions. Its parameters are the mean and the standard deviation. The mean identifies the center of the distribution and the standard deviation the spread. The normal distribution is a continuous distribution. It is also symmetrical.

Generating Other Random Numbers

The subroutines in the Program Listing contain the algorithms necessary to convert the random number that results from RND(0) to a random number from one of the three distributions discussed. Since the parameters of the distribution determine the shape of the resulting distribution, the user must set the parameters before calling the desired subroutine.

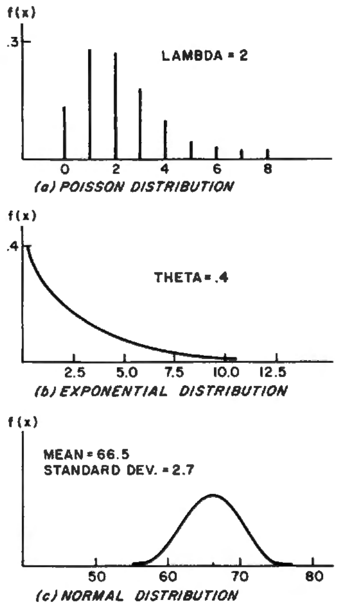


Fig. 1.

Lambda must be set in the variable LA, if a Poisson random number is desired. Theta must be set in the variable TA for an exponential random number. Before calling the normal random number generator, the mean must be set in EX and the standard deviation in SD. In all cases, the random number is returned in the variable RN. Comments in the subroutines also indicate what variables are used internally by each subroutine.

Several goodness-of-fit tests have been conducted on the numbers generated by these subroutines. The tests were accomplished using the Statistical Package for the Social Sciences (SPSS). As written, the subroutines provide numbers that fit the specified distributions and parameters.

Those interested in the "mathemagics" necessary to generate these random numbers will find several textbooks available. My reference was Robert E. Shannon's text entitled *Systems Simulation—The Art and Science*, published in 1975 by Prentice-Hall, Inc. This text also contains FORTRAN listings of similar subroutines. ■

Eds. Note: The line printer used by 80 Microcomputing *prints a [instead of an* ↑.

Program Listing

```
1 CLS:PRINT"LINES 10 - 700 ARE A PROGRAM I HAVE WRITTEN
    TO GENERATE
  RANDOM NUMBERS FROM THE VARIOUS DISTRIBUTIONS.
2 PRINT"THE ARTICLE REFERS TO THE SUBROUTINES THAT BEGI
    N AT
LINE 720.
3 PRINT:INPUT"HIT ENTER TO CONTINUE";A$
10 RANDOM
20 CLS:CLEAR400
30 F$="#####.###":F1$="####"
40 DEFINT I,J,K
50 P$=STRING$(128," ")
60 ON ERROR GOTO 690
70 PRINT "1--> N(EX,ST)
80 PRINT "2--> POSN(LAMBDA)
90 PRINT "3--> EXPON(THETA)
100 PRINT "4--> UNIFORM(0,MAX)
110 INPUT "WHICH DISTRIBUTION DO YOU WANT TO GETNERATE
    1,2,3, OR 4";J
120 INPUT "HOW MANY NUMBERS DO YOU WANT TO GENERATE";II

130 IF II>4000 THEN PRINTII;"IS TO LARGE, TRY SOMETHING
    LESS THAN 4000":GOTO120
140 DIM NUM(II)
150 DIM C(128)     '************ HISTOGRAM
160 ON J GOTO 170  ,190  ,210  ,320
170 INPUT "ENTER THE MEAN";EX
180 INPUT"ENTER THE STANDARD DEVIATION";SD:GOTO 220
190 INPUT "ENTER  LAMBDA";LA
200 GOTO 220
210 INPUT "ENTER  THETA";TA
220 '
230 FOR I=1 TO II    ' HOW MANY NUMBERS TO GENERATE
240 PRINT"I'M WORKING AS FAST AS I CAN":PRINT@814,I
250 ON J GOTO 260  ,270  ,280
260 GOSUB730   : NUM(I)=RN: GOTO290
270 GOSUB830   : NUM(I)=RN: GOTO290
280 GOSUB960   : NUM(I)=RN: GOTO290
290 IK=RN+.5:IF IK<0 OR IK>128 THEN IK=128
300 C(IK)=C(IK)+1:NEXT
310 GOTO350
320 INPUT"ENTER THE MAX";EX
330 FOR I=1TOII:NUM(I)=RND(EX):C(NUM(I))=C(NUM(I))+1
340 PRINT"I'M WORKING AS FAST AS I CAN":PRINT@814,I:NEX
    T
350 INPUT"PRINT NUMBER  Y/N";N$:IF N$="N" THEN 430
360 IF II<=5 THEN J=II: GOTO 420
370 I=II/5: J=II-(I*5): K=0
380 FOR KK=1TOI: PRINT USING F1$;KK+1000;:PRINT"=";:
390 PRINT USING F$;NUM(K+1);NUM(K+2);NUM(K+3);NUM(K+4);
    NUM(K+5)
400 K=K+5: NEXT
410 IF (II-(I*5))=0 THEN 430
420 PRINT USING F1$;KK+1000;:PRINT"=";: FOR K=1 TO J: P
    RINT USING F$;NUM(K);: NEXT
430 PRINT:INPUT "SAVE ON DISK    Y/N";N$
440 IF N$="N" GOTO 530
450 INPUT "ENTER FILENAME";N$
460 OPEN "O",1,N$
470 I=II/5: J=II-(I*5): K=0
480 FOR KK=1TOI: PRINT#1,KK+1000;CHR$(61);:
490 PRINT#1,USING F$;NUM(K+1);NUM(K+2);NUM(K+3);NUM(K+4
    );NUM(K+5)
500 K=K+5: NEXT
510 IF (II-(I*5))=0 THEN 690
520 PRINT#1, KK+1000;CHR$(61);: FOR K=1TOJ: PRINT#1,USI
    NG F$;NUM(K);:NEXT
530 INPUT"IF YOU WANT A HARD COPY OF THE HISTOGRAM
 GET THE PRINTER READY AND TYPE Y  ELSE N";N$
540 CLS:PRINT"MEAN = ";EX,"STANDARD DEVIATION = ";SD
550 FOR IK=1TO128
560 IY=47-C(IK)/(INT(II/1000)+1)
570 FOR IR=47 TO IY STEP -1
580 SET (IK-1,IR)
590 NEXT IR,IK
600 IF N$="N" END
610 LPRINT "MEAN = ";EX,"STANDARD DEVIATION = ";SD,"NUM
    BER =";II:LPRINT:LPRINT
620 FOR ID =3 TO 47
630 FOR IX =0 TO 127
640 IF POINT (IX,ID) THEN MID$(P$,IX+1,1)="*"
650 NEXT IX
660 IF INSTR(1,P$,"*")=0THEN GOTO680
670 LPRINT P$:P$=STRING$(128," ")
680 NEXT ID:END
690 CLOSE
700 END
710 '*****************************************************
    ****
720 '  tHE ARTICLE REFERS TO THE SUBROUTINES THAT FOLLO
    W
730 'GENERATES A NUMBER FROM THE NORMAL DITRIBUTION  N
    (EX,SD)
```

Program continues

STATISTICAL ANALYSIS PACKAGE

"Absolutely the best statistical analysis package I've seen for a microcomputer..."
Eugene Allred, President of Consolidated Data Services

StatPac is a 3-part program written and designed for research, survey and questionnaire processing, data entry, statistical analysis, management and market analysis.

Whether you're a researcher, a programmer or reader and user of data — StatPac can fit your needs because you create the data file format to fit your specifications.

Stat Pac is the SPSS of small computers. It eliminates the need for expensive programming or renting time on large mainframe computers.

Available for Model I (requires NEWDOS) or Model III, 2 disk drives, 48K RAM and a lineprinter. StatPac handles all phases of research analysis including: frequency analysis, descriptive statistics, crosstabs & chi-square, t-test, correlation & linear regression, analysis of variance, multiple linear regression and management reports.

CODEBOOK AND DATA ENTRY

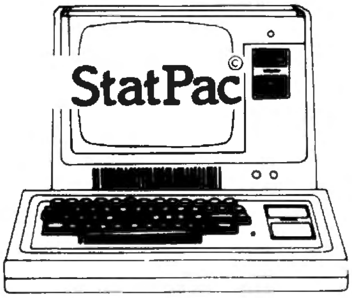
Enter data using key-to-disc system with automatic back-up of data to protect against system crashes. Enter survey, report, test or lab data using fixed or free format. Programs feature screen prompting to guide data entry operator, automatic error detection, user-definable tab stops, repeating keyboard, plus much more.

DATA MANIPULATION

Edit your files with fast machine language editor using "pencil"-like commands, better than "pencil." Verify data files for "out-of-bounds" entries. Recode and transform data using conditional and absolute recodes and unlimited logical operators. Edit files larger than memory size. Accepts up to 2000 cases and up to 255 variables.

STATISTICAL ANALYSIS AND BATCH PROCESSING

Create a "task control file" to specify all analysis for batch processing. Save, edit or reuse task control file on other data sets. "Select if" feature allows analysis on partial data sets. All outputs to parallel or RS232 lineprinter. Error trapping features minimize aborted jobs.



Total package of 3 disks and extensive step-by-step user's manual is $185. Manual may be purchased separately for $30, and may be applied to the purchase price. Entire package includes over 30 integrated programs.

Write or call for free 16 page brochure.

Walonick Associates ✓363
5624 Girard Ave. So.
Minneapolis, MN 55419
(612) 866-9022

Professional software **TRS-80® MODEL II** NOW AVAILABLE!

BASIC CROSS REFERENCE

- FIND WHERE NAMES ARE USED FAST!
- CAN YOU DELETE THAT LINE? FIND OUT!
- DO YOU HAVE DEAD CODING?
- WANT A NICE PROGRAM LISTING WITH DATE & TIME IN HEADING
- SAVE HOURS!
- PARALLEL/SERIAL PRINTERS
- SEVERAL OPTIONS
- 6/8 LPI VARIABLE LINE WIDTH & PAGE DEPTH

$59.99 +$3.00 POSTAGE & HANDLING
$100.00 FOR VECTOR GRAPHICS SYSTEMS.
DOCUMENTATION ONLY $10.00 DEDUCTIBLE ON PURCHASE

DISK SORT

- YOU DON'T HAVE TO BE A PROGRAMMER TO USE IT!
- MENU DRIVEN
- RANDOM FILES
- CHAINS TO SYSTEM OR BASIC PROGRAMS
- EASY TO USE
- FAST!
- SPECS SAVED ON DISK
- EASY TO INSERT INTO JOB STREAM FOR NON-STOP RUNNING!

$69.99 +$3.00 POSTAGE & HANDLING
DOCUMENTATION ONLY $10.00 DEDUCTIBLE ON PURCHASE

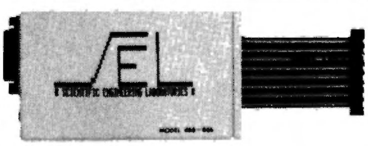
BASIC COMPILER

- TRSDOS*/BASIC COMPATIBLE!
- FASTER THAN BASIC BY UP TO 30 TIMES
- EASY TO USE
- AUGMENTED WITH OUR DOCUMENTATION
- WRITTEN BY MICROSOFT

$350.00 +$5.00 POSTAGE & HANDLING

'TRS-80'* & 'TRSDOS'* ARE REGISTERED TRADEMARKS OF TANDY CORP

GOOD-LYDDON DATA SYSTEMS ✓218
5486 RIVERSIDE DR., CHINO, CA. 91710
MASTER CHARGE or VISA accepted.



Model 488-80B For Model 1 Operation

Model 488-80C For Model 3 Operation

IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1 or 3, Level 2 or DOS with a minimum of 16K.

488-80B or 488-80C $245.
+ shipping, insurance & tax

Optional Relocatable Machine Level GPIB Driver for Assembler Level Programming-$35.
WHEN ORDERING SPECIFY DISK OR TAPE

SCIENTIFIC ENGINEERING LABORATORIES
11 Neil Drive • Old Bethpage, NY 11804
Telephone: (516) 694-3205 ✓291

**Trademark of Tandy Corp.*
There is no affiliation between Scientific Engineering Laboratories and Tandy Corp. or Radio Shack.

SICK

OF PROGRAMS THAT TREAT YOU LIKE AN IDIOT, WASTING TIME & MONEY??

RANDOM ACCESS PAYROLL

- NO Complicated Initialization
- EDIT & LIST
- QUARTERLY Reports
- PROGRAM Loads In Less than 30 sec.

PAY ANY EMPLOYEE ANYTIME

- SALARIED OR HOURLY
- COMPLETE Including EIC
- PRINTS on NEBS 9020 CHECKS
- NO SPECIAL PAYROLL CHECKS
- SPECIAL HOURS—SPECIAL PAY
- TWO SAVINGS—INCLUDING RIA
- STATE TAX—WORKMENS COMP.
- CLASSED BY Occupation or Dept.
- PAYSTUB Shows Year-to-date
- No Filenames—All Automatic

SEND YOUR STATE TAX SCHEDULE

- We'll Customize for your State Tax

WRITTEN IN BASIC FOR COMPLETE CONTROL

Documentation & Disk $55.00
Documentation only $10.00-credit to purchase

MASS ADD 5% or Exempt number

Requires:
Min. 32K-1 Drive-Printer = 50 EMP
48K-2 Drives-Printer = 100 +

TEL. 7 AM-9:30 PM EASTERN
(617)-359-2364/6370

MEDFIELD ✓421
COMPUTER SOFTWARE
39 GREEN ST., MEDFIELD, MA 02052

```
        EX--MEAN   SD--STANDARD DEVIATION
        ROUTINE USES VARIABLES V8,V9,EX,SD, AND RN
        EX AND SD MUST BE SET BY USER.  RANDOM NUMBER IN
       RN.
740 '
750 '   SEE  SHANNON'S  "SYSTEM SIMULATION"  PAGE 360-3
    62
         FOR AN EXPLANATION OF THE METHOD.
760 '
770 V9=2*RND(0)-1
780 V8=V9[2 + (2*RND(0)-1)[2
790 IF V8>=1 GOTO 770
800 RN = EX + (V9*SQR((-2.0*LOG(V8))/V8))*SD
810 RETURN
820 END
830 '
840 'GENERATES A NUMBER FROM THE POISSON DISTRIBUTION W
    ITH A
     MEAN AND VARIANCE OF LAMBDA.
     LA--LAMBDA
     ROUTINE USES V7,V8,LA, AND RN
     USER MUST SET LA.  RANDOM NUMBER RETURNED IN RN.
850 '
860 '   SEE SHANNON'S  "SYSTEM SIMULATION" PAGE 358-359

        FOR AN EXPANATION OF THE METHOD.
870 RN=0.0
880 V7=EXP(-LA)
890 V8=1.0
900 V8=V8*RND(0)
910 IF (V8-V7)<0 THEN RETURN
920 RN=RN+1.
930 GOTO 900
940 RETURN
950 END
960 '
970 'GENERATES A RANDOM NUMBER FROM AN EXPONENTIAL DIST
    RIBUTION      WITH A MEAN OF THETA.
     TA--THETA
     ROUTINE USES V7,TA, AND RN.
     USERS PROVIDES VALUE FOR THETA IN TA.  RANDOM NUM
    BER IN RN
980 '   SEE SHANNON'S  "SYSTEM SIMULATION" PAGE 360
        FOR AN EXPANATION OF THE METHOD.
990 RN=-TA*LOG(RND(0))
1000 RETURN
```

THE ARTS

COMPU-CAROLS—We are proud to present a selection of Christmas carols, played by your TRS-80. Just place an AM radio next to your keyboard and you'll be amazed at the quality of this computer-generated music. You'll hear AWAY IN A MANGER, NOEL, SILENT NIGHT, O LITTLE TOWN OF BETHLEHEM and eight more of your favorite carols. **(T1) Order No. 0036R $9.95.**

DOODLES AND DISPLAYS II—It includes: • DOODLE PAD: Draw pictures and save them on cassette tapes. • SYMMETRICS: An electronic kaleidoscope that's constantly changing. • DRAWING: Like DOODLE PAD, but for the serious artist. Over 40 user commands. • RANDOM PATTERN DISPLAY: The computer does the drawing, but those with itchy fingers can make alterations. • MATHCURVES: Bring those geometry lessons to life. Six different geometrical curves on the screen of your TRS-80. • RUGPATTERNS: Designs rug patterns with a choice of user or computer control. **(T1) Order No. 0042R $7.95.**

MUSIC MASTER—Includes these four audio treats: • MICRO-ORGAN: This program changes your computer into a musical instrument, with a range of four octaves with three voices! You can play sharps and flats to imitate the sounds of an organ, harpsichord or piano. • KALEIDOPY: Now you can have a computerized "player piano." Generate a symmetrical graphics pattern and then see it transformed into music. • COMPOSER: Experiment with computer-generated music. You can select the length of the piece, its scale, and its tempo. • KEYMANIA: Test your memory and your musical ear. One to four players try to repeat the melody that the computer creates. **(T1) Order No. 0084R $9.95. Model III compatible**

ELECTRONICS

HAM PACKAGE I—This versatile package lets you solve many of the problems commonly encountered in electronics design, including: • BASIC ELECTRONICS WITH VOLTAGE DIVIDER: Solve problems involving Ohm's Law, voltage dividers and RC time constants; • DIPOLE AND YAGI ANTENNAS: Design antennas easily, without tedious calculations. **(T1) Order No. 0007R $7.95. Model III compatible**

ELECTRONICS I—This package will not only calculate component values for you, it will also draw a schematic diagram. Included are: • TUNED CIRCUITS AND COIL WINDING: Design tuned circuits without restoring to cumbersome tables and calculations; • 555 TIMER CIRCUITS: Design astable or monostable timing circuits using this popular IC; • LM-381 PREAMP DESIGN: Design IC preamps with this low-noise IC audio amp. **(T1) Order No. 0008R $7.95. Model III compatible**

QSL MANAGER—Ever looked at your log book and wondered if you sent a QSL card to the operator you worked last week? Maybe you sent a QSL but can't remember getting one in return. The QSL MANAGER will help you set up a computerized log book that gives you *instant access* to your records. Make complete log entries which include: Date, Time, Call sign, Name, Band, both the sent and received Signal Reports, the Mode, whether a QSL card was sent or received and any remarks you want to add. The QSL MANAGER program has built-in editing features that let you keep your log book up to date. **(T2) Order No. 0151RD $19.95. Model III compatible**

HOME EDUCATION

MONEY MADNESS—You can experience the Raw Power of High Finance with two Big Money empires. • MILLIONAIRES: Can you manipulate $1000 into a million dollars in fifteen years? It all depends on your strategy as you buy and sell properties, negotiate bank loans, collect rentals and accept sealed bids. • TIMBER BARON: An in-depth experience of the timber business, from the time you cut the trees until your milled lumber reaches the market. These transactions are affected by those tough, unexpected eventualities that can upset the most careful plans. **(T1) Order No. 0156R $9.95. Model III compatible**

TEACHER'S AIDE—Now you can have the benefits of Computer Aided Instruction (CAI) in your own home. Create a question and answer lesson (up to 8000 characters), save the lesson on disk, then create an entire sequence of lessons. Perfect for parents, teachers and students who need the unlimited patience and undivided attention only a computer can provide. **(T2) Order No. 0214RD $34.95. Model III compatible**

GRADE BOOK—Teachers, now you can use the speed and accuracy of the computer to help calculate student grades. Just type in the grades for tests, quizzes, homework, classwork or special projects to calculate and display individual grade averages. You can also obtain a cumulative grade for a specific marking period—or a whole year! **(T1) Order No. 0050R $9.95. Model III compatible**

TEACHER—This program enables you to create your own tests, quizzes and exercises for the education of your children. You can even provide "graphic" reward for your children and provide hints for problem solving. **(T1) Order No. 0065R $9.95. Model III compatible**

LIFE—Create "living" organisms in which cells are constantly active. They are born, they multiply, they die. This computerized version of LIFE is based on the well known game popularized by Martin Gardner. You can create one-cell organisms, then observe their growth patterns. The library of commands give you unlimited versatility in the control of the cell patterns *you* have arranged. **(T1) Order No. 0078R $9.95. Model III compatible**

ARCHIMEDES' APPRENTICE—This two-part package will teach you the formulas used to find the volume of any solid object including parallelopipeds (cubes and rectangular solids), prisms, pyramids, cylinders, cones and spheres. It will show you on-screen diagrams of these figures, and present you with the formulas you'll need to compute their volumes. **(T1) Order No. 0092R $9.95. Model III compatible**

TYPING TEACHER—This complete seven-part package takes you from initial familiarization with the keys, through typing words and phrases, to complete mastery of the keyboard. Your computer can even become a bottomless page for typing practice. **(T1) Order No. 0099R $9.95. Model III compatible**

VIDEO SPEED READING TRAINER—Most people's reading speed is limited simply because they read individual letters or words. Now you can increase your reading speed and comprehension by reading whole words and phrases. This package will train your mind to quickly recognize numbers, words, letters and phrases. Start at any speed level at which you are comfortable and the computer will automatically advance you as your reading speed and comprehension increases. **(T1) Order No. 0100R $9.95. Model III compatible**

WORDWATCH—four different programs to entertain and educate. • WORD RACE—race to the finish line of defining words correctly; • HIDE N SPELL— find the misspelled word, then correct it; • SPELLING TUTOR—a spelling lesson, but beware, the spelling may become unusual. There you have it, Wordplay x four = WORDWATCH. **(T1) Order No. 0111R $7.95. Model III compatible**

MIND WARP—This game includes: • MIND TWIST: a Mastermind-type game with a twist. Try to guess the computer's secret digit sequence. • MIND BENDER: A multi-level game where you must discover the computer's secret code. It's no mystery, the MIND WARP package is for puzzle lovers everywhere. **(TI) Order No. 0118R $9.95. Model III compatible**

INVESTOR'S PARADISE—Here are two programs to test your skill in the stock market. • STOCK TREK: a stock market simulation in which you and up to five other investors buy and sell stocks. • SPECULATION: a step beyond a mere simulation, you enter financial data on up to 25 real companies and start playing the market. This package lets you experience the thrills and triumphs of the stock market without risking a dime! **(T1) Order No. 0125R $9.95. Model III compatible**

IQ TEST—IQ TEST will administer and score an intelligence test in just 30 minutes. There are three equivalent tests, each consisting of 3 questions that survey your general knowledge and problem solving abilities. **(T1) Order No. 0157R $9.95.**

SPECIAL BUSINESS

BOWLING LEAGUE SECRETARY—This package is simple to operate and provides a dynamic reference to all the names of individual bowlers, their team numbers, scores, team names, league data and all necessary statistics. The system is highly adaptable, with 17 different scoring options that allow you to custom tailor the program to suit your league's special needs. And, if you even have any problems, simply type HELP and the program will give you an explanation of what information is needed—complete with a sample entry. The system puts at your fingertips all individual weekly scores, team cumulative scores, bowler cumulative scores and individual leaders in the following categories: high single, high series, high average and high points. **(T2) Order No. 0095RD $49.95. Model III compatible**

*TRS-80 is a trademark of Tandy Corporation.

SEE YOUR LOCAL
INSTANT SOFTWARE DEALER OR

Just Call Toll-Free
1-800-258-5473

BEGINNER'S RUSSIAN—In order to understand a foreign culture, you must know its language. The three programs in this package will give you on-screen displays of the characters of the Cyrillic alphabet, detailed instructions of their proper pronunciation and exercises that will have you recognizing and speaking simple Russian words. An excellent package for students, businessmen, scientists or anyone who is interested in learning the Russian language. **(T1) Order No. 0136R $9.95. Model III compatible**

EVERYDAY RUSSIAN—will acquaint you with the words for various foods, places to eat, signs and the names of stores—exactly what a traveller needs to know. Each of the three parts of the package not only teaches you the words but quizzes you on them as well. You can even practice typing in Russian. Discover the Russian language today! **(T1) Order No. 0137R $9.95. Model III compatible**

NO MATTER WHAT YOUR NEEDS ARE, INSTANT SOFTWARE HAS A PROGRAM FOR YOU.

WRITE FOR OUR NEW INSTANT SOFTWARE CATALOG

BOWLING LEAGUE STATISTICS SYSTEM—Keeps a computerized list of league data, team data and data for each bowler. Extremely flexible, it has a total of 16 different options to let you modify the program to suit your league's rules. It is easy to use and has a built-in "HELP" feature to aid you. **(T1) Order No. 0056R $24.95.**

HOME/PERSONAL

HOUSEHOLD ACCOUNTANT—Save with these two programs: • BUDGET & EXPENSE ANALYSIS: It has nine sections for income and expenses and an option for quarterly/yearly reviews. • LIFE INSURANCE COST COMPARISON: Compare the total costs of various insurance policies. Contrast term with whole life. It will store and display up to six prospective policies. **(T1) Order No. 0069 $7.95. Model III compatible**

PERSONAL BILL PAYING—You can keep a computerized list of ALL your bills (up to 22 accounts), each listed with its name, number, due date and amount owed. Individual accounts can be displayed with a month-by-month breakdown of payments (including check numbers) and current accounts can be seperated from inactive ones. It allows you to save the data to tape for future use. **(T1) Order No. 0103R $7.95 Model III compatible**

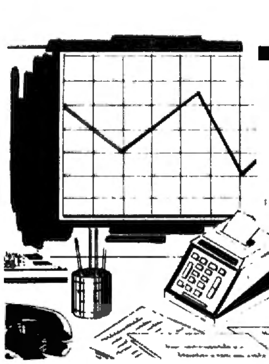
We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

Instant Software™

PETERBOROUGH, N.H. 03458



TRS-80* Software From Function to Fantasy

POPULAR GAMES

BEGINNER'S BACKGAMMON/KENO—Why sit alone when you can play these fascinating games: • BACKGAMMON: Play against the computer in a game that's sure to sharpen your skills; • KENO: Enjoy this popular Las Vegas gambling game—guess the right numbers and win big! **(T1) Order No. 0004R $7.95. Model III compatible**

CHESSMATE-80—This versatile chess opponent gives you a choice of ten levels of play, from the "blitz" level (the computer has 3 seconds to move) to the infinity level (where the computer will consider every possible move—which could take years). This machine-language program is a conservative player and follows all the rules of international play. CHESSMATE-80 can teach you how to move and allow you to set up the board and play end games or special problems. CHESSMATE-80 battled Sargon II to a draw at two minutes a move and beat Microchess 1.5 in six moves. **(T1) Order No. 0057R $19.95. Model III compatible**

YOUR CRIBBAGE AND CHECKERS PARTNER—CRIBBAGE is a two-person game that you are sure to enjoy. This is NOT a tutorial—it is a game worthy adversary. CHECKERS: An old favorite which follows international rules, including multiple jumps. **(T1) Order No. 0068R $9.95. Model III compatible**

CARDS—A one-player package to let you play, with your computer, these famous games: • DRAW AND STUD POKER: These programs will keep your game sharp; • NO-TRUMP BRIDGE: Develop your strategy and (hopefully) increase your skill. **(T1) Order No. 0063R $7.95. Model III compatible**

FLIGHT SIMULATIONS

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE—•RAMROM PATROL: Destroy the RamRom ships before they capture you. •TIE FIGHTER: Wipe out the enemy Tie fighters and become a hero of the Rebellion. •KLINGON CAPTURE: You must capture the Klingon ship intact. **(T1) Order No. 0028R $7.95. Model III compatible**

FLIGHT PATH—This three-part package includes: •MOUNTAIN PILOT: Become a daring bush pilot and fly supplies to a remote mining camp. You must cross mountain ranges and struggle with headwinds, tricky navigation and rapidly diminishing fuel. •O'HARE: A control tower simulation for you would-be Air Traffice Controllers. You are responsible for the lives of hundreds of passengers as you guide aircraft through your control sector. •PRECISION APPROACH RADAR: Combines the skills of pilot and Air Traffic Controller, as your commands guide an aircraft in its approach to the field and a safe landing. **(T1) Order No. 0171R $9.95. Model III compatible**

BALL TURRET GUNNER—Imagine yourself at the control console of a strategic laser weapon, deep in the space lanes. Your hindsight detector informs you of a Gnat fighter coming in for an attack so you swivel you laser turret until you can see the target. Watch the Range Indicator and your Targeting Computer's readout closely, because you'll only have a fraction of a second to catch him in your sights. Will you transform the Gnat into a ball of ionized gas or will you see that blinding flash that means The Big Demotion? BALL TURRET GUNNER, with you choice of multiple levels of difficulty, optional sound effects and excellent graphics, is more than a game. It's an event to be savored. **(T1) Order No. 0051R $9.95. Model III compatible**

JET FIGHTER PILOT—In this brilliantly realistic simulation, you become the pilot of a twin turbo-jet fighter. Begin your mission from either the deck of a carrier or from an airfield. During flight, you'll need to constantly monitor your display and make the necessary adjustments to the throttle, flaps, and air spoilers; you must decide when to retract landing gear and release your drop tanks! There is an on-board Navigational Computer, a Glideslope/Localizer and a Weapons Control Computer. Earn your wings with JET FIGHTER PILOT. **(T1) Order No. 0159R $14.95. Model III compatible**

SPACE TREK II—Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power and warp drive. **(T1) Order No. 0002R $7.95. Model III compatible**

AIR FLIGHT SIMULATION—Take off and land your aircraft without making a crater. This "instruments only" simulation starts you with a full tank of fuel, which gives you a maximum range of about 50 miles. You'll get constant updates of air speed, compass heading and altitude. After you've acquired a few hours of flight time, you can try flying a course against a map or doing aerobatic maneuvers. **T(1) Order No. 0017R. $9.95. Model III compatible**

SPACE TREK IV—•STELLAR WARS: Engage and destroy Tie fighters in your attack on the Death Star. For one player. •POPULATION SIMULATION: A two-player game where you control the economy of two neighboring planets. You must decide: Guns or Butter? **(T1) Order No. 0034R $7.95. Model III compatible**

BASIC AND INTERMEDIATE LUNAR LANDER—Bring your lander in under manual control. The basic version is for beginners; the intermediate version is more difficult, with a choice of landing areas and rugged terrain. **(T1) Order No. 0001R $7.95. Model III compatible**

COSMIC PATROL—We put you in command of a small interstellar patrol craft. You must defend Terran space and prey on the Quelon freighters that carry vital war supplies—but beware of their I-Fighter escorts. They're well armed, extremely fast and they NEVER miss! With its real-time action, impressive sound option and superb graphics, this machine-language program is the best of the genre. **(T1) Order No. 0223R $14.95. Model III compatible**

Airmail Pilot —Return to the early days of aviation. You must fly the mail from Columbus to Chicago. Your Jenny, a cloth-covered biplane, must take you through unpredictable winds, hail and electrical storms. Your mission is to get the mail through in the shortest possible time. There is an on-board clock to time you flight, from takeoff to touchdown... assuming you are able to complete it. **(T1) Order No. 0106R $9.95. Model III compatible**

NIGHT FLIGHT—Your mission is to fly over the North Atlantic and make a nighttime photo/recon flight above the enemy fleet. NIGHT FLIGHT lets you take-off, fly and land a propellar-driven aircraft. You can practice approaches and landings with an on-screen display of the landing field information—it will practically teach you to fly. **(T1) Order No. 0117R $9.95. Model III compatible**

COMP-U-NOVELS

WHO-DUN-IT? Criminal elements have committed five dastardly crimes. As the investigating detective, you must solve them.

You can compete against either Detective Nybbles, a computerized sleuth, or up to four other human detectives.

•DEDUCTION: Guess the order of four symbols out of six or seven different ones. To make things even more complicated, you can let the computer repeat symbols and have a range of 2401 possibilities. **(T1) Order No. 0047R $7.95. Model III compatible**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. **(T1) Order No. 0043R $7.95. Model III compatible**

There are over 450 Instant Software dealers throughout the U.S.A. and the world.

We ship our orders within 48 hours.

*TRS-80 is a trademark of Tandy Corporation.

CODE—Minimum System Required

(T1) = TRS-80 Model I Level II, 16K RAM
(T2) = TRS-80 Model I Level II, 16K RAM with Expansion Interface 16 + K RAM and one disk drive
(T3) = TRS-80 Model II, 32K RAM

* A trademark of Tandy Corporation

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR

Just Call Toll-Free 1-800-258-5473

WRITE FOR OUR NEW INSTANT SOFTWARE CATALOG

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

Instant Software™

PETERBOROUGH, N.H. 03458

HOME/PERSONAL

THE WORDSLINGER—An economical word processing program that was designed for the individual user or small business featuring: automatic formatting; text editing; and tape storage. Once you've used the WORDSLINGER, you won't want to go back to your typewriter. **(T1) Order No. 0129R $29.95.**

MIMIC—Test your memory and reflexes with five versions of this popular game. You must match the sequence and location of symbols displayed on your monitor within the time limit. Instructions on how to produce accompanying sound effects. **(T1) Order No. 0066R $7.95**

CLIMATE COMP—This two-program package includes: WEATHER FORECASTER, which gives you a short range weather forecast based on the information that you enter and WEATHER PLOT, which will display climatological data for any major city in the United States. **(T1) Order No. 0102R-1 $19.95. Model III compatible**

BODY BUDDY—Includes these three programs: ● ADULT CALORIC REQUIREMENTS: Will determine your Basal Metabolic Rate and suggest strategies to achieve your ideal weight! ● FLEXI-DIET: Creates an "infinite" number of diet menus, on a day-to-day basis. Choose your caloric intake, from 600 to 2400 calories per day. The ● ANATOMY QUIZ program teaches a mini-lesson on the various organs of the human body, giving location, size and function(s). **(T1) Order No. 0109R $9.95. Model III compatible**

ENERGY CONSUMPTION—This program will record and analyze your utility bills for up to five years, when you supply the following information. Gas/Water/Electricity used and their respective costs. It will calculate six monthly usage averages and unit costs. Data can be compared for any month or multi-month periods. **(T1) Order No. 0132R $9.95. Model III compatible**

BUSINESS

SALES ANALYSIS—If your business is sales, you're faced with some unique problems. This package is divided into several modules to help solve those problems: The SALES ANALYSIS module is designed to provide guidelines for determining sales performance, to analyze this performance and show you where it can be improved. The DATA STORAGE module allows you to store data in an automated processing ledger. The MANAGEMENT ANALYSIS module can take all the sales records for your group and show you who your best salespersons are, who needs more training and give you a sales forecast. Finally, the MARKET ANALYSIS module can show you where determined sales efforts can produce the most success. **(T1) Order No. 0131R $24.95. Model III compatible**

ORACLE-80—will provide you with business analysis and forecasting capabilities previously available only on large computer and time-sharing systems. A flexible, professional time series analysis and forecasting package for use in product planning, business planning, sales forecasting and more. Financial managers and economists can analyze economic climates and investigate business cycles. ORACLE-80 is designed to be used and understood by the typical businessperson. All input and output is written in plain English and the package documentation carefully explains *all* the functions of the program. ORACLE-80 puts the future in your hands. **(T2) Order No. 0140R $75.00. Model III compatible**

BUSINESS PACKAGE IV—This business package contains two programs: ● BUSINESS CYCLE ANALYSIS: This program can plot the expansion and contraction cycles of any aspect of your business. ● FINANCIAL ANALYSIS: Now you can get the figures for any type of annuity, sinking fund, or mortgage and compute the yield and value for bonds. The package includes a blank data tape. **(T1) Order No. 0019R $9.95. Model III compatible**

FINANCIAL ASSISTANT—Compute the figures for a wide variety of business needs, including: ● DEPRECIATION: Figure depreciation on equipment five different ways. ● LOAN AMORTIZATION: Enter a few essential factors and get a complete breakdown of all costs and schedules of payment for any loan. ● FINANCIER: Performs thirteen common financial calculations. ● 1% FORECASTING: Use it to forecast sales, expenses, or any other historical data series. **(T2) Order No. 0072R $7.95. Model III compatible**

CHECK MANAGEMENT SYSTEM—Use this program for writing checks and maintaining records. You can make entries, edit/correct entries and print out the checks. It will also search and display records by number, code, date, description or amount. A Code and Search routine allows you to print a report of all checks written for specific expenses. You can print your letterhead and account number at the top of each report. System requirements: **(T2)** with a compatible tractor-feed printer. **0147RD $39.95. Model III compatible**

ACCOUNTS RECEIVABLE/ACCOUNTS PAYABLE—These Model I programs will handle the drudgery involved in AR/AP entries. They will also provide invoices, statements, reports and more. Each program is capable of handling up to 1500 entries per month, posted to as many as 760 accounts. The AR/AP package is ideal for any small business and can easily be used by anyone familiar with AR/AP operations. System requirements (in addition to T2: Three disk drives and a Line Printer (tractor-feed). **Order No. 0075RD $199.95. Model III compatible**

MAIL/LIST—With a five-inch drive, you can store up to 600 names per disk without DOS, or 300 names with DOS. The program maintains separate alphabetical and ZIP code files under constant sort. When you add a name or ZIP code to your list, it will be inserted into its correct position in the file. The program will record your data in nine fields: address, city, state, ZIP code, phone number, phone extension and name (2) plus a five character code field. The best feature of this program is the sort process that lets you determine alphabetical or ZIP code order for label printing. **(T2) Order No. 5000RD $99.00 Model III compatible**

ONE-D MAILING LIST—A comprehensive mailing list program that will run on only ONE disk drive! Up to 17 fields of selection for name/address retrieval. Its features include: Auto-sort (alphabetic or ZIP code). Easy error correction and recovery. Prints selective listings. Supports up to 4 drives. Prints mailing labels and listing of all names on file. **(T2) Order No. 0123RD $24.95. Model III compatible**

EXECUTIVE EXPENSE REPORT GENERATOR—Provides you with emergency relief in the form of a clear, plausible expense layout. Input your grand total and cash advance (if any), and you'll receive an itemized expense report, from breakfast to snacks. **(T1) Order No. 0135R $9.95. Model III compatible**

GAMES

WINNER'S DELIGHT—Do you enjoy a challenge? Then try WINNER'S DELIGHT including: ● AMAZING: You must escape from a maze, one that you view from the inside, working against the clock; ● JUNIOR CHECKERS: Not your usual game of checkers . . . the challenge is to beat the computer in the fewest number of moves; ● JUMBO JIGSAW: Fit the pieces together in the fewest number of tries; ● THIRTEEN WAYS: Try to fill up your columns with the numbers you roll on the dice—the computer will try to fill its columns first! **(T1) Order No. 0124R $9.95. Model III compatible**

FUN PACKAGE I—Why call it "Fun Package"? Judge for yourself! This entertaining package includes: •ROCKET PILOT: Flying it is easy—it's the landing that's tough! •PAPER, ROCK, SCISSORS: It's the time-honored game just as you remember it, played against your TRS-80. •HEX I: Just when you master this puzzle game, the computer will increase the difficulty. •MISSILE ATTACK: **Use your missiles to protect your city from jet attack. Requires a TRS-80 Level I 16K. Order No. 0037R $7.95. Model III compatible**

DEMO III—The biggest package ISI has ever released, including: ● RACE 1: Careen around the race course as you try to beat the clock; ● TARGET UFO: Destroy all the invading UFOs; ● LIFE: Experiment with this simulation of the life cycle of a colony of bacteria; ● PHONE NUMBER CONVERTER: Change those hard to remember 7-digit phone numbers into easily remembered words; ● BIORHYTHM: Plot biorhythm curves for anyone, anytime; ● GRAPHICS PROGRAM: This program will show you what your TRS-80's graphics display can do; ● RACE 2: Five different tracks for the more experienced driver; ● HORSE RACE: Up to nine players can bet on and enjoy our most entertaining horse race program; ● DRAWING BOARD: Draw pictures or messages and store them in memory or on cassette tape with this easy-to-use program; ● 24-HOUR CLOCK: Transform your computer into an accurate digital clock. **(T1) Order No. 0055R $7.95**

OIL TYCOON—Avoid oil spills, blowouts and dry wells as you battle to become the world's richest oil tycoon. Two players become the owners of competing oil companies as they search for oil and control their companies. **(T1) Order No. 0023R $7.95.**

BOWLING—Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. **(T1) Order No. 0033R $7.95. Model III compatible**

DEMO II—contains: ● TIC-TAC-TOE: An old time favorite with three levels of difficulty; ● TIME TRIALS: Try to beat the clock as you race your car through curves, chutes, and chicanes; ● MAZE: One or two players can search through the maze for the secret square; ● HANGMAN: One or two players can try to guess the secret word; ● WHEEL OF FORTUNE: Choose your number, place your bet and see if you can break the bank (for one to eight players); ● HURRICANE: You can track and monitor hurricanes in *any* part of the world; ● BUGSY: Can you build your Z-80 bug before the computer does? ● HORSE RACE: Pick a sure winner and place your bet (for 1 to 100 players). **(T1) Order No. 0049R $7.95. Model III compatible**

BATTLEGROUND—It is late 1944 and the Allied forces are sweeping toward Berlin. As General in command, you study the map. At your command are tanks, planes, artillery, infantry, engineers, and vehicles. The battle map of your sector will fill with markers to show the development of your forces. You and your opponent will assume the roles of warring Generals, as the battle unfolds. The stark reality of World War II comes alive in BATTLEGROUND. **(T1) Order No. 0141R $9.95. Model III compatible**

SKIRMISH-80—Check out these great games: ● MISSION IMPOSSIBLE: Your objective in this real-time simulation is to drive your tank into a prison courtyard, rescue a jailed prisoner and escape; ● TRAP: A two-player game, in which you must maneuver your opponent into a position where he is hopelessly trapped; ● WIPEOUT: A two-player game in which your mobile gun gets points by destroying as many obstacles as possible, but be careful—some of those obstacles are explosive mines; ● BLOCK-'EM: A two-person competition in which your moving "snake" tries to force your opponent to hit either (1) your trail, (2) his own trail, (3) the boundaries of the field, or (4) any randomly place barriers. The strategy is, of course, to leave you opponent no safe move. **(T1) Order No. 0070R $9.95. Model III compatible**

POPULAR GAMES

OTHELLO—In the game of Othello, there is no such thing as lucky move. The game is a constant test of concentration and tactics.

Othello pits your strategic powers against a merciless, computerized opponent. You play on a board of 64 squares. When you capture your opponent's game disks (by bracketing them with your own), they immediately change sides, to become members of your set.

Here's a maddening, frustrating, but always engrossing, game for your TRS-80. (T1) **Order No. 0046R $9.95. Model III compatible**

SEE YOUR LOCAL
INSTANT SOFTWARE DEALER OR

Just Call Toll-Free
1-800-258-5473

*TRS-80 is a trademark of Tandy Corporation.

NO MATTER WHAT YOUR NEEDS ARE,
INSTANT SOFTWARE HAS
A PROGRAM FOR YOU

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

✓2

Instant Software™

PETERBOROUGH, N.H. 03458

Take the Hertz out of continental computing with this battery charger based power supply.

An Alternate Source

John G. Conner KG6JIC
CTF 67 Box 2
FPO New York, NY 09521

I have been operating my TRS-80 on 110 V ac 50 Hertz for the past two months, but have had some problems.

I am in the U.S. Navy and am required to move periodically to new locations around the world. I have been an amateur radio operator for several years and recently acquired the computer bug.

About three weeks before my departure from Guam, I acquired a 16K TRS-80 Level II. My next duty station was to be Naples, Italy.

Blown Fuse

After my arrival in Naples, two of the three boxes in which my 80 and a Quick Printer II were packed, arrived. I plugged the keyboard into 110 V ac 50 cps power available in my office. The VDT was in the last box that had not yet arrived, but a few quick LPRINTs on the keyboard revealed that it was OK.

Four weeks later the last box arrived. I unpacked it hastily and plugged it in; the screen lit up but no MEM SIZE?. The red light on the keyboard was not glowing. The power supply was plugged in, so I concluded that no power was getting to the keyboard. There was a fuse inside the power supply, but Radio Shack's technical manual was no help as it just showed a square marked "power supply."

I checked the ac plug with an ohmmeter. It read open. I pried open the power supply and found a wired 1/2-Amp fuse on a printed circuit board. The fuse was indeed open, so I replaced it and put the case back together with plastic electricians' tape. The TRS-80 worked for about 10 minutes, then the fuse blew again.

A few checks with an ohmmeter indicated that the keyboard was not overloading the power supply, so I decided that the 50 cps power was the culprit. I increased the size of the fuse to 3/4 Amp and put the power supply back into service. The 3/4-Amp fuse lasted about a half hour.

In desperation I soldered a 1-Amp fuse in place and turned the computer on again. The TRS-80 operated for the rest of the day.

Returning to work the next day, the fuse blew again. Removing the electricians' tape, I saw that the transformer had gotten very warm during the night and part of the plastic case was deformed from overheating. Additional checks with an ohmmeter revealed terminal damage; the windings were shorted. Back to the drawing board.

It appeared that the transformer was operating to its limits on 60 Hertz, and it could not tolerate the additional current when operating on 50.

I checked a hardware store and found a 10 Amp 6/12 V battery charger. 10 A seemed like a little overkill. The specs on the bottom of the Radio Shack power supply listed the output at 17 V ac at 1 A and 19 V dc at 350ma. I decided to give the 10 A brute a try. A shipmate supplied me with two chassis mounting fuse holders, a toggle switch and a 10,000 uF 50 volt Cap. A trip to the audio department at the PX turned up a standard 5-pin DIN plug. I drew out the schematic of the battery charger (Fig. 1) and opened it to make the small changes to turn it into a computer power supply.

The outputs were fused to protect the supply and the keyboard. Initial voltage readings (no load) showed the ac output to be 26 V ac and the dc output to be 16 V dc. The 16 V dc looked close enough, but I was worried about the 26 V ac.

My shipmate let me paw through his junk box, and I came up with two 10-Watt, 18-Ohm resistors. I placed these in parallel with each other then in series with the ac output. Ohms law said that if the ac circuit did draw one Amp, then this combination should drop nine volts. I turned the computer on to measure the voltage under load. The dc remained at 16 V dc. The ac dropped less than one-half a volt. Apparently the ac load is considerably less than 1 A.

After I carefully studied the information available on the schematic in the technical manual, I was convinced that it should be

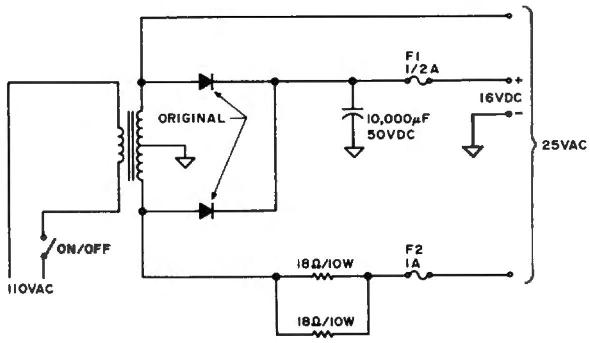


Fig. 1. 10 Amp Battery Charger

OK and I continued to operate the computer.

The TRS-80 has been in operation six to seven hours a day for the last month with this power supply with no apparent problems.

Not once have the severe dips in power during the evening caused me to drop a single bit of information. The transformer in the converted battery charger runs cool and I am running well within its ratings.

That's it. If you're coming to Europe and planning to bring your TRS-80, the voltages throughout Europe are usually 220 V ac at 50 Hertz. The 220 V ac is no problem, as stepdown transformers are readily available from 100 Watts to 5 kW. They are nicely built and accommodate a range of common voltages. I have several in the house and am currently running the TRS-80, cassette, VDT and Quick Printer II on a small 150-Watt transformer. I've discovered no problems. ■

A copy utility for single disk systems.

One Wheel Drive

Dan Keen and Larry Pezzuto
RD 1 Box 432
State Highway 83
Cape May Court House,
NJ 08210

Many times you need a program on several disks. Perhaps you want a safety backup, in case gremlins get into the only existing copy of your favorite program. If you have a program like TSHORT, you may want to transfer it to a disk you use often, so you don't have to load it separately. Or, again, maybe you want to duplicate a data file so you can work on the program that created the file, or back up an important business file.

If you have two disk drives, backup copies are rapidly made with the COPY command.

If you only have one disk drive, however, you may have to slowly copy a text file by loading it into memory and then saving it. If the file is sequential, random data, or machine language, you may have to use the BACKUP command and copy the contents of the entire disk, then delete everything except the program you want. This also means copying onto a blank disk.

Here's a copy utility that comes to the rescue of the one-drive owner.

This program allows you to copy any file up to 16K (you must have 32K in your computer to use this utility). It also allows you to change the file name of the copied version. Keep the extension on machine language programs, however, or they won't function.

First load the program you want to copy into memory. When you have the destination disk in the drive, it dumps the memory contents back onto disk.

After using this program, type CLEAR 10 in the command mode, or you won't have enough memory space to run anything else, as this program clears 20000. ■

```
4 CLS:CLEAR20000:DIMS$(64):DIMS2$(64):PRINT" COPY  UTIL
     ITY ":PRINT
5 INPUT"ENTER FILENAME OF PROGRAM TO BE COPIED: ";A$
6 INPUT"ENTER THE NAME OF THE COPIED FILE: ";B$
8 PRINT:INPUT"HIT <ENTER> WHEN YOU HAVE SOURCE DISK REA
     DY ";C$
10 OPEN"R",1,A$:FIELD1,255ASMY$,1ASLB$
20 FORX=1TOLOF(1):GET1,X:S$(X)=MY$:S2$(X)=LB$:NEXT
30 EF=LOF(1):CLOSE
35 PRINT"FILE ";A$;" HAS BEEN STORED IN RAM."
40 PRINT:PRINT"READY DESTINATION DISK FOR RECIEVING COP
     Y":PRINT
50 INPUT"HIT ENTER WHEN READY";L$
60 OPEN"R",2,B$:FIELD2,255ASNW$,1ASLB$
70 FORX=1TOEF:LSETNW$=S$(X):LSETLB$=S2$(X):PUT2,X:NEXT
80 PRINT"COPY OPERATION IS COMPLETE.":CLOSE:END
```

Program Listing 1

Check our book pages for the latest books about microcomputers.

mention 80 microcomputing when you query an advertiser

ADVENTURES FOR 8K PET® 16K TRS-80® 16K TRS-80® COLOR

ESCAPE FROM MARS—Explore an ancient Martian city while you prepare for your escape.
NUCLEAR SUBMARINE—Fast moving excitement at the bottom of the sea.
PYRAMID—Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.
VAMPIRE CASTLE—A day in old Drac's castle. But it's getting dark outside.
DEATH SHIP—It's a cruise ship—but it ain't the Love Boat and survival is far from certain.
TREK ADVENTURE—Takes place on a familiar starship. Almost as good as being there.

$14.95 each

Aardvark Technical Services ✓129
1690 Bolton
Walled Lake, MI 48088
(313) 669-3110 or (313) 624-6316

PET & TRS-80 are trademarks of Commodore

LOWER CASE & GRAPHIC SYMBOLS GENERATOR KIT FOR TRS-80™ CG 16 ...... $64.50

TRUE 2 LINE DECENDER LOWER CASE. ELECTRONIC SYMBOLS, THIN LINE GRAPHICS, GAME SYMBOLS, TEXTURED BACKGROUNDS AND MANY MORE DEMO CASSETTE IS INCLUDED
FULLY ASSEMBLED WITH DETAIL INSTRUCTION FOR EASY INSTALLATION.
REQUIRES ELECT PENCIL TYPE L.C. MOD OR ORDER MEMORY AND SWITCH KIT. SMK. FOR $18.50

SYNCHRONOUS DATA SEPARATOR FOR DISK USERS. SDS $24.50

THE SYNCHRONOUS DATA SEPARATOR WILL ELIMINATE 100% OF THE SOFT READ ERRORS AND SPEED UP DISK ACCESS TIME BY ELIMINATING RETRIES BY THE DISK CONTROLLER

THE SDS PLUGS INTO THE DISK CONTROLLER'S SOCKET AND HAS ONLY TWO WIRES TO CONNECT. NO TRACES HAVE TO BE CUT. IT IS THE MOST RELIABLE AND ACCURATE DATA SEPARATOR AVAILABLE

ADD $2.50 FOR S&H. CALIF RES ADD 6% SALE TAX
SEND CHECK OR MONEY ORDER TO

G.P. ASSOCIATES ✓203
P.O. BOX 22822, SACRAMENTO, CA 95822 (916) 392-0257

TRS-80 IS A TRADE MARK OF TANDY CORP

Model III Owners

Copy Scripsit and/or Visicalc™

★ Unlimited disk back-ups of YOUR original

★ No programming or technical knowledge required

★ Works if you have already backed up twice

★ Usable only by original purchaser and intended for his private usage

All orders shipped within 10 days
$15.00
No credit cards, please.

SALES DATA, INC. ✓143
BOX 257 BUHLER, KANSAS 67522

```
*!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←%$#!()*
* LLIST                                                        LPRINT.*
* * * * *              SERIAL PRINTER USERS                  * * * * *
* LPRINT                                                        LLIST.*
* > > > >      E N D  O F  S O F T W A R E  H A S S L E S ! !  < < < <*
*              * * * * * * * * * * * * * * * * * * * * * * * *        *
* MAGIC BOX (TM) PLUG IT AND GO! NO SOFTWARE NEEDED TO LPRINT / LLIST.*
* **************                                  ************** *
*  PLUG IN THE PARALLEL PRINTER PORT, COMPATIBLE WITH MOST POPULAR    *
*    SERIAL PRINTERS : TELETYPES, H14, DIABLO, QUME, NEC, ECT. . . .  *
*      USES ALL LEVEL2 ROM PRINTER ROUTINES FOR FULL FLEXIBILITY.     *
* *      SELECTABLE SPEED 55 TO 9600 BAUDS.(SPECIFY WHICH SPEED)    * *
* * *     AUTO LINE FEED AFTER C.R. AUTO CARRIAGE RETURN.         * * *
* * *       STOP BITS, PRINTER HANDSHAKE, 20MA LOOP OR RS232.     * * *
* $ $ $  INTRODUCTORY OFFER  109.00 DOLLARS FOR A LIMITED TIME  $ $ $ *
* * *               NINETY DAYS LIMITED WARRANTY.                 * * *
* *              MR X CONSULTING, INC.  ✓466                        * *
*                                                                     *
*       1-813-973-0517- 602 GLENWOOD DR. E. ZEPHYRHILLS, FLA 33599    *
*       WATCH FOR SERIAL PRINTER TO CPU INTERFACE COMING SOON.        *
*   NO R.S. INTERFACE OR SOFTWARE WILL BE REQUIRED TO LPRINT / LLIST  *
*!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←%$#!()*
```

SOFTWARE YOU CAN BET ON!

STUD5

A five card stud program so realistic you'll swear you're facing six professionals. You decide who's bluffing and who has the cards. Your opponents learn your style and that of the others and modify their betting strategies as the game proceeds. You can define the players' characteristics to match those of your Friday night poker club or let your computer define them randomly.

Only $13.00

TOUT3

A vastly improved version of the original TOUT horse race handicapper. Now, thousands of races are simulated in seconds to give you each horse's probability of winning and identify overlays. Horses are screened using factors derived from an operations research study. This handicapper outperforms programs and calculators selling for much more.

Only $19.00

DRAW5

Jacks or better to open with a joker that can be used as an ace or in straights or flushes (Gardenia CA rules). You play against six computer-controlled opponents. This program has true artificial intelligence capabilities. DRAW5 was developed originally by Real World Simulations. After three years of refining, it's now available for the TRS-80. It will provide hours of fun, challenge and education.

Only $13.00

TRADE IN your old advertised handicapping and poker-playing programs and receive a $6.00 credit. Include original cassette/disk and documentation with your order.

All programs on cassette for Level II 16K (Models I and III). Loading guaranteed. Add $3.50 to get one or more programs on a formatted diskette. Pa. residents add 6% sales tax.

WILSON ✓279
Software Division • 539 Springhouse Lane • Camp Hill, PA 17011

TRS-80 is a Tandy Corporation trademark.

QUALITY PRODUCTS FROM..... THE ALTERNATE SOURCE

SCREENEDIT: Easier to use than the normal EDIT command; more powerful too! To edit a Basic line, LIST it and type EDIT without an argument. You have now activated SCREEN EDIT. Simply typing a letter will change what was there (overwrite), shift left arrow deletes, shift right arrow inserts spaces. Shift down arrow is useful for combining lines. Changing the line number will duplicate that line. Will edit any line 255 characters or less. Enter ends the editing process and Break escapes from editing and cancels all changes. Also allows you to re-execute command lines without retyping them! Don't let the price fool you, this will be a much used utility!!

JUST $9.95

Model I or III — Level II or Disk Basic

THE ALTERNATE SOURCE
1806 Ada Street
Lansing, MI 48910
517-487-3358
✓138 **517-485-0344**

Include $2.00 for crush-proof box and postage, $1.50 for COD, 4% for charge.

✓See List of Advertisers on page 306

Reduce disk read/write errors by calibrating drive speed.

Speedset

Sherman B. Winings
9620 Leeta Cornus Lane
Nokesville, VA 22123

Read/write errors on TRS-80 disk drives are often caused by improper drive speed. Getting the speed reset at Radio Shack means too much down time and trouble to make me happy.

For those bold enough to open the case on the drive (Warning: Opening the case voids any warranties), speed adjustment is rather simple. Both the case and base plate must be removed. Tools needed are a Phillips head screwdriver that fits the case and base plate retaining screws, a jewelers' screwdriver that fits the motor speed control, a bit of clay and a florescent lamp.

First remove the four screws holding the case to the drive. Slip the case from the drive and set it aside. Place the unit upright with the diskette door facing you; the major electronics board will be on your right and the flywheel will be on your left. The motor speed control board, at the rear of the drive between the drive and the power supply, is positioned crosswise to the drive. The power supply is fastened to the back plate.

Place the unit with the flywheel up, diskette removed. Actuate the drive by pushing the reset button on the TRS-80 keyboard. Examine the strobe template on the flywheel under a florescent light. If the 60-cycle pattern appears stationary, the speed is properly adjusted and your read/write errors are caused by other problems than speed control adjustments. On the other hand, if the 60-cycle strobe pattern rotates rapidly or looks like a grey circle, speed adjustment is necessary.

The motor speed control is the rectangular "box" (potentiometer: you gotta admit it looks like a box!) located near the base plate on the flywheel edge of the board. Notice the adjusting screw points towards the base plate. This screw controls the motor speed. Remove the base plate, which is held in place by two screws in the drive unit and two bolts which hold the power supply back plate on.

Once the base plate is removed, the power supply and drive unit have a tendency to separate. Hold the two parts together and place the flywheel side up. Again actuate the drive and adjust the speed control *slowly* until the 60-cycle pattern appears stationary.

Place the drive upside down and loosely fasten the two screws that go into the drive unit. Place a bit of modeling clay (chewing gum will do if you're not particular about aesthetics.) on one of the nuts and put it in place. Start a bolt into it. The nuts come with an attached lock washer so it will hold when tightening. Start the second nut before tightening the first nut. Tighten the drive unit screws first. Now push the power supply and board firmly into the drive unit. Tighten the bolts.

Test the unit for speed before replacing the case. If it passes the strobe test, try the read/write test.

This process is greatly simplified by drilling a 3/4-inch hole in the case to observe the strobe pattern, and by drilling a 1/4-inch hole in the base plate to adjust the speed. The holes should be placed as shown in Figs. 1 and 2. The holes should be dressed with a round file and painted with grey and black paint, respectively.

The next time adjustment is needed, you can simply place the unit on its side, view the strobe through the hole in the case and adjust the speed through the hole in the base plate.

This adjustment and modification will not solve all read/write errors, but if it saves you one trip to Radio Shack, it's worth the time and effort. ■

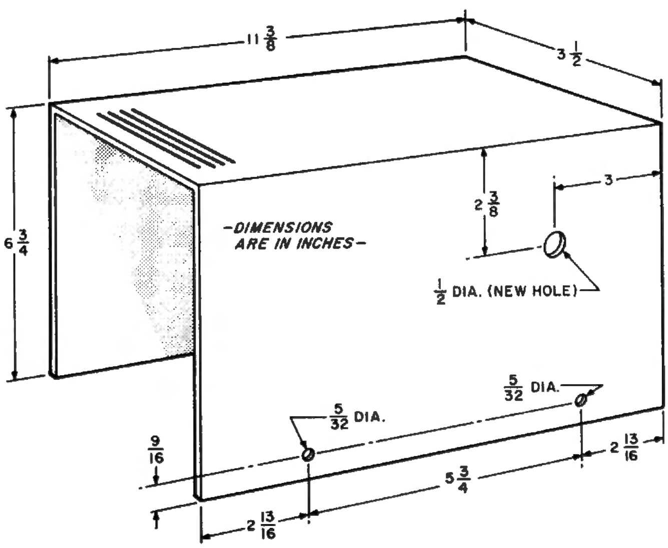


Fig. 1

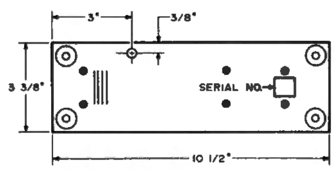
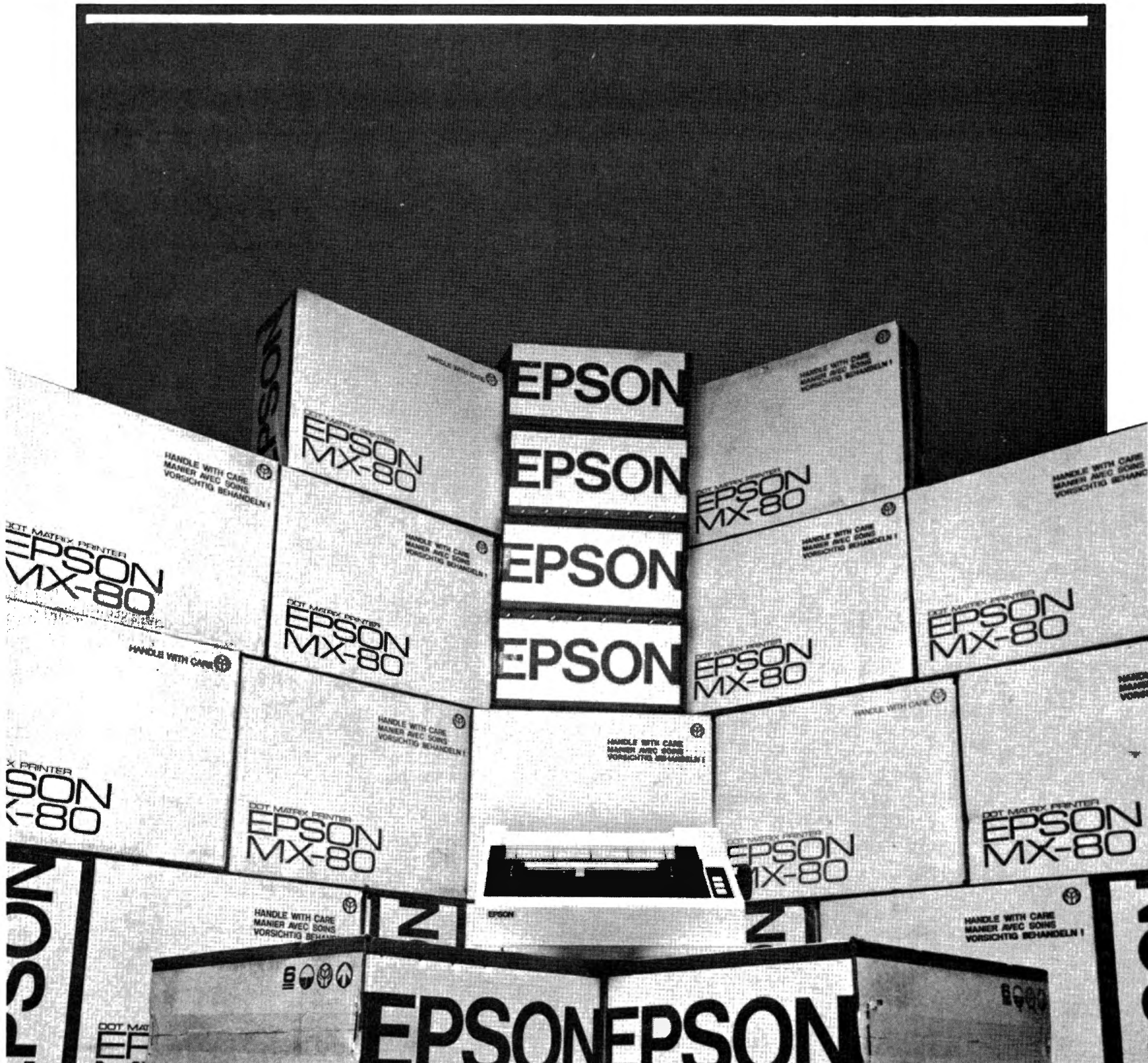


Fig. 2

ALPHA BYTE DOES IT AGAIN...



Call us for our never undersold price.

ALPHA BYTE STORES

69

(213) 883-8594

5115 Douglas Fir Road • Suite B • Calabasas, CA 91302

This short subroutine for shift reversal will change your gears.

Shift Lock

Martin C. Hambel
P.O. Box 515
Lisle, IL 60532

The addition of a lowercase conversion, or a printer capable of lowercase output, to the TRS-80 Level II makes it necessary to reverse the shift of alphabetic character output from the keyboard.

The Level II keyboard driver routine generates uppercase ASCII codes in the unshifted condition, but lowercase codes only when the shift key is pressed.

A common remedy is the inclusion of a shift-reversing function in the driver software, for the output device. This solution, while it requires modifying every output driver routine, still leaves the keyboard output reversed.

Problem

The use of a shift-reversing function within the keyboard interpreting software itself is a much better alternative. It would be convenient to have shift lock from the keyboard.

Level II ROM looks to the keyboard device control block in lower RAM to determine the address of the keyboard driver routine. It is found in memory locations 4016H and 4017H. It is possible to write a new keyboard interpreting routine to substitute for that in ROM.

The disadvantage to this is the length of the program required. Because of memory conservation, this approach is discarded and a scheme to add a patch to the ROM routine substituted.

Disassembly of the Level II driver entry routine reveals an instruction at address 03C2H to PUSH the return address 03DDH onto the stack. The instruction at 03DCH causes a jump to the appropriate driver routine, which returns to the saved 03DDH. The driver routine for the keyboard is the one pointed to by the contents of 4016H and 4017H.

The net effect of all this is a CALL to the routine beginning at the address in 4016H and 4017H.

Solution

Adding a short subroutine with the shift reversal function to the ROM driver is done as follows:

- The device control block address is changed to point to a new driver entry module.
- This module alters the return address of the ROM driver routine from 03DDH to a shift reversing subroutine.
- A short shift reversing subroutine is written, ending in a JUMP to the original 03DDH.

The program, SHFTRV, shown in the Program Listing, is organized in three modules. These correspond to the three steps described.

The first module, LOADER, is executed after the program is loaded. This puts the address of the second module (7FDDH in the Program Listing) into the keyboard device control block. It returns to the operating system (1A19H for a non-DOS system).

Next, KBDMD, called every time a keyboard entry is made, removes the return address of 03DDH from the stack. It replaces it with the start address of the shift reverse module, NWDRV (7FE5H in the Listing).

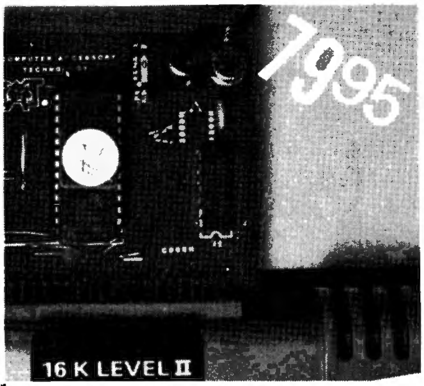
Finally, NWDRV checks for the shift-lock control character. (The shifted down arrow, ASCII code 1AH, is used in the Program Listing but any key or combination not in use, such as the @ with an ASCII code of 40H, can be used if the appropriate change is made to the coding at 7FE6H.)

If the control character is found, the program changes the state of the JUMP instruction at INDIC from conditional to unconditional or vice-versa. Then it proceeds through the shift reversing program steps.

Control, graphic, and numeric characters are returned unchanged, while alphabetic characters are subjected to the shift-reversing operation XOR 20H. Note that when the instruction at INDIC is changed to 18H, the alphabetic characters are returned unchanged.

The program as shown in the Program Listing is assembled into the top 44 bytes of 16K memory. It uses 1AH (shifted down arrow) as the shift lock control character. To use it without modification, prepare a system tape using T-BUG or an assembler program.

The program may be reassembled into any other area of protected memory with appropriate changes to the addresses



KBJ-80 EPROM EXTENDER

Enchance your DISK or Level II basic by installing the KBJ-80. Package includes power supply and firmware module (SYSTAPE). No assembly required!! Firmware module can be used on any kind of EPROM extenders.

(2716-2K) ADDITIONAL FIRMWARE MODULE PRICING:-

SYSTAPE ($39.95)- Adds 9 permanent utilities to Level II basic: system tape Load/Punch and Execute; Repeat keys; Lower case; Abbr. keys; Beeper; Copy and Debounce. (Free with purchase of KBJ-80)

EFL-80 ($59.95)- Used in conjunction with 80-GRAPHIX board provides inverse video w/o using any rams, also includes Debounce; Abbr. keys; Repeat keys and Lower case. A MUST FOR 80-GRAPHIX OWNERS!

KEEPII 3.2 ($39.95)-Latest version from Dennis Kitsz!

TOOLKIT-80 ($39.95)- Add 5 utilities to Level II basic: Renumber; Append; Repeat keys; Debounce and step. You have been programming too long w/o this one!

CURSOR ($24.95)- Full cursor control. You haven't seen any thing like this before!!

IMMEDIATE DELIVERY - CHECK/M.O. TO:
COMPUTER ACCESSORY TECHNOLOGY ✓206
P.O. BOX 2673, KOKOMO, IND. 46901.
TEL:(317)-453-9715
DEALERS INQUIRIES INVITED!!!

"80" SHROUDS

Raise & tilt monitor Cover cabling et...

New...**MDX DEMI-CASE...$32.50**
P.C. Board mounts inside Shroud

...ofessionally ...neered module of ...ally molded fiberglass

Other Models

LNW DEMI-CASE..$32.50
LNW SHROUD.....$27.50
"80" SHROUD.....$24.50

Add $2.50 shipping & handling

N.Y.S. residents add 7% sales tax

✓358

SYRACUSE R&D CENTER
Box 125, Dewitt, N.Y. 13214
"Specializing In Electronic Packaging"

TRS-80™ SOFTWARE

For The Model I & Model III Computers

MACHINE LANGUAGE SOFTWARE

MONITOR #3 .... $39.95
Disassembler; memory displays, memory move, search verify, and modify; read and write object tapes; hexadecimal arithmetic; object code relocator; unload programs for disk; symbolic output tapes; 41-page instruction manual.

MONITOR #4 .... $49.95
Same as Monitor #3 but adds: save and read disk files; direct input and output of disk sectors; send, receive, or talk to another computer via RS-232-C interface; symbolic disassembly on disk.

SMART TERMINAL $69.95
Enables your TRS-80 to be used as a remote terminal to a time-sharing system. Supports lowercase and full range of control keys. Automatic transmission between memory and host computer. Much more.

FASTSORT $9.95
Machine-language sorting program for use by Basic programs. Many times faster than other methods!

BASIC SOFTWARE

MAILING LIST $69.95
Maintains mailing list files of over 1000 names per diskette. Add, delete, change, find name, machine language sort, print file.

SMALL BUSINESS ACCOUNTING $49.95
Based on Dome Bookkeeping Journal #612, keeps track of income, expenditures, and payroll for a small business of up to 16 employees. Daily, monthly, year-to-date summaries.

HOWE SOFTWARE
14 Lexington Road, New City, N.Y. 10956
* TRS-80 is a registered trademark of Tandy Corp ✓103

DISCOVER THE 6809 IN YOUR COLOR COMPUTER

Now you can explore the Radio Shack Color Computer's impressive potentials—as an inexpensive development system, a color peripheral, a process controller—ad infinitum. The Micro Works introduces these powerful software tools for utilizing the color computer at the assembly language level.

MONITOR TAPE: A cassette tape which allows you to:

- Examine or change memory using a formatted hex display
- Save areas of memory to cassette in binary (a "CSAVEM")
- Download/upload data or programs to a host system
- Move the video display page throughout RAM
- Send or receive RS-232 at up to 9600 baud
- Investigate and activate features of your computer, such as hi-res graphics or machine-language music
- Use your color computer as an intelligent peripheral for another computer, a color display or a 6809 program development tool

The monitor has 19 commands in all, and is relocatable and re-entrant.
80C Monitor Tape Price: $29.95

MONITOR ROM: The same program as the monitor tape, supplied on ROM. This allows BASIC to use the entire RAM space. And you don't need to re-load the monitor each time you use it. The ROM plugs into the Extended Basic ROM Socket or a modified ROMPACK.
80C Monitor ROM
Price: $39.95

C BUG IS HERE!!

INSIDE THE COLOR COMPUTER: This package is a disassembler which runs on the color computer and enables you to generate your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. Disassembler features include cross-referencing of variables and labels; output code which can be re-assembled; output to an 80-column printer, small printer or screen; and a data table area specification which defaults to the table boundaries in the interpreter ROM. A 16K system is required for the use of this cassette.
80C Disassembler Price: $49.95

THE MICRO WORKS ✓109

Mastercharge and BankAmericard

P.O. BOX 1110 DEL MAR, CA 92014 714-942-2400 in the Listing. Also, care must be used in selecting an alternate control character, as the driver routine returns the control character in the A register. It will print, if it is a printable character. The 1AH code selected is not recognized by many printers. Another possibility is the shift—down arrow—B combination, which returns 02H.

In the event that other keyboard routines such as debounce are desired, they may be linked into the program by changing the address at 7FE2H to point to the start of the desired additional routine.

Operation

Load the program tape upon power-up, after answering MEMORY SIZE? with 32723. Execute it using /ENTER. Unshifted key entries will then be output as lowercase ASCII codes. Shifted entries will be uppercase. Each time the shifted down-arrow is pressed, the keyboard output will toggle between normal or reverse shift, and vice-versa. ■

```
                 ;      SHFTRV - A PROGRAM TO PERMIT SHIFT-REVERSAL
                 ;               OF THE ALPHABETIC KEYBOARD OUTPUT
                 ;               UNDER SOFTWARE CONTROL.
                 ;
7FD4                       ORG      7FD4H
                 ;
7FD4 21DD7F      LOADER    LD       HL,KBDMD     ;LOAD HL WITH ADDRESS OF
                 ;                               KEYBOARD DRIVER MOD
7FD7 221640                LD       (4016H),HL   ;PUT IT INTO "ROM SWTCHBRD"
7FDA C3CC06                JP       6CCH         ;RET TO BASIC
                 ;                               1A19H FOR LEVEL II BASIC
                 ;
7FDD E1          KBDMD     POP      HL           ;CLEAR ORIG DRVR RET ADDR
                 ;                               (03DDH) FROM STACK
7FDE 21E57F                LD       HL,NWDRV     ;ADDRESS OF SHIFT-REVERSING
                 ;                               SUBROUTINE
7FE1 E5                    PUSH     HL           ;PUT IT ON THE STACK AS THE
                 ;                               NEW DRIVER RET ADDRESS
7FE2 C3E303                JP       03E3H        ;BACK TO ROM DRIVER ROUTINE
                 ;
7FE5 FE1A        NWDRV     CP       1AH          ;SHIFTED DOWN-ARROW?
7FE7 200A                  JR       NZ,SHIFT     ; NO-DO SHIFT REVERSAL
7FE9 F5                    PUSH     AF           ;  YES-SAVE CHAR. TO DSPLY
7FEA 3AF57F      LOCK      LD       A,(INDIC)    ;GET REL JP INSTR (38 OR 18)
                 ;                                FROM SHIFT REVERSAL ROUTINE
7FED EE20                  XOR      20H          ;FLIP-FLOP IT (38/18 OR 18/38)
7FEF 32F57F                LD       (INDIC),A    ;PUT IT BACK
7FF2 F1                    POP      AF           ;RESTORE CHAR TO BE PRTD
7FF3 FE41        SHIFT     CP       41H          ; LESS THAN U.C. "A"?
7FF5 3806        INDIC     JR       C,RET        ;RETURN IF SO, (OR RET ANYWAY
                 ;                                IF ROUTINE CHANGED THIS
                 ;                               ;  INSTR TO "18  06"
                 ;
7FF7 FE7B                  CP       7BH          ; GREATER THAN L.C."Z"?
7FF9 3002                  JR       NC,RET       ;RETURN IF SO
7FFB EE20                  XOR      20H          ;ELSE, CHANGE SHIFT
7FFD C3DD03      RET       JP       03DDH        ;RET TO ORIG ROM RET ADDRESS
7FD4                       END      LOADER       ;WRITE ENTRY-POINT ADDR
00000 TOTAL ERRORS
```

Program Listing

A speedy quick sort utility for your lists of names.

The Spare Time Generator

Thomas C. Mehesan, Jr.
1210 Kahler Court
Sheldon, IA 51201

What do you do while your micro is sorting that membership or mailing list into alphabetical order?

I used to sit and watch the blank screen and wonder just where it was hung up. Or if I should break in to see if all is well.

Well, I got tired of it. So I started looking for a way to build a better lettertrap. Here's what I found.

Common Routines

Several sorting routines are used on a microcomputer. The most common are the Ripple sort, Modified Ripple sort, Bubble sort, and the Shell-Metzner routine. (See "5 Minutes or 5 Hours," by Doyle, *Kilobaud*, May, 1978.)

The fastest is the Shell-Metzner, see Program Listing 1. Using this particular routine you can sort 150 numbers from reverse to ascending order in 36 seconds.

Dealing with string sorts is a different matter, and often 30 minutes is common for sorting a list of 150 names alphabetically.

With Qwik sort, however, the total time to load and sort 150 names for a membership list is *less* than three minutes. On top of that, the screen keeps you informed of just what the CPU is up to.

Program Description

In our example, the membership list is entered and kept on disk in a file buffer. The buffer consists of two sub-records accessed by membership number. The first two fields of the buffer (Program Listing 2, line 150) after the dummy, provides the information, name (NM$) and membership number (NR$). The information allows the sort to be performed and will reaccess the files to print the final list.

Name consists of 20 bytes, last name first—space—first name or initials. Membership number is kept on disk as the two-byte string representation of the number value.

In Quick sort each name is loaded into a sorted 26 × 30 array (B$(26,30) using the first letter of the last name to index "row". You can index up to 30 names in each row, each starting with the same letter. This will accomodate a list of 200 names. For a larger list, simply increase the size of the array and its companion (A(26,30)), which holds the related membership number.

After the arrays are built, a Shell-Metzner routine sorts only the entries in each row of array B$(26,30) but switches both arrays simultaneously. This reduces the number of switches necessary in sorting by a factor of perhaps 10X.

When each row is sorted, the final results are posted from the companion array to a master list of membership numbers (L(N)). The printing information is drawn from this. Some additional memory could be saved by eliminating the master list and printing from the companion array.

In any event, now you have a program which allows you to sort a list of 150 names in reasonable time. It uses only 12243 bytes to load the program and execute it. (It takes 32 bytes for each additional name.) This program can perform an in-memory sort of more than 250 names with only 16K memory or more than 425 names with 32K and a disk.

Program Analysis

Now to the fun stuff.

Lines 50-70 inform the hardware of the number of names that must be sorted. To do this simply go to LOF(1) and see if both halves are occupied. Remember, the total number of names is either the second sub-record (NR$) or, if that value is zero, 1 + (2*(LDF(1) − 1)). If you're using a cassette, just place your name and number files into B$(26,30) and A(26,30).

When using cassette or sequential disk input, you may have to set up other companion arrays to handle final output to the print routines. It may be impractical to go back to your files for other information about

Program Listing 1. Shell-Metzner Sort

```
250 ' ***** SHELL-METZNER SORT ROUTINE *****
260 M=R(Q)-1
270 M=INT(M/2)
280 IF M=0 THEN 420
290 J=1:K=(R(Q)-1)-M
300 I=J
310 L=I+M
320 IF B$(Q,I)<=B$(Q,L) THEN 380
330 T$=B$(Q,I):B$(Q,I)=B$(Q,L):B$(Q,L)=T$
340 G=A(Q,I):A(Q,I)=A(Q,L):A(Q,L)=G
350 I=I-M
360 IF I<1 THEN 380
370 GOTO310
380 J=J+1
390 IF J>K THEN 270
400 GOTO 300
```

Program Listing 2. Qwik Sort

```
10 ' ***** ALPHABETICAL SORTING PROGRAM *****
20 CLS:CLEAR 5000
30 PRINT:PRINTTAB(20)"ALPHABETICAL MEMBERSHIP LIST"
```

Program continues

```
40 PRINT:INPUT"PRESS <ENTER> TO START SORT";Z
50 ' ***** DETERMINE NUMBER OF NAMES TO SORT *****
60 OPEN "R",1,"MLIST :1":GET 1,LOF(1):FIELD 1,128 AS DM
   $,2 AS NR$
70 IF CVI(NR$)<1 THEN N1=1+(2*(LOF(1)-1)) ELSE N1=CVI(N
   R$)
80 DEFINT L, A
90 ' ***** DIMENSION ARRAYS *****
100 DIM A(26,30), L(N1), R(30), B$(26,30)
110 ' ***** LOAD NAMES AND MEMBERSHIP NUMBERS TO ARRAYS
    *****
120 CLS:PRINTTAB(10)"WORKING":FOR J=1 TO 26:R(J)=1:NEXT
    J
130 PRINT"RETRIEVING DATA #"
140 FOR N=1 TO N1:PRINT @ 81, N:PR%=INT((N-1)/2)+1:SR%=
    N-(2*(PR%-1))
150 GET 1,PR%:FIELD 1, ((SR%-1)*128) AS DM$, 2 AS NR$,
    20 AS NM$
160 J=ASC(LEFT$(NM$,1))-64
170 A(J,R(J))=CVI(NR$):B$(J,R(J))=NM$:R(J)=R(J)+1:GOTO1
    80
180 NEXT N:J=0:N=0:R=0:CLOSE
190 ' ***** SORTING ARRAY ELEMENTS *****
200 PRINT:PRINT:PRINT"SORTING  `"
210 P=0
220 FOR Q=1 TO 26:PRINT @ 267, CHR$(Q+64)
230 IF A(Q,1)=0 GOTO450
240 IF A(Q,2)=0 THEN P=P+1:L(P)=A(Q,1):GOTO 450
250 ' ***** SHELL-METZNER SORT ROUTINE *****
260 M=R(Q)-1
270 M=INT(M/2)
280 IF M=0 THEN 420
290 J=1:K=(R(Q)-1)-M
300 I=J
310 L=I+M
320 IF B$(Q,I)<=B$(Q,L) THEN 380
330 T$=B$(Q,I):B$(Q,I)=B$(Q,L):B$(Q,L)=T$
340 G=A(Q,I):A(Q,I)=A(Q,L):A(Q,L)=G
350 I=I-M
360 IF I<1 THEN 380
370 GOTO310
380 J=J+1
390 IF J>K THEN 270
400 GOTO 300

410 ' ***** LIST SORTED NAMES TO PRINTING ARRAY *****
420 FOR N=1 TO R(Q)-1
430  P=P+1:L(P)=A(Q,N)
440 NEXT N

450 NEXT Q
460 GOTO 630
470 END
480 ' ***** SCREEN PRINT SORTED LIST *****
490 CLS:PRINTTAB(20)"MEMBERSHIP LIST"
500 PRINTTAB(10)"NAME";TAB(28)"NUMBER";TAB(41)"TYPE";TA
    B(50)"PHONE"
510 OPEN"R",1,"MLIST":Q=0
520 FOR N=1 TO N1
530 IF L(N)=0 GOTO 600
540 PR%=INT((L(N)-1)/2)+1:SR%=L(N)-(2*(PR%-1))
550 GET 1,PR%:FIELD 1,((SR%-1)*128) AS DM$, 2 AS NR$, 2
    0 AS NM$,
66 AS NU$, 40 AS RM$
560 PRINTTAB(5)NM$;TAB(30);USING"###";CVI(NR$);:PRINTTA
    B(40)LEFT$(RM$,18)
570 Q=Q+1
580 FOR J=12 TO N1 STEP 15:IF J=Q INPUT"PRESS <ENTER> F
    OR MORE";Z
590 NEXT J
600 NEXT N:CLOSE
610 INPUT"PRESS <ENTER> TO CONTINUE";Z
620 ' ***** SELECT PRINTING MODE *****
630 CLS:PRINT:PRINT:PRINT"SELECT AS FOLLOWS:"
640 PRINTTAB(10)"1. REVIEW LIST":PRINTTAB(10)"2. PRINT
    HARD COPY"
650 PRINTTAB(10)"3. END
655 INPUT Z
660 IF Z<>1 AND Z<>2 AND Z<>3 GOTO 630
670 ON Z GOTO 490  ,680  ,840
680 CLS:PRINT"MAKE SURE LINE PRINTER IS ON AND SET TO P
    RINT"
690 PRINT:INPUT"ENTER TODAY'S DATE (##/##/##): ";DA$
700 PRINT:INPUT" PRESS <ENTER> TO PRINT";Z
710 LPRINTTAB(30)"NAME OF LIST"
720 LPRINTTAB(27)"MEMBERSHIP LIST -- ";DA$
730 LPRINT,CHR$(10):LPRINTSTRING$(75,"=")
740 LPRINTTAB(15)"NAME";TAB(38)"NUMBER";TAB(47)"TYPE";T
    AB(57)"PHONE"
750 LPRINTSTRING$(75,"-")
760 OPEN"R",1,"MLIST"
770 FOR N=1 TO N1:J=L(N)
780 PR%=INT((J-1)/2)+1:SR%=J-(2*(PR%-1))
790 GET 1,PR%:FIELD 1,((SR%-1)*128) AS DM$, 2 AS NR$, 2
    0 AS NM$,
66 AS NU$, 40 AS RM$
800 LPRINTTAB(10)NM$;TAB(40);USING"###";CVI(NR$);:LPRIN
    TTAB(46)RM$
810 NEXT N:CLOSE
820 GOTO630
830 END
840 STOP:' *** OR RETURN TO MENU ***
```

each name.

Lines 140-180 load the two arrays, B$(26,30) and A(26,30), from disk, sorting by the first letter as they go. The membership number is displayed on the screen as each file is loaded. The formula J = ASC(LEFT$(NM$,1))-64 performs this. We all know that the ASCII value for A is 65 and values increment up by 1s from there. (When sorting lowercase strings substitute 97 for 64 in the formula. Remember, the names are on disk by last name first.) Already you have sorted the entire list by first letter as you load it, and the whole process took only 42 seconds!

After everything is neatly in the arrays, it's important to find out how many entries there are in each row. No need to sort one entry, is there? We could look at the pointer (R(J)) you created in line 120 (to tell the CPU where to put the next entry in the row). It seems simpler, however, to look at the first and second entries in lines 230 and 240. If there is no entry, (no name with that first letter in the list) skip to the next row. If there is only one entry, shown by the second entry equaling zero, post this entry directly to the master list. No sort is necessary. The pointer for the master list (P(N)) at lines 210, 240 and 430 will put this posting in proper sequence.

If you fail the tests in lines 230 and 240, go right into the Shell-Metzner sorting routine, lines 200-400. Notice that at lines 200 and 220 there is some screen activity indicating which letter the CPU is handling. This tells you the program hasn't crashed.

After sorting each row, the rearranged entries in A(26,30) are posted to the master list at lines 400-440. At line 450 you can jump back to start the process for the next letter.

When you hit line 460, you are done sorting. Surprise... only 2 minutes and 42 seconds have elapsed.

Your biggest problem, after you print out your nice new list (lines 490 to end), is to figure what you're going to do with all that spare time.■

Hardware to speed up cassette I/O.

The FCI-80 and the TC-8

TC-8
JPC Products, Inc.
Albuquerque, NM
$120

FCI-80
Personal Micro Computers, Inc.
Mt. View, CA
$188

by Harley Dyk
1644 Grant
Grand Haven, MI 49417

Owners of cassette based TRS-80 computers have four mass-storage options: Disks, tape units, speed-up devices for cassette, and simple cassettes. The advantages of disk have been fairly well publicized; the May issue of *80 Microcomputing* ran an article on the Stringy Floppy and Beta-80, and there is little to write about the slow and not too reliable TRS-80 Cassette Operating System (COS). This article looks at two ways to speed up tape storage and make it more reliable. The products under consideration are the FCI-80 by Personal Micro Computers Inc., and the TC-8 by JPC Products, Inc.

FCI-80

The FCI-80 takes a novel approach to increasing cassette load speed. Rather than increasing the density of tape data, the tape is moved faster. Programs are saved as usual at the normal speed, but when one wants to load a program using the FCI-80, fast forward and play are pressed simultaneously on a modified CTR-41 recorder. This increases load speed to a minimum of 6000 baud (about 12 times the conventional load speed).

The FCI-80 connects to the keyboard with a 40-conductor ribbon cable, connects to a 9V AC power adapter, and plugs into the earjack on the recorder. The unit is about the size of the recorder and is designed to be placed under the recorder.

The operating system is in ROM at address 12288 (3000 H), an area just above Level II BASIC called reserved. After turning the computer on, simply type SYSTEM followed by /12288 to make the FCI-80 functional. To load a conventionally saved program, type LOAD. This clears the screen and displays the question: BASIC(B) OR SYSTEM(S)? After answering, you can specify a program name, and finally the message PRESS PLAY & FFWD ON THE TAPE DECK is displayed.

Volume must be at maximum; the break key will get you back to BASIC at any time. Since fast forward and play must work together, the recorder must be modified. You can purchase such a recorder, or full instructions come with the FCI-80 to modify your CTR-41.

Loading speed is almost unbelievable for an audio cassette. The speed is not fixed since the fast forward speed of a tape increases as the size of the take-up reel increases. I was not able to get the unit to work near the end of a 300 ft. tape, but it worked fine from the beginning through at least the 100 ft. point. This means you should be able to use tapes with at least 10 minutes per side.

At its slowest, the FCI-80 loaded a 14K program in 16.5 seconds. The program took 228 seconds with TRS-80 COS, a speed-up factor of more than 13. I also tried a 4.3K program at the beginning of a tape and it loaded in about 5.7 seconds. I recorded the same program at about the 100 ft. point of the tape and was able to load the program in 3.9 seconds, which is nearly 9000 baud (a speed-up factor of nearly 18).

Due to the speed of the FCI-80, cassette head alignment is very important. I was not initially able to get tapes I had recorded or purchased programs to load. I rerecorded these programs on the modified CTR-41 and then they loaded without problem. I could have elected to try to match the head alignment to my tapes.

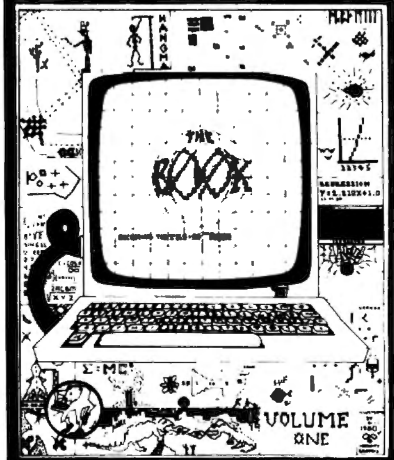
The unit is ideal for someone who frequently uses large programs but rarely saves them. The unit does load programs rapidly, but I feel two very important features are missing. Programs must be saved at the same slow speed, and writing data to tape (one of the weakest features of TRS-80 COS) has not been changed.

One nice additional feature of the FCI-80 is that a keyboard debounce program is also in ROM at address 12800 (3200 H). This program also provides automatic key repeat when a key is held down, and will beep when you press a key if the cassette output plug is connected to an amplifier.

	TC-8	FCI-80
Cost	$90 kit, $120 assembled, $3.50 for postage and handling	$188, plus $25 for cable plus $8 for power pack, add $95 for modified CTR-41 if needed. Prices include shipping.
Size: W × L × H	4" × 5.5" × 2.75"	5.25" × 10.25" × 2"
Operating System	In RAM—takes about 1.3 k	In ROM at 12288 (3000 H)
Warranty	60 day	90 day, 30 day money back
Keyboard Debounce	Yes	Yes
Contact	JPC Products Inc. Albuquerque, NM 87112	Personal Micro Computers Inc. Mt. View, CA 94043
Speed	2400 baud	6000-9000 baud load only

Table 1. Comparison table.

THE BØØK



If you ever do Assembly language programming, or you just want to know more about your TRS-80 ROM, "THE BOOK"s are for you.

*TRS-80 is a trademark of Tandy Corp.

Each volume is priced at: $14.95 + $1.50 S&H = $16.45 ($17.05 in VA) Overseas add $2.50 for air shipment

Insiders Software Consultants, Inc. ✓305
P.O. Box 2441, Dept. SUM2
Springfield, VA 22152

☐ **Please send me Volume I of THE BOOK**

☐ **Please send me Volume II of THE BOOK**

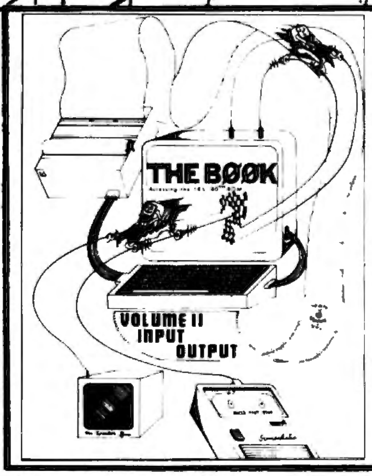
NAME ______________________
ADDRESS ______________________
CITY, STATE ______________ ZIP CODE ________

[ ] Check payable to Insiders Software Consultants, Inc.
[ ] MASTER CHARGE MC Bank Code
[ ] VISA Exp. Date ________ Card Number ________

Signature ______________________

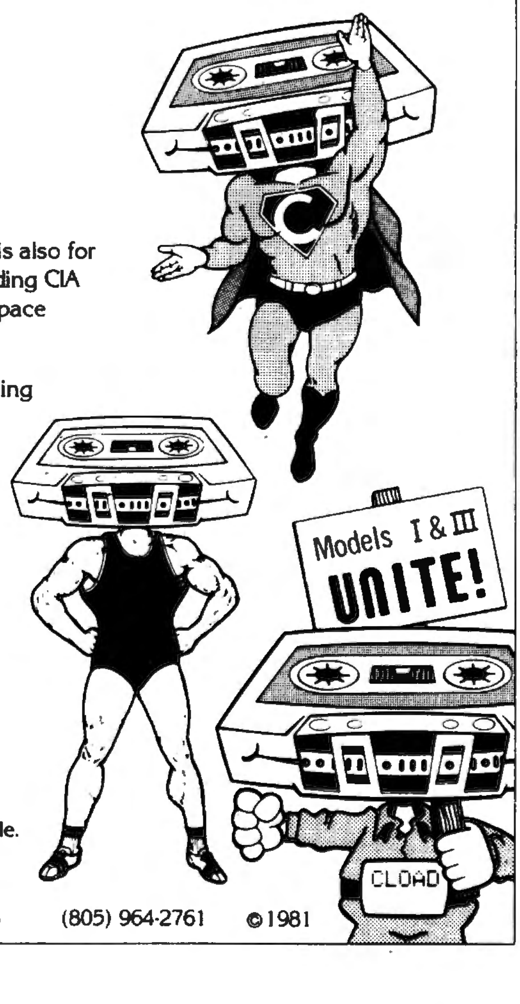
Dealer Inquiries Invited

Volume I: The most comprehensive book yet on the math routines in Level II ROM, Models I & III. Includes a fully commented listing from 0708H to 1607H and an incredibly complete map of the ROM and reserved areas of RAM.



Volume II: Everything you want to know about the video, keyboard, cassette, and printer driver routines. Learn how to write your own! Remarkably detailed listings illustrate well-commented source code. Complement Volume I, now.

WE'RE EXCITED . . .



CLOAD MAGAZINE, the cassette magazine for your **TRS-80** Model I, is also for your TRS-80 Model III! Over 90% of the monthly offerings on cassette (including CIA Adventure, Hustle, States Quiz, RAM Test, and Alien — a machine language space invader-type game) will load and run on the Model III directly.

Now you Model I'ers and Model III'ers can both spend your time RUNning programs, rather than ENTERing and EDITing them. Each tape contains 6 to 8 ready-to-**CLOAD** fun, educational, and practical programs, delivered to your door by First Class Mail each month.

Join Clyde's march for reduced keystrokes, and Exercise your computer with Super, ready-to-load programs.

Get **CLOAD MAGAZINE** and get excited too!

The Fine Print:
Overseas rates slightly higher—please write for them.
Back issues available—ask for our list.*
TRS-80 is a trademark of Tandy Corporation
California residents add 6% to single copies and anthologies. Programs are for Level II 16K, Model III 16K, and occasionally for disks.
*24 Level I back issues also available.
Mastercard/Visa Welcome.

PRICES

1 year subscription	$42.00
6 month subscription	$23.00
Single copies	$4.50
Anthology-volume 1	$10.00
Anthology-volume 2	$15.00

CLOAD ✓32 MAGAZINE INC. P.O. Box 1267 Goleta, CA 93116 (805) 964-2761 ©1981

TC-8

The TC-8 has been dubbed the "poor man's floppy," and the name fits. The unit saves and loads programs almost five times as fast as conventional cassette, offers a significant improvement in raw data storage, and improves reliability.

The TC-8 connects to the keyboard with a 40-conductor ribbon cable, connects to a 9V AC power adapter, and to the recorder. The operating system (called Util) is loaded from TRS-80 COS tape so the recorder is initially connected to the keyboard. Memory size must be set to 31400 on power up, and the operating system (which occupies about 1.3K) must be loaded at the conventional speed. You can then plug the recorder into the TC-8 and proceed (save & load) at 2400 baud. The results are what one would expect at this speed: A 4.3K program loads in 15 seconds, and a 14K program in 48 seconds.

The TC-8 uses a modified Manchester pulse code (phase encoded) and claims less than one error per million bits. The unit works well with Radio Shack recorders but I found that it would not work with *all* recorders.

The TC-8 has a very smooth operating system and excellent documentation. You can use up to eight characters for program names, and 11 commands are provided, most for saving, loading, and verifying BASIC and machine language programs. There is also a directory command which displays the names of programs as they are encountered on a tape. The KILL command gets rid of Util so that a large program that has been loaded can be run.

The syntax for saving raw data on the TC-8 is: PRINT #1,X. The only difference from standard cassette syntax is that TRS-80 COS uses # – 1. Using that statement in a loop, it took 428 seconds to store 100 three-digit random numbers with the TRS-80 COS. The TC-8 performed the task in 22 seconds, almost 20 times faster! Although the speed of TRS-80 COS data storage can be improved by placing several variables in the same PRINT # – 1 statement, this is not always convenient or even possible in a given program.

The TC-8 software includes a monitor called Tiny. The monitor provides a means to find the beginning, end, and execution addresses of most machine language programs. Tiny also permits you to examine memory locations, relocate Util (for 32K or 48K systems), write backup copies of Util, and jump to any hex address.

The TC-8 can be purchased in kit form. The instructions for the kit are superb and the 11 resistors, nine capacitors, three diodes, two transistors, five ICs, one regulator, plus miscellaneous hardware, can be assembled in about one hour. Assuming all parts are good, you can be up and running for a mere $90.

The FCI-80 and TC-8 are certainly worth considering if you want to improve your cassette storage without investing a major amount. ■

✓See List of Advertisers on page 306

A program-packing utility for those with short memories.

Comprs

Stephen Barker
7700 Parkway Dr. #44
La Mesa, CA 92041

Comprs is a machine language utility program which will remove REM statements and extra spaces from BASIC programs. This allows you to fully document your file copy of a program and then compress it into a smaller work copy. Spaces within quotes will not be removed. Warning: This program not only removes the REM statement but the entire line. Therefore, any references to that line must be changed.

I have included two versions of this program. The first is a source listing for the machine language version. This can be assembled using the Radio Shack Editor/Assembler. The second version is a BASIC listing in which the machine language version has been broken into data statements. This program will POKE the machine language version into memory.

If you choose to use the BASIC version, write a small loop to add up the sum of all the data statements. The total should be 23,416. Use this to cross-check the numbers you entered.

Instructions

Machine Language: Enter the BASIC program to be compressed by CLOADing. Enter Comprs using the SYSTEM command. Execute the program by typing / and Enter. When READY is shown on the screen, the program has been compressed.

BASIC: Enter and run the BASIC version of Comprs. CLOAD the program to be compressed. Type SYSTEM and Enter. When ? appears on the screen type /32512 and Enter. When READY is shown on the screen, the program has been compressed. ■

Program Listing 2

```
100 'THIS IS A BASIC VERSION OF COMPRESS. THIS WILL POK
    E
110 'THE MACHINE LANGUAGE PROGRAM INTO MEMORY.
120 FORX=32512TO32715:READA:POKEX,A:NEXT:END
130 DATA33,233,66,34,254,126,33,253,126,54,0,42,254,126
    ,35
140 DATA35,35,35,126,254,147,40,14,254,34,40,73,254,32,
    40,90
150 DATA254,0,40,41,24,236,42,254,126,35,35,35,35,1,4,0
    ,35
160 DATA3,126,254,0,32,249,3,35,229,42,249,64,237,91,25
    4,126
170 DATA175,237,82,229,193,225,237,176,42,254,126,43,35
    ,229
180 DATA126,254,0,32,6,35,126,254,0,40,60,225,34,254,12
    6,33
190 DATA253,126,54,0,24,167,58,253,126,254,0,40,7,62,0,
    50,253
200 DATA126,24,159,62,1,50,253,126,24,152,58,253,126,25
    4,1
210 DATA40,145,229,229,229,42,249,64,209,175,237,82,229
    ,193
220 DATA225,35,237,176,225,43,195,17,127,33,233,66,34,2
    54,126
230 DATA42,254,126,126,254,0,32,13,35,126,254,0,32,7,35
    ,34
240 DATA249,64,195,25,26,42,254,126,35,35,35,35,126,254
    ,0,32
250 DATA250,35,229,229,193,42,254,126,113,35,112,225,34
    ,254
260 DATA126,24,207
```

Program Listing 1

```
                         ;COMPRS IS A PROGRAM TO REMOVE REM STATEMENTS AND
                         ;SPACES FROM BASIC PROGRAMS
7F00                     ORG     7F00H
7F00             START   EQU     $
7EFE             REF     EQU     START-2         ;REFERENCE POINTER
7EFD             QTE     EQU     START-3         ;QUOTE FLAG
7F00  21E942             LD      HL,42E9H        ;BEGINNING OF BASIC
7F03  22FE7E             LD      (REF),HL        ;SET POINTER
7F06  21FD7E             LD      HL,QTE          ;RESET FLAG
7F09  3600               LD      (HL),0
7F0B  2AFE7E     SEARCH  LD      HL,(REF)        ;SEARCH FOR KEY CODES
7F0E  23                 INC     HL              ;SKIP 4 BYTES
7F0F  23                 INC     HL
7F10  23                 INC     HL
7F11  23         NXT     INC     HL
7F12  7E                 LD      A,(HL)
7F13  FE93               CP      93H             ;REM?
7F15  280E               JR      Z,REM
7F17  FE22               CP      22H             ;QUOTES?
7F19  2849               JR      Z,QUOTE
7F1B  FE20               CP      20H             ;SPACE?
7F1D  285A               JR      Z,SPACE
7F1F  FE00               CP      0               ;END OF LINE?
```

Program Continues

```
7F21 2829               JR      Z,ENTER
7F23 18EC               JR      NXT
7F25 2AFE7E      REM    LD      HL,(REF)       ;THIS DELETES REM LINE
7F28 23                 INC     HL             ;SKIP 4 BYTES
7F29 23                 INC     HL
7F2A 23                 INC     HL
7F2B 23                 INC     HL
7F2C 010400             LD      BC,4           ;BYTE COUNTER
7F2F 23          R1     INC     HL
7F30 03                 INC     BC
7F31 7E                 LD      A,(HL)
7F32 FE00               CP      0              ;END OF LINE?
7F34 20F9               JR      NZ,R1
7F36 03                 INC     BC             ;ADD BYTE TO COUNTER
7F37 23                 INC     HL
7F38 E5                 PUSH    HL
7F39 2AF940             LD      HL,(40F9H)     ;END OF PROGRAM
7F3C ED5BFE7E           LD      DE,(REF)       ;POINTER
7F40 AF                 XOR     A
7F41 ED52               SBC     HL,DE          ;# OF BYTES TO BE MOVED
7F43 E5                 PUSH    HL
7F44 C1                 POP     BC             ;# 0F BYTES DELETED
7F45 E1                 POP     HL
7F46 EDB0               LDIR                   ;DELETE LINE
7F48 2AFE7E             LD      HL,(REF)
7F4B 2B                 DEC     HL
7F4C 23          ENTER  INC     HL             ;LAST LINE?
7F4D E5                 PUSH    HL
7F4E 7E                 LD      A,(HL)
7F4F FE00               CP      0
7F51 2006               JR      NZ,E1          ;CANNOT BE LAST LINE
7F53 23                 INC     HL
7F54 7E                 LD      A,(HL)
7F55 FE00               CP      0
7F57 283C               JR      Z,POINT        ;END OF PROGRAM
7F59 E1          E1     POP     HL
7F5A 22FE7E             LD      (REF),HL
7F5D 21FD7E             LD      HL,QTE
7F60 3600               LD      (HL),0         ;CLEAR QUOTE FLAG
7F62 18A7               JR      SEARCH
7F64 3AFD7E      QUOTE  LD      A,(QTE)        ;THIS CHECKS TO SEE IF
7F67 FE00               CP      0              ;QUOTE FLAG IS SET
7F69 2807               JR      Z,Q1
7F6B 3E00               LD      A,0            ;RESET FLAG
7F6D 32FD7E             LD      (QTE),A
7F70 189F               JR      NXT
7F72 3E01        Q1     LD      A,1            ;SET FLAG
7F74 32FD7E             LD      (QTE),A
7F77 1898               JR      NXT
7F79 3AFD7E      SPACE  LD      A,(QTE)        ;THIS REMOVES SPACE IF
7F7C FE01               CP      1              ;NOT WITHIN QUOTES
7F7E 2891               JR      Z,NXT          ;QUOTES ARE SET
7F80 E5                 PUSH    HL
7F81 E5                 PUSH    HL
7F82 E5                 PUSH    HL
7F83 2AF940             LD      HL,(40F9H)     ;END OF PROGRAM
7F86 D1                 POP     DE
7F87 AF                 XOR     A
7F88 ED52               SBC     HL,DE          ;# OF BYTES TO BE MOVED
7F8A E5                 PUSH    HL
7F8B C1                 POP     BC
7F8C E1                 POP     HL
7F8D 23                 INC     HL
7F8E EDB0               LDIR                   ;REMOVE SPACE
7F90 E1                 POP     HL
7F91 2B                 DEC     HL
7F92 C3117F             JP      NXT
7F95 21E942      POINT  LD      HL,42E9H       ;THIS RESETS ALL LINE
7F98 22FE7E             LD      (REF),HL       ;POINTERS
7F9B 2AFE7E      P1     LD      HL,(REF)
7F9E 7E                 LD      A,(HL)
7F9F FE00               CP      0              ;LAST LINE?
7FA1 200D               JR      NZ,P2
7FA3 23                 INC     HL
7FA4 7E                 LD      A,(HL)
7FA5 FE00               CP      0
7FA7 2007               JR      NZ,P2
7FA9 23                 INC     HL
7FAA 22F940             LD      (40F9H),HL     ;SET END OF PROGRAM
7FAD C3191A             JP      1A19H          ;RETURN TO BASIC
7FB0 2AFE7E      P2     LD      HL,(REF)
7FB3 23                 INC     HL             ;SKIP 4 BYTES
7FB4 23                 INC     HL
7FB5 23                 INC     HL
7FB6 23          P3     INC     HL
7FB7 7E                 LD      A,(HL)
7FB8 FE00               CP      0              ;END OF LINE?
7FBA 20FA               JR      NZ,P3
7FBC 23                 INC     HL
7FBD E5                 PUSH    HL
7FBE E5                 PUSH    HL
7FBF C1                 POP     BC
7FC0 2AFE7E             LD      HL,(REF)
7FC3 71                 LD      (HL),C         ;SET POINTER LSB
7FC4 23                 INC     HL
7FC5 70                 LD      (HL),B         ;SET POINTER MSB
7FC6 E1                 POP     HL
7FC7 22FE7E             LD      (REF),HL       ;SET REFERENCE
7FCA 18CF               JR      P1
7F00                    END     START
00000 TOTAL ERRORS
```

This time-saving gambit allows you to save, then reload, ongoing Sargon struggles.

The Sargon Saver

Thomas L. Quindry
TLQ Enterprises
6237 Windward Dr.
Burke, VA 22015

Ever wish you could play Sargon at its highest level of play without tieing up your computer for weeks at a time? Sargon I takes up to 16 hours to analyze a move at its top level of play. Sargon II, though faster, takes between four and seven hours at its top level.

While this game is in progress your computer is not available for any other use. You could power up Sargon every day and set up the current board position, but that doesn't sound too promising. After all, it's a tough job to manually set up the game position, let alone remembering it from day to day.

Computer Set-up

It doesn't have to be a time-consuming job. You bought your computer to free you from slavery. All you have to do is tell your computer how to set up the board position.

The following are programs that modify the cassette-loading versions of Sargon I and II. Once modified, Sargon will save the current board position to tape, along with the move number, or load a saved tape with the above information to restore the game.

Program Listing 1 is the Sargon I patch program, which I call MODIA. The Sargon II patch program called MODIIA is Program Listing 2. When your Sargon program is modified by the appropriate patch, it will generate a data tape which can be later loaded into either Sargon I or II. If you currently have Sargon I and plan to get Sargon II, you can save important Sargon I game positions for later strategy planning with Sargon II and vice versa.

Loading the patch programs is simple. For Sargon I, load the game program, and when you receive the prompt, load MODIA. At the second prompt, enter / or /28922. Loading the patch for Sargon II is a bit more complicated. Sargon II normally loads in two parts; with the patch, it must be loaded in three. Load the short loader program as you normally would. Then, remove the game tape from your cassette recorder and load MODIIA. After MODIIA is loaded, replace the Sargon II tape without rewinding, hit / and enter and the final part of Sargon II will load.

The patch programs interrupt Sargon in subtle ways (which have no effect on its chess playing abilities) to provide I/O capabilities via cassette tape. Messages have been added to prompt you: With Sargon I, answer the questions with a yes or no. With Sargon II, hit S to save the current board position, L to load a previous board position, and G to start a new game or C to change the board.

The MODIIA patch starts at address 44A0H (line 150). This address interrupts the Sargon II loader program just before its jump into the main part of the chess program, enabling our modifications to be made before we lose command control of the program.

Lines 160 to 190 make changes to the title. Lines 200 to 250 add prompting messages to allow you to save or load the board position and move number. Since the move number is used in the decision making process of Sargon II, it is important to have it.

Lines 260 to 320 provide the patches to the main program and then return control to Sargon II. The proper messages are displayed via the patch address to start in lines 330 to 390. Control is then returned to Sargon II in line 400. Conditional statements for the load and save routines are accessed by the patch to COND in line 410. If no key is entered or any letter other than an S or L is entered, the program returns to Sargon II in line 440. Otherwise, the program jumps to the load (lines 450 to 570) or save (lines 580 to 700) routines and then back to Sargon II at the proper locations. The ready subroutine gives the READY CASSETTE message and sets registers for the load and save routines.

When playing Sargon I or II, you may save a board position and then continue playing where you left off by giving the command to analyze a position or change board. After loading a game position, you are prompted to tell what color you are playing and, in Sargon II, the move number. Since the move number has been loaded into memory, you must hit enter twice to preserve it.

The MODIA format is different, since I tried to preserve the general format of operation used in each game program. In lines 160 to 200 of Listing 1 is a patch to correct the need for

White Original Position	White After Moving	Piece	Black Original Position	Black After Moving
01	09	Pawn	81	89
02	0A	Knight	82	8A
03	0B	Bishop	83	8B
04	0C	Rook	84	8C
05	0D	Queen	85	8D
06	0E	King	86	8E
00		Empty Square		

Table 1. Game position notation for Sargon I and II (Given in hexidecimal notation)

GET ORGANIZED

With These Utility Programs From Instant Software

FIND IT QUICK

Put an end to the Misplaced Information Syndrome! Here is a reliable, fast, subject-oriented information locator—FIND IT QUICK! This information indexing and retrieval system is versatile and flexible—the ideal tool for doctors, lawyers, engineers, businessmen, educators, anyone who needs to locate information.

Designed with a journal/magazine format, FIQ is versatile enough to be used to store and retrieve any type of reference information including client lists, vendor advertisements, report bibliographies and more—even poems and famous quotations. Subjects can be indexed or called according to author, journal and date or by up to nine keywords that you determine. Keywords can be used singly or in combination during the search routines.

FIQ can be used by anyone. It requires no special skill, just the ability to type. Article references to be stored can be preselected and then entered into the system by a secretary or assistant. The contents of entire file cabinets can even be indexed and stored!

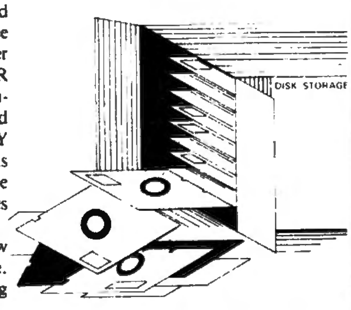
FIND IT QUICK requires a minimum system consisting of a TRS-80 Model I Level II with 16K of RAM, an Expansion Interface with 16K of RAM and at least one disk drive. Up to 1080 items can be stored on one data disk. Instant Software's Tiny DOS operating system is included on the program disk so that you can use the program without fuss or bother.

Make your computer work like your own personal librarian with FIND IT QUICK. Can you afford to wait another day?

Order No. 0258RD $49.95. Disk

MASTER DIRECTORY

The MASTER DIRECTORY is a disk file storage program that reads the files on all your disks and stores the file names and extensions and even records the free space on each disk. All you have to do is number the disks in your library and the MASTER DIRECTORY will keep track of their contents. You can read the names, displayed alphabetically, search the DIRECTORY for file names and extensions, delete disks and search for free space. You can store 5000 files or 320 disks, whichever comes first.



Your disk storage problems are over now that the MASTER DIRECTORY is here. This package requires the following minimum system:

1. A TRS-80 Level II with 16K RAM.
2. An Expansion Interface.
3. One (or more) disk drives.

Order No. 5005RD (disk-based version) $29.95

TO ORDER:

SEE YOUR LOCAL
INSTANT SOFTWARE DEALER
OR CALL TOLL-FREE
1-800-258-5473

Instant Software™

PETERBOROUGH, N.H. 03458

TEXAS COMPUTER SYSTEMS

Radio Shack ®
Authorized Sales Center
We offer the lowest prices on

TRS-80 COMPUTERS

MODEL II 64K $3349 — An excellent computer for your business needs. Easy expandability & compatibility. No formal operator training needed. All accessories available—disk expansions, printers and software. Quick delivery-airfreight from Dallas

Model III 16K LIII $849
Model III 32K 2 Disks - $Call
Color Computer 4K LI $339
Pocket Computer & Acc. $Call

PRINTERS

DAISY WHEEL II $1695 — Carbon ribbon typewriter quality for word processing. 43 cps. Standard Centronics parallel interface. Full 16" width, individual sheets or friction feed. Tractor feed $250 extra. Fastest Daisy Wheel available at this price.

EPSON MX-80 $Call — Letter quality matrix similar to Line Printer IV and Centronics Printer but has full software control of 40, 80, 66 or 132 columns. 80cps bidirectional, tractor feed, disposable printhead. $300 less than nearest competitive printer. Best buy in multi-strike letter quality printer. Lists at $645. Our price includes cable & shipping.

ANADEX 9500/9501 $1385 — 200cps 7x9 or 9x9 matrix, dot addressable graphics, excellent quality & price. 10, 12, 13.3, 16.7 cpi; 6 or 8 lpi, standard parallel, serial, and current loop interfaces included

LINE PRINTER V $1616 — 160cps Designed for heavy business use. Bidirectional, 7x9 matrix

LINE PRINTER VI $1015 — Fast, versatile, takes single sheets & pin fed forms, 100 cps bidirectional, low profile List $1160

CUSTOM SOFTWARE FROM TCS

Model I & III Programs for the Epson MX-80 Printer

You get 6 programs for one low price. A library on disk or tape of programs and subroutines designed for gaining maximum use of the Epson MX-80 printer

BIGLETT/BAS—Produce large graphic letters and dump them to the printer/screen/disk Works from any Basic program
EPSON/SUB—Subroutine to easily use all of the EPSON printer commands Merge into any Basic program
DEFSTATE/SUB—Collection of pre-defined commands accessible from any Basic program Auto line centering, titles with borders etc., instantly with one word commands
DEMO/BAS—Tutorial program with 100's of REM remarks to guide you thru the use of your computer
JKL PATCH—A patch to NEWDOS 2 1/80's JKL screen printer function to print graphics from screen in the Regular Epson printer mode.
ARTICLE—Complete article on many printer possibilities plus new ideas, concepts, & trouble shooting tips

Model I or III, Tape or Disk
$39.95 for all 6 programs
$29.95 if purchased with Epson printer

The DOS Disk Library of Programs

A library of 5 programs for NEWDOS80, NEWDOS 2.1, TRSDOS (Model I & III)

DIRMAINT—A super directory-maintenance program Purge, copy, rename, re-format, etc., quickly with minimum typing Super graphic displays for easy use
FILEMAINT—Creates master directory of all files on your library of diskettes - Master file is handled on disk, not on memory for maximum speed Access any file in 2 seconds. 48K supports 200 diskettes 32K about 100 Takes less than hour to run 100 diskettes thru system
CHAINBLD/BAS—A super chaining file creation editor & assembler for NEWDOS80 Very fast, easy to use Reads and edits existing files
HOMEMAIL—Unique home mailing list that stores data by modifying program itself Fast & efficient for home user with less than 100 names
GASMILE—Maintains excellent records of auto gas use & complete reports on MPG, etc Also provides car maintenance alerts

Model I or III Disk **$39.95** for all 5 programs

Infinite Basic Diskette

A diskette of programs written in RACET's Infinite Basic User must have Infinite Basic (both modules) Extremely useful program with REM remarks to guide you in easy use of the 100 extra commands provided by this software system

Included on the disk is program to maintain a Video Cassette Library, an in-memory Mailing List, a demonstration of Matrix Techniques and equation solution, a screen input system routine Also a complete article on Infinite Basic with valuable information on commands that are glossed over in the original documentation

Model I or III Disk only **$39.95**

We can Air Freight from Dallas to major a/p near you for fast, efficient service. Call for information.

*Payment, Money Order, Cashier's Check, Certified Check, Personal Checks require 3 wks. to clear. VISA, MC, add 3%
*Prices subject to change any time
*No tax out-of-state TX add 5%
*All items new, guaranteed by manufacturer
*Delivery subject to availability
*Shipping extra, quoted by phone

TEXAS COMPUTER SYSTEMS ✓25
An Authorized **RADIO SHACK®** Sales Center F701
Box 1174, Brady, Texas 76825
TOLL FREE Number 800-351-1473
Texas Residents 915-597-0673

double entry of the letter Y or N when asked by Sargon I whether you care to play a game. (The Sargon I manual accuses Radio Shack of having an error in ROM, but it appears to be a Sargon error. Sargon I asks for two inputs. The first input is ignored.)

Sargon I does not store the move number in ASCII format as does Sargon II; appropriate conversions are made in the load and save routines for compatibility of data tapes.

In Sargon II the buffer area for the game position starts at 5115H (20757 decimal). The Sargon I buffer starts at 50C9H (20681 decimal). Each is formatted the same. Table 1 gives the code for each game piece and blank squares. The board position bytes are taken directly from the Sargon program. The first eight bytes on the data tape represent the first row of the chess board on the white side. Two more bytes, each containing FFH, follow and then the second row of the chess board follows that. Then come two more bytes of FFH, etc. Thus, it takes 78 consecutive bytes to represent the chess board.

The bytes containing FFH which occur between every row of eight board bytes mean nothing to us at this time but it is easier to record them on tape than to skip them. After these 78 bytes is the byte representing the move number in ASCII format.

One other useful location in Sargon II will be of interest to assembly language programmers. Sargon II has a keyboard scan routine starting at 767AH (30330 decimal). By patching into this location things can be added such as a tone (using the cassette output to an amplifier) to let you know when an input is expected from you; or such as adding a printing routine to print moves displayed on the left side of the screen or a graphic picture of the current board set-up. The print routines could use transparent codes, i.e., commands such as SHIFT P which will not show up on the screen but will signal the computer to branch to the correct routine. ■

Program Listing 1. MODIA patch for Sargon I.

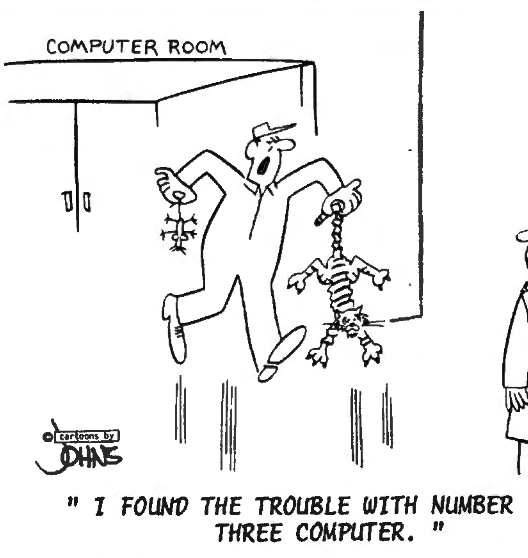
```
               00100 ;MODIA
               00110 ;SARGON I MODIFICATION
               00120 ;SAVE OR LOAD PIECE POSITION OF
               00130 ;PREVIOUS GAME TO CASSETTE TAPE
               00140 ;DATA TAPE GENERATED IS COMPATABLE
               00150 ;FOR USE WITH SARGON II WITH MODIIA
69F0           00160        ORG    69F0H           ;DOUBLE ENTR
Y CORRECTION
69F0 0000      00170        DEFW   00
69F2 0000      00180        DEFW   00
69F4 0000      00190        DEFW   00
69F6 00        00200        DEFB   00
6DDC           00210        ORG    6DDCH
6DDC 004E      00220        DEFW   LDDATA          ;PATCH MODIA
 ADDRESS IN SARGON I
4E00           00230        ORG    4E00H
4E00 21914E    00240 LDDATA LD     HL,MESS1        ;ADD "LOAD"
MESSAGE
4E03 062A      00250        LD     B,42            ;MESSAGE LEN
GTH
4E05 CD4B69    00260        CALL   694BH           ;SARGON PRIN
T AND KEYBOARD ENTRY ROUTINES
4E08 CD566D    00270        CALL   6D56H
4E0B CD4569    00280        CALL   6945H
4E0E FE59      00290        CP     89
4E10 2802      00300        JR     Z,LOAD          ;IF "L" LOAD
 GAME POSITION FROM CASSETTE
4E12 1830      00310        JR     SVDATA          ;JUMP TO SAV
E DATA? ROUTINE
4E14 CDF84E    00320 LOAD   CALL   READY           ;LOAD ROUTIN
E
4E17 CD1202    00330        CALL   0212H
4E1A CD9602    00340        CALL   0296H
4E1D CD3502    00350 LOOP1  CALL   0235H
4E20 77        00360        LD     (HL),A
4E21 23        00370        INC    HL
4E22 1D        00380        DEC    E
4E23 7B        00390        LD     A,E
4E24 20F7      00400        JR     NZ,LOOP1
4E26 CD3502    00410        CALL   0235H           ;GETS MOVE N
UMBER IN ASCII FORMAT
4E29 0630      00420        LD     B,48
4E2B FE0A      00430 ASC    CP     10              ;CHANGE MOVE
 NUMBER FROM ASCII
4E2D DA354E    00440        JP     C,ENT
4E30 04        00450        INC    B
4E31 D60A      00460        SUB    10
4E33 18F6      00470        JR     ASC
4E35 C630      00480 ENT    ADD    A,48
4E37 327D68    00490        LD     (687DH),A       ;BUFFER FOR
ONE'S DIGIT OF MOVE NUMBER
4E3A 78        00500        LD     A,B
4E3B 327C68    00510        LD     (687CH),A       ;BUFFER FOR
TEN'S DIGIT OF MOVE NUMBER
4E3E CDF801    00520        CALL   01F8H
4E41 C3656E    00530        JP     6E65H           ;JUMP BACK T
O SARGON AFTER LOAD
4E44 21BB4E    00540 SVDATA LD     HL,MESS2        ;ADD "SAVE"
MESSAGE
4E47 061F      00550        LD     B,31
4E49 CD4B69    00560        CALL   694BH
4E4C CD566D    00570        CALL   6D56H
4E4F CD4569    00580        CALL   6945H
4E52 FE59      00590        CP     89
4E54 2803      00600        JR     Z,SAVE          ;IF "S" SAVE
 GAME TO CASSETTE
4E56 C3CC06    00610        JP     06CCH           ;JUMP TO BAS
IC IF NO OPTION SELECTED
4E59 CDF84E    00620 SAVE   CALL   READY           ;SAVE ROUTIN
E
4E5C CD1202    00630        CALL   0212H
4E5F CD8702    00640        CALL   0287H
4E62 7E        00650 LOOP2  LD     A,(HL)
4E63 CD6402    00660        CALL   0264H
4E66 23        00670        INC    HL
4E67 1D        00680        DEC    E
4E68 7B        00690        LD     A,E
4E69 20F7      00700        JR     NZ,LOOP2
4E6B 3A7C68    00710        LD     A,(687CH)       ;GET TEN'S D
IGIT OF MOVE NUMBER
4E6E D630      00720        SUB    48              ;ASCII FORMA
T ROUTINE
4E70 0600      00730        LD     B,0
4E72 2809      00740 ZERO   JR     Z,NOINCR
4E74 F5        00750        PUSH   AF
4E75 78        00760        LD     A,B
4E76 C60A      00770        ADD    A,10
4E78 47        00780        LD     B,A
4E79 F1        00790        POP    AF
4E7A 3D        00800        DEC    A
4E7B 18F5      00810        JR     ZERO
4E7D 3A7D68    00820 NOINCR LD     A,(687DH)       ;GET ONE'S D
IGIT
4E80 D630      00830        SUB    48
4E82 80        00840        ADD    A,B             ;MOVE NUMBER
 NOW IN ASCII FORMAT
4E83 CD6402    00850        CALL   0264H           ;SAVE MOVE N
UMBER IN ASCII
4E86 3EFF      00854        LD     A,0FFH
4E88 CD6402    00858        CALL   0264H
4E8B CDF801    00860        CALL   01F8H
4E8E C3FA70    00870        JP     28922           ;JUMP TO SAR
GON START LOCATION
4E91 4C        00880 MESS1  DEFM   'LOAD PREVIOUS GAME POSITION
 FROM CASSETTE?'
4EBB 53        00890 MESS2  DEFM   'SAVE GAME POSITION TO CASSE
TTE?'
4EDA 52        00900 MESS3  DEFM   'READY CASSETTE AND PRESS EN
TER'
4EF8 21DA4E    00910 READY  LD     HL,MESS3        ;READY CASSE
TTE MESSAGE
4EFB 061F      00920        LD     B,31
4EFD CD4B69    00930        CALL   694BH
4F00 CD566D    00940        CALL   6D56H
4F03 CD4569    00950        CALL   6945H
4F06 CDC901    00960        CALL   01C9H           ;CLEAR SCREE
N
4F09 3E00      00970        LD     A,0
4F0B 21C950    00980        LD     HL,50C9H        ;SARGON STAR
T OF BOARD POSITION BUFFER
4F0E 1E4E      00990        LD     E,78            ;LENGTH OF S
ARGON BUFFER
4F10 C9        01000        RET
70FA           01010        END    28922           ;START ADDRE
SS OF SARGON
00000 TOTAL ERRORS
```

Program Listing 2. MODIIA patch for Sargon II.

```
               00100 ;MOD IIA
               00110 ;SARGON II MODIFICATION
               00120 ;SAVE OR LOAD BOARD POSITION OF
               00130 ;PREVIOUS GAME TO CASSETTE TAPE
               00140
44A0           00150        ORG    44A0H           ;PATCH INTO
SARGON LOAD TAPE
44A0 21AF45    00160        LD     HL,TITLE        ;BLOCK MOVE
TO CHANGE SARGON TITLE
44A3 11CF6E    00170        LD     DE,6ECFH
44A6 011200    00180        LD     BC,12H
44A9 EDB0      00190        LDIR                   ;BLOCK MOVE
44AB 21C145    00200        LD     HL,MESS         ;BLOCK MOVE
TO CHANGE ENDING OF SARGON "NEW GAME - -" MESSAGE
44AE 110D6F    00210        LD     DE,6F0DH
44B1 010600    00220        LD     BC,6
44B4 EDB0      00230        LDIR                   ;BLOCK MOVE
44B6 3E24      00240        LD     A,24H           ;CHANGE LENG
TH OF MESSAGE
44B8 325A70    00250        LD     (705AH),A
44BB 21CF44    00260        LD     HL,START        ;PATCH FOR M
ESS1 AND MESS2 ADDITION
44BE 225770    00270        LD     (7057H),HL
44C1 3EC3      00280        LD     A,0C3H          ;CHANGE TO J
P INSTRUCTION
44C3 325670    00290        LD     (7056H),A
44C6 21E544    00300        LD     HL,COND         ;PATCH TO CO
NDITIONAL STATEMENTS
```

Program continues

```
44C9 226870   00310        LD    (7068H),HL
44CC C30050   00320 SARGON JP    5000H            ;JUMP TO BEG
INNING OF SARGON PROGRAM
44CF 112D45   00330 START  LD    DE,MESS1         ;ADD OUR MES
SAGES
44D2 062A     00340        LD    B,2AH
44D4 CDD175   00350        CALL  75D1H            ;SARGON PRIN
T ROUTINE
44D7 115745   00360        LD    DE,MESS2
44DA 061F     00370        LD    B,1FH
44DC CDD175   00380        CALL  75D1H
44DF 11EF6E   00390        LD    DE,6EEFH         ;RESTORE SAR
GON INSTRUCTION LOST BY PATCH
44E2 C35970   00400        JP    7059H            ;JUMP BACK T
O SARGON AFTER MESSAGE INSERT
44E5 FE53     00410 COND   CP    83
44E7 2822     00420        JR    Z,SAVE           ;IF "S" SAVE
 GAME POSITION TO CASSETTE
44E9 FE4C     00430        CP    76
44EB 20DF     00440        JR    NZ,SARGON        ;IF "L" LOAD
 GAME POSITION FROM CASSETTE, IF NOT JUMP TO SARGON
44ED CD9645   00450 LOAD   CALL  READY            ;CASSETTE LO
AD ROUTINE
44F0 CD1202   00460        CALL  0212H
44F3 CD9602   00470        CALL  0296H
44F6 CD3502   00480 LOOP1  CALL  0235H
44F9 77       00490        LD    (HL),A
44FA 23       00500        INC   HL
44FB 1D       00510        DEC   E
44FC 7B       00520        LD    A,E
44FD 20F7     00530        JR    NZ,LOOP1
44FF CD3502   00540        CALL  0235H
4502 323550   00550        LD    (20533),A        ;GET NUMBER
OF MOVES PLAYED
4505 CDF801   00560        CALL  01F8H
4508 C37470   00570        JP    7074H            ;JUMP BACK T
O SARGON AFTER LOAD
450B CD9645   00580 SAVE   CALL  READY            ;CASSETTE SA
VE ROUTINE
450E CD1202   00590        CALL  0212H
4511 CD8702   00600        CALL  0287H
4514 7E       00610 LOOP2  LD    A,(HL)
4515 CD6402   00620        CALL  0264H
4518 23       00630        INC   HL
4519 1D       00640        DEC   E
451A 7B       00650        LD    A,E
451B 20F7     00660        JR    NZ,LOOP2
451D 3A3550   00670        LD    A,(20533)        ;SAVE NUMBER
 OF MOVES PLAYED
4520 CD6402   00680        CALL  0264H
4523 3EFF     00684        LD    A,0FFH
4525 CD6402   00688        CALL  0264H
4528 CDF801   00690        CALL  01F8H
452B 189F     00700        JR    SARGON
452D 4C       00710 MESS1  DEFM  'LOAD PREVIOUS GAME POSITION
 FROM CASSETTE,'
4557 53       00720 MESS2  DEFM  'SAVE GAME POSITION TO CASSE
TTE,'
4576 52       00730 MESS3  DEFM  'READY CASSETTE AND PRESS EN
TER:'
4595 88       00740        DEFB  136
4596 117645   00750 READY  LD    DE,MESS3         ;READY CASSE
TTE MESSAGE
4599 0620     00760        LD    B,20H
459B CDD175   00770        CALL  75D1H
459E CD2B00   00780 BACK   CALL  2BH
45A1 B7       00790        OR    A
45A2 28FA     00800        JR    Z,BACK
45A4 CDC901   00810        CALL  01C9H            ;CLEAR SCREE
N
45A7 3E00     00820        LD    A,0
45A9 211551   00830        LD    HL,5115H         ;SARGON STAR
T OF BOARD POSITION BUFFER
45AC 1E4E     00840        LD    E,78             ;LENGTH OF S
ARGON BUFFER
45AE C9       00850        RET
45AF 20       00860 TITLE  DEFM  ' SARGON-IIA ***** '
45C1 2C       00870 MESS   DEFM  ',L,S):'         ;ADD INSTRUC
TIONS FOR LOAD OR SAVE ROUTINES
4415          00880        END   4415H            ;START ADDRE
SS OF SARGON LOAD ROUTINE
00000 TOTAL ERRORS
```



" I FOUND THE TROUBLE WITH NUMBER THREE COMPUTER. "

✓See List of Advertisers on page 306

If thou doth protest too much, use your 80 to generate nasty letters of complaint.

Computerized Complaint

Dennis J. Gillig
708 Hingham Lane
Schaumburg, Illinois 60193

Evil and injustice are rampant in the land: nameless, faceless people harass us from credit card companies, politicians promise everything at election time and forget later, and computers send us form letters instead of service.

Good news! Here's a powerful new weapon. You can use your trusty computer to rattle their cages. This weapon works especially well against machines turned monsters by programmers, and operators and executives that have forgotten that flesh and blood exist out here in the real world.

When you read in the papers about your congressman voting against your desires you can use this program to fire a letter to him/her. When a department store sends you the ninth erroneous bill with that same overcharge (plus interest) on a computerized form saying they are about to repossess your microwave if you don't pay, you can use this program to send that blasted computer a piece of your mind.

The Basic of Writing a Nasty Letter

The basic element of a letter to get action are in the program. As you will see, you'll be able to add new forms to it for your specific needs. In fact, I'd appreciate printouts of any super letters you make up. If we get some

Continues to 279

Complaint Letter

From: Dennis J. Gillig
7822 Berkshire Drive
Hanover Park, ILL. 60103

To: Wonderful Drapery and Flange Com., Inc.
231456 East North West South Street
Sing Sing, NY 91788278

Attn: Carter MC Jimmy

This is contact number 1 with you about this complaint. I have a complaint about your low quality drapes, model 4345. I don't like the way they fade after 14 years of use.

The facts are:
1. Velvateen fiberglass rots with age.
2. The tie backs are only nine inches long and will not fit around the curtain.
3. I paid $2.00 for them at a garage sale and feel the re-sale is not too good.

This is what I want done:
Since you have a life time guarantee please replace them.

I do expect an immediate reply!

Figure 1.

Complaint Letter

From: Dennis J. Gillig
703 Hingham Lane
Schaumburg, ILL. 60103

To: Congressman Forest Trees
Capital Hill
Washington, DC 00000

Attn:

This is contact number 99 with you about this complaint.

Please sit back and RELAX.

I need your undivided attention. Prop your feet on your desk, tell your secretary to hold all calls, turn off the intercom, take a swig of coffee, a deep breath and concentrate on what I have to say.

I have a complaint about your vote against April Fool's Day as a national holiday.

I don't like your assumption that America is not ready for a day to commemorate congress.

The facts are:
America is ready for congress to take a holiday.

This is what I want done:
1. Declare an immediate vacation for congress of 6 months twice a year.
2. The above will do it all since no man's wealth and rights is in more jeapordy than when congress is in session.

A reply is not required.

Figure 2.

Complaint Letter

From the computer of:
Dennis J. Gillig
703 Hingham Lane
Schaumburg, ILL. 60103

To the computer of:
VRE Computerized Betting Service
2312 W. 33rd Street
Kildare, Ohio 00001

Dear fellow computer,

We have a problem. I figured (I do a lot of that) that I would write you directly since humans usually end up blaming us computers for their foul-ups anyway.

The problem concerns your daily betting bulletins.

I don't like the accuracy of the billing of the betting to me.

The facts are:
1. On September 2 your records show I bet on whether Teddy's mother would let him run—my bet was against not for.
2. On August 5th you recorded my bet as the cubs winning the world series—I dream but don't bet.
3. On October 1 you say I bet Carter would ask Rosalynn if he couldn't run she would run instead—no bet.

This is what I want done:
1. Get your act together.
2. Refund my losses of $1.49

I do expect an immediate reply

Figure 3.

"... operators and executives ... have forgotten that flesh and blood exist out here in the real world."

Program Listing.

```
30 CLEAR4000:DEFSTRA-F:DEFINTG-Z:DIMA(33):D0="COMPUTERI
   ZED CONSUMERISM PROGRAM":D1=STRING$(64,"-")
40 CLS:PRINTTAB(10)D0CHR$(13)D1:PRINT@192,"CHOOSE THIS:
   "TAB(25)"TO DO THIS:"CHR$(13):PP=0
50 PRINT"<1>"TAB(25)"CREATE NEW LETTER":PRINT"<2>"TAB(2
   5)"REVIEW <DISK> FILE":PRINT"<3>"TAB(25)"REVIEW <T
   APE> FILE":PRINT"<4>"TAB(25)"INSTRUCTIONS":INPUTG
60 ONG GOTO 70 , 15040 , 15010 , 15130
70 CLS:PRINTTAB(10)D0CHR$(13)D1:PRINT@192,"CHOOSE THIS:
   "TAB(25)"TO DO THIS:":PRINT:G=0
80 PRINT"<1>"TAB(25)"STANDARD LETTER"
90 PRINT"<2>"TAB(25)"COMPUTER TO COMPUTER"
100 PRINT"<3>"TAB(25)"SIT BACK AND RELAX"
110 PRINT"<4>"TAB(25)"(RESERVED)
120 PRINT"<5>"TAB(25)"(RESERVED)
130 PRINT"<6>"TAB(25)"(RESERVED)
140 PRINT"<7>"TAB(25)"(RESERVED)
150 PRINT"<8>"TAB(25)"(RESERVED)"
160 PRINT"<9>"TAB(25)"(RESERVED)":INPUTG:IFG>9GOTO 70
170 CLS:PRINTTAB(10)D0CHR$(13)D1:PRINT@192,"OFFENDING C
    OMPANY
STREET
CITY
STATE
ZIP
ATTN:
DATE"
180 FORI=1TO7:READP:READL:GOSUB10000:A(I)=B:NEXT
190 GOSUB20010:PRINT"MY COMPLAINT IS ABOUT ":P=281:GOSU
    B 15210 :PRINT@768,"INSERT SPECIFIC PRODUCT, SERVI
    CE, OR ISSUE":L=255:GOSUB10000:A(8)=B
200 GOSUB20010:PRINT"I DON'T LIKE ":P=281:L=255::GOSUB
    15210 :PRINT@768,"INSERT SHORT COMMENT ABOUT <WHAT
    > IS WRONG.  (MAX. 256 LETTERS AND SPACES.)":GOSUB
    10000:A(9)=B
210 I=10:S=1
220 GOSUB20010:PRINT"THE FACTS ARE:":PRINT
230 P=281:GOSUB 15210 :GOSUB 15220 :PRINT@P,S:P=P+3:GOS
    UB10000 :A(I)=B:I=I+1:S=S+1:IFI=21GOTO 270
240 A="":PRINT@768,CHR$(31):INPUT"CHOOSE:  <M>ORE FACTS
```

Program continues

good ones we'll share them with the editors and perhaps they will print some of the best.

Program Description

Please refer to Program Listing 1. It's long but it packs a punch. Normally, after writing a program I put it through another program that packs lines and removes all deadwood. However, you'll find this version easier to read.

The idea is to standardize protest or complaint letters with a minimum of fuss. If we don't want to take the time to go through the usual process of typing or writing a complaint letter, this program quickly gets the facts printed and on their way to the offender.

You are first asked if this is to be a new (create new file) letter or input from disk or tape. The data on tape or disk contains only the minimum facts you previously input. Sort of like having a carbon copy but no messy carbon. If you elect to start a new file you can select from three standard forms. The first is in bare-bones style and useful for just about any occasion. (See Fig. 1.) The second letter is from your computer to theirs. (See Fig. 3.) The third letter is to the big shot; which can be sent to the president of the offending company or your congressman. (See Fig. 2.)

Your first duty is to input the offender. Your name and address should be inserted where mine appears in the program and will automatically be inserted when printed. (Change line 400).

Note that program lines such as 170 are formatted with the down arrow. This saves memory since it is not necessary to type print to cause the display to drop a line. It does make the listing appear longer. Please note that this is formatted for my Integral Data IP-125 printer. Some of the control codes will not make sense to other printers.

Let It All Hang Out

You get to tell them what you don't like, what is wrong, the facts, what you want done, and whether or not you want to reply. You also tell them how many previous contacts you have made with the company via telephone, letter, etc.

When all information has been entered you'll be asked to ready the printer. Provisions are made to make more copies if needed. You can set margins by answering two questions. (Both margins are calculated from the left.) Then you can save on tape or disk if needed. I added a safeguard for disks: as the data is brought in from disk it is also displayed on the screen which ensures me I have the right file.

A few friends have some far out form letters they are adding to the area between lines 4000 and 9000. If you are severely paranoid this could be fantastic therapy for you. You could also convert this to love letters. It can also serve as a practical tool if you have a small business. In real estate sales we have a few standard letters we need to send where the basic information is about the same. This will help me send those just listed a house on your block or just sold a house on your block letters with ease.

If you intend to add other forms to this program you'll only need to add the name in the menu and the contents at the appropriate line from 4000 to 9000. If only a first paragraph is desired make the computer go to line 1010 to pick up the balance of the letter.

By changing the series of questions asked, you can use it to fill out a bill, invoice, or whatever you choose.

The first line in all my programs uses the same format. Put a REM after the CLS if you choose. (Ever save the program using another program's name and thereby wiping out some very valuable code? Put the name up front and don't save unless you list the beginning.) It also helps to know the version and date since my programs are after rewritten.

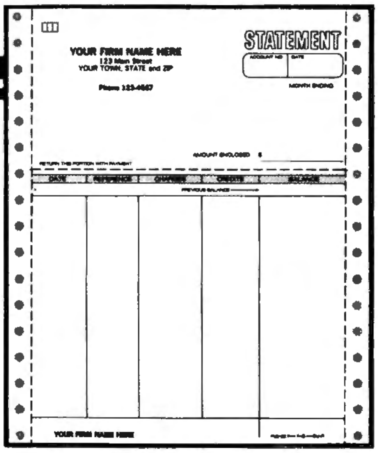
If you can tie a shoe and spell interface you should be able to master this program. If you don't like the program you may send me a complaint letter. ■

```
      ENTERED- <R>EVIEW- <C>ONTINUE";A:IFA="R"GOTO 1534
     Ø
250 IF A="M"GOTO 220
260 IFA="C"GOTO 270 ELSE 240
270 I=20:S=1
280 GOSUB20010:PRINT"THIS IS WHAT I WANT DONE:":PRINT:G
     OSUB 15210 :GOSUB 15220 :P=281:PRINT@P,S:P=P+3:GOS
     UB10000 :A(I)=B:S=S+1:I=I+1:IFI=31GOTO 320
290 A="":PRINT@768,CHR$(31):INPUT"CHOOSE:  <M>ORE DEMAN
     DS ENTERED- <R>EVIEW- <C>ONTINUE";A:IFA="M"GOTO 28
     Ø
300 IF A="R"GOTO 15380
310 IFA="C"GOTO 320 ELSE 290
320 GOSUB20010:INPUT"DO YOU WANT A REPLY (Y/N)";A(31):A
     (31)=LEFT$(A(31),1)
330 GOSUB20010:INPUT"HOW MANY TIMES HAVE YOU COMMUNICAT
     ED WITH THEM IN THE PAST";S2
340 GOSUB20010:INPUT"READY PRINTER AND PRESS <ENTER>";A
     :INPUT"HOW MANY COPIES";J:POKE16425,Ø
350 INPUT"WHAT IS LEFT MARGIN";ML
360 INPUT"WHAT IS RIGHT MARGIN";MR:PRINT@532,"** OUTPUT
      TO PRINTER **"
370 FORX1=1TOJ
380 LPRINTTAB((MR-ML)/2);"COMPLAINT LETTER":IFH=2THENLP
     RINTCHR$(13)CHR$(13)TAB(ML)FROM THE COMPUTER OF:":
     GOTO320
390 LPRINTSTRING$(5,13);TAB(ML)"FROM:  "
400 LPRINTTAB(ML)"DENNIS J. GILLIG";CHR$(13);TAB(ML);"7
     Ø8 HINGHAM DRIVE";CHR$(13);TAB(ML);"SCHAUMBURG, IL
     . 6Ø193";CHR$(13)TAB(ML)"PHONE: 312/98Ø-475Ø"STRIN
     G$(3,13)
410 IFH=2THENLPRINTTAB(ML)"TO THE COMPUTER OF:":GOTO 43
     Ø
420 LPRINTTAB(ML)"TO:"
430 FORX=1TO7:LPRINTTAB(ML)A(X):NEXT:LPRINTSTRING$(3,13
     )
440 ONGGOSUB 1ØØØ , 2ØØØ ,3ØØØ,4ØØØ,5ØØØ,6ØØØ,7ØØØ,8ØØØ
     ,9ØØØ
450 NEXT
460 GOSUB 1523Ø :RUN
1000 LPRINTTAB(ML)"THIS IS CONTACT NUMBER  ";S2+1; "  W
     ITH YOU ABOUT THIS COMPLAINT.";CHR$(13)
1010 IFA(8)<>""LPRINTTAB(ML)"I HAVE A COMPLAINT ABOUT:"
     :I=8:LPRINTTAB(ML);:GOSUB 18ØØØ :LPRINTCHR$(13)
1020 IFA(9)<>""LPRINTTAB(ML)"I DON'T LIKE:":I=9:LPRINTT
     AB(ML);:GOSUB 18ØØØ :LPRINTCHR$(13)
1030 LPRINTTAB(ML)"THE FACTS ARE:":FORS=1TO1Ø:I=S+9:IFA
     (I)=""GOTO 1Ø5Ø
1040 LPRINTTAB(ML);:GOSUB 18ØØØ :NEXT:LPRINTCHR$(13)
1050 LPRINTCHR$(13)TAB(ML)"THIS IS WHAT I WANT DONE:":F
     ORS=1TO1Ø:I=S+19:IFA(I)=""GOTO 1Ø7Ø
1060 LPRINTTAB(ML);:GOSUB 18ØØØ :NEXT:LPRINTCHR$(13)
1070 IFA(31)="Y"LPRINTCHR$(13)TAB(ML)"I DO EXPECT AN IM
     MEDIATE REPLY":GOTO 1Ø9Ø
1080 LPRINTCHR$(13)TAB(ML)"A REPLY IS NOT REQUIRED."
1090 LPRINTCHR$(11):RETURN
2000 REM COMPUTER TO COMPUTER
2010 LPRINTTAB(ML)"DEAR FELLOW COMPUTER,";CHR$(13)
2020 LPRINTTAB(ML)"WE HAVE A PROBLEM.  I FIGURED (I DO
     A LOT OF THAT)":LPRINTTAB(ML)"THAT I WOULD WRITE Y
     OU DIRECTLY SINCE HUMANS USUALLY END UP BLAMING US
     ":LPRINTTAB(ML)"COMPUTERS FOR THEIR FOUL-UPS ANYWA
     Y."
2030 LPRINTTAB(ML)"I HOPE THAT TWO COMPUTERS CAN GET TH
     IS FIGURED OUT.
2040 GOTO1Ø1Ø
3000 LPRINTTAB(ML)"PLEASE SIT BACK AND ";CHR$(1);"RELAX
     ";CHR$(2);"."
3010 LPRINTTAB(ML)"I NEED YOUR UNDIVDED ATTENTION.  PRO
     P YOUR FEET ON YOUR DESK,":LPRINTTAB(ML)"TELL YOUR
      SECRETARY TO HOLD ALL   CALLS, TURN OFF THE INTER
     COM, TAKE A SWIG OF COFFEE,"
3011 LPRINTTAB(ML)"A DEEP BREATH AND CONCENTRATE ON WHA
     T I HAVE TO SAY."CHR$(13)
3020 GOTO 1Ø1Ø
4000 GOTO7Ø:REM RESERVED FOR LETTER
5000 GOTO7Ø:REM RESERVED FOR LETTER
6000 GOTO7Ø:REM RESERVED FOR LETTER
7000 GOTO7Ø:REM RESERVED FOR LETTER
8000 GOTO7Ø:REM RESERVED FOR LETTER
9000 GOTO7Ø:REM RESERVED FOR LETTER
10000 B="":A=""
10010 PRINT@P,STRING$(L,136);
10020 PRINT@P,"*";:FORK=1TO2Ø:NEXT:PRINT@P," ";:FORK=1T
     O2Ø:NEXT:A=INKEY$:IFA=""THEN1ØØ2Ø
10030 IFA=CHR$(13)RETURN
10040 IFA=CHR$(8)THENIFA<>""THEN PRINTA;:IFLEN(B)=ØGOTO
     1ØØ2ØELSEB=LEFT$(B,LEN(B)-1):P=P-1:PRINT@P,STRING$
     (L-LEN(B),136);:GOTO1ØØ2Ø
10050 PRINT@P,A;:B=B+A:P=P+1:IFLEN(B)=LGOTO1ØØ7Ø
10060 GOTO1ØØ2Ø
10070 A=INKEY$:IFA=CHR$(13)RETURN
10080 IFA=CHR$(8)THENGOTO1ØØ4ØELSE1ØØ7Ø
15000 REM SERIES OF SUBROUTINES
15010 GOSUB2ØØ1Ø:PRINT@468,"**   READY TAPE   ***":GOSU
     B2ØØØØ
15020 GOSUB2ØØ1Ø:PRINT@468,"**** READING TAPE ****
15030 FORI=1TO31:INPUT#-1,A(I):PRINTA(I):NEXT:GOTO 15Ø7
     Ø :REM FOR PRINT-OUT
```

Program continues

```
15040 PP=2:GOTO 15300
15050 GOSUB20010:PRINT@468,"** READING DISK **":OPEN"I"
      ,1,A
15060 FORI=1TO32:INPUT#1,A(I):PRINTA(I):NEXT
15070 GOSUB20010:INPUT"DO YOU WANT A PRINT-OUT OF THIS
      DATA (Y/N)";A:IFA="N"GOTO40
15080 GOSUB20010:PRINT@468,"**  READY PRINTER  **":GOSU
      B20000
15090 FORI=1 TO 32:IFA(I)=""THEN 15110
15100 LPRINTA(I)
15110 NEXT:GOTO 40
15120 PRINT"NO SUCH FILE":GOTO 40
15130 REM INSTRUCTIONS
15140 CLS:PRINT"WHEN YOU NEED ACTION CALL ON YOUR COMPU
      TER TO ASSIST YOU IN THE BATTLE AGAINST EVIL AND I
      NJUSTICE.
15150 PRINTSTRING$(64,"*");"YOU GET TO SELECT FROM THRE
      E FORM LETTERS.

FIRST- PLAIN STATEMENT.  USEFUL TO A COMPANY OR POLITIC
      IAN.

SECOND- COMPUTER TO COMPUTER.  FOR THOSE CHARGE CARDS A
      ND OTHER IMPERSONAL ORGANIZATIONS

15160 PRINT"THIRD- TO THE BIG SHOTS.  HOPEFULLY YOU'LL
      GET THEIR ATTENTION.

OTHER STYLES CAN BE DEVELOPED AS YOU NEED THEM AND ADDE
      D.
";STRING$(64,"*")
15170 GOSUB20000
15180 CLS:PRINT"YOU CAN SAVE ON TAPE OR DISK BUT KEEP S
      EVERAL HARDCOPIES.  SUGGESTION: MAKE FOUR COPIES.
      1) TO OFFENDING PERSON OR COMPANY.  2)TO THIRD PAR
      TY IF NEEDED.  3) FOR YOUR FILES.  4)  TO OFFENDIN
      G PERSON OR COMPANY IF THEY FAIL TO COMPLY
15190 PRINT"
JUST ANSWER THE QUESTIONS AND ONLY ONE WARNING ----EACH
      ENTRY IS LIMITED TO 256 LETTERS AND SPACES.





HAVE FUN AND HERE'S HOPING YOU WIN THE DAY.

SAVING ON TAPE OR DISK SAVES THE PERTINENT DATA FOR FUT
      URE USE.
15200 GOSUB20000:GOTO40
15210 PRINT@704,D1:RETURN:REM SEPARATES INSTRUCTIONS FR
      OM DATA ON SCREEN
15220 PRINT@768,"ENTER UP TO 10 ITEMS  (EACH MAX. 256 L
      ETTERS AND SPACES).":RETURN
15230 REM SAVE ON TAPE OR DISK
15240 PRINT@192,CHR$(31):A="": INPUT"DO YOU WANT TO SAV
      E ON
<T>APE
<D>ISK
<Q>UIT";A:IFA="Q"GOTO40
15250 IFA="D"GOTO 15300
15260 IFA<>"T"GOTO 15240
15270 GOSUB20010:INPUT"PREPARE RECORDER - TYPE NAME OF
      FILE AND PRESS <ENTER> WHEN READY";A
15280 GOSUB20010:PRINT@468,"** RECORDING DATA **":PRINT
      #-1,A
15290 FORI=1TO32:PRINT#-1,A(I):PRINT"*";:NEXT:GOTO40
15300 GOSUB20010:INPUT"READY DISK AND INPUT

FILE NAME (MAX. 8 LETTERS)";A
15310 INPUT"DRIVE NUMBER";A1:A=A+":"+A1:REM FILE NAME
15320 IF PP=2 GOTO 15050
15330 PRINT@468,"** WRITING DISK **":OPEN "O",1,A:FORI=
      1TO32:PRINT#1,CHR$(34);A(I);CHR$(34);:NEXT:CLOSE:G
      OTO40
15340 FORI2=10TO20:GOSUB20010:PRINT"REVIEW OF FACTS

15350 PRINTA(I2):IFA(I2+1)=""GOTO 15370
15360 GOSUB20000:NEXT
15370 GOSUB20000:GOTO 240
15380 FORI2=20TO30:GOSUB20010:PRINT"REVIEW OF DEMANDS

15390 PRINTA(I2):IFA(I2+1)=""GOTO 15410
15400 GOSUB20000:NEXT
15410 GOTO290
18000 A="":A=A(I):L=LEN(A):K=1
18010 M1=MR-ML
18020 IFM1<=LGOTO 18050
18030 IFM1+K>LTHENM1=L-K
18040 IFM1>LTHENIFMID$(A,M1,1)<>" "THENM1=M1-1:GOTO 180
      40
18050 FORX=1TOM1
18060 LPRINTTAB(ML)MID$(A,K,1);:IFK=LTHEN 18090
18070 K=K+1:NEXT:LPRINTCHR$(13)
18080 IFK<LGOTO 18010
18090 RETURN
20000 INPUT"PRESS <ENTER> TO CONTINUE";C:RETURN
20010 PRINT@192,CHR$(31):RETURN
32000 DATA 217,25,281,25,345,25,409,2,473,5,537,20,601,
      8
```

TRS-80™ Compatible "carbonless" Continuous Statements



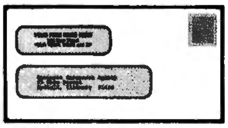
small quantities, low prices, fast delivery

Order as few as 500 statements imprinted with your firm name and address.

Only 27^{95}

NEBS 9062 Statements are software compatible with the TRS-80, Model 1, Level II, Accounts Receivable package #26-1555.

TRS-80 is a Trade Mark of the Radio Shack Co., Subsidiary of the Tandy Corp.



SPEED COLLECTIONS
Product 772 DU-O-VUE® Envelope (3⅝"x 6⅞") eliminates envelope addressing.

Product 9062 — Size 6"x 8½" detached. Prices include your firm name, address and phone in top section, plus your name only in lower section. Printed in black ink. Available in single (white) or duplicate (white, canary) continuous sets.

QUANTITY	SINGLE Product 9062-1	DUPLICATE Product 9062-2	Product 772 DU-O-VUE® Envelopes
10,000	$192.00	$355.00	$138.00
6,000	128.00	228.00	92.00
4,000	99.00	169.00	64.50
2,000	59.00	99.00	36.25
1,000	38.75	61.00	20.75
500	27.95	39.95	12.25

ORDER TODAY! MONEY-BACK GUARANTEE.
FAST SERVICE BY MAIL or PHONE TOLL FREE 1+800-225-9550 (Mass. residents 1+800-922-8560). It is our policy to ship within 6 working days following our receipt of your order.

Please ship: Date____________19____ CODE 460

________9062-1 STATEMENTS (Single)

________9062-2 STATEMENTS (Duplicate)

________772 DU-O-VUE® Envelopes

________Information on continuous checks and other computer forms.

HEADING TO BE PRINTED ON FORMS: (Please type or print)

STREET

CITY and STATE ZIP

PHONE

AUTHORIZED SIGNATURE

If you wish us to BILL and SHIP differently from above please indicate.

Nebs Computer Forms

194

78 Hollis Street, Groton, Mass. 01450
A division of New England Business Service, Inc.

COMPUTER HOBBYIST SPECIALS!

Daisy Wheel Terminals

RS-232, used, off-lease. Features Diablo Hytype 1 printer, up to 30 CPS, plot & text modes, dual pitch and much, much more. Includes data, operation manual, schematics. "As-is" machines may need some service.

Add $40 for Hdlg. and Pkg. Pay Shipping When Delivered.

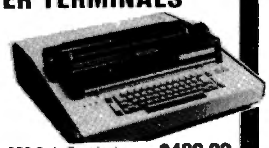
TESTED & FUNCTIONAL NOW ONLY $1299.00 ea

"AS-IS"... ONLY $899.00 ea

SURPLUS

* "SELECTRIC" TYPEWRITER TERMINALS

15" "Selectric" typewriter terminals are complete with solenoid drivers and RS-232 Serial I/O circuitry. BCD or correspondence code. Each unit includes operational printer, power supply, and untested RS-232 I/O board. Features:



- Full KSR Capability
- 88 Character Set (Upper and Lower Case)
- 134.5 Baud I/O
- 132 Col.
- 14 CPS Rate
- Standard Typewriter KBD
- 117 VAC. 60 Hz Operation

BDC Code Terminal ..... $489.00 EACH
Correspondence Code Terminal ..... $569.00 EACH
Printer Maintenance Manual ..... $20.00 EACH

No type element included. Line cord and some data included.

* Reg. Trademark of IBM Corp.

Add $20 For Hdlg. & Pkg. - Pay Shipping When Del

Write or Call for our Bargain-Packed PERIPHERAL FLYER!!
Prices are Subject to Change on All Items

316

CFR Associates, Inc.

MAIL ORDERS: Box 144, Newton, N.H. 03858

PHONE ORDERS: 617/372-8536 Sorry, No Collect Calls. MasterCharge & VISA Welcome

WAREHOUSE: 18 Granite Street Haverhill, Mass. 01830

PACIFIC OFFICE SYSTEMS

COMPUTER INTERFACES & PERIPHERALS

- **POS-100 NRZ1 TAPE DRIVE CONTROLLER/FORMATTER** — Now your micro can read and write IBM/ANSI compatible NRZ1 format 9-track magnetic tapes. The POS-100 consists of S-100 bus card, 6' ribbon cable, tape drive controller card, cable to Pertec-Standard NRZ1 Tape drive, plus documentation and Z-80 or 8080 software (specifiy). Power is derived from tape drive and S-100 bus. Ship Wt.. 10 lbs. Suggested Retail Price $995.00
- **POS 103/202 "MIX or MATCH" MODEM** — Unique POS control design permits use in one housing of both Bell-compatible 103 (0-300 baud) and 202 (0-1200 baud) modem modules originally made by VADIC Corp. for a telephone company subsidiary. FEATURES: RS-232 serial Interface, auto-answer, auto-dial, LED display, telephone line interface via acoustic coupler, manual DAA, or auto-answer DAA (sold separately). FULLY ADJUSTED; no special tools required. 3,000 mile range over standard dial-up telephone lines. Ship wt.: 15 lbs.
  PRICES ..... POS 103 Modem — $199.95; POS 202 Modem — $299.95; POS 202 Modem w/Auto-Answer — $349.95; POS 103/202 Modem — $499.95; FCC-Approved Auto-Answer DAA — $125.00; Acoustic Coupler — $29.95.
- **POS DAISY-WHEEL PRINTER INTERFACE for TRS-80** — Will drive Diablo HyType I, HyType II, and Qume Q and Sprint 3 printers. Includes 1K user-available memory for custom print routines (such as graphics, bidirectional printing, etc.). Programmed to respond to print commands from BASIC ELECTRIC PENCIL™ and SCRIPSIT™ software. Draws its power from printer. Ship wt.: 5 lbs. Price $250.00
  Cables, each (Specify HyType I, HyType II, or Qume) ..... $ 25.00
- **POS ASCII INTERFACE for IBM I/O SELECTRIC** — This Centronics-style parallel printer interface will drive an IBM Model 731 or 735 I/O typewriter (EBCD and Correspondence codes). No software needed. Features on-board EPROM which holds up to 8 ASCII-to-IBM code tables for different type spheres. Closed-loop operation runs at maximum printer speed; stops and starts on a single character without loss of data. Requires +12VDC and ±5VDC power source. Ship wt.: 5 lbs. Price $249.95
  Power Supply (±5VDC, +12VDC, +24VDC for Solenoids on Printer) ..... $ 49.95
- **CONVERT OFFICE SELECTRIC TO I/O TYPEWRITER** — Kit includes assembled solenoids, switches, wire harness, magnet driver PCB plus instructions for installation and mCPU interface. Price $150.00
- **"FORMALINER" Variable Width Forms Tractor for 15" Selectrics** $95.00
- **GTE Model 560 ASCII SELECTRIC I/O Terminal** — With RS-232 Serial Interface and digital cassette deck for use as memory typewriter. Ship wt.: 100 lbs. Price, tested and adjusted $1,195.00
- **POS ASCII IBM SELECTRIC PRINTER** — 15" Selectric from GTE terminal cleaned and adjusted with POS Centronics-style ASCII printer interface. UC/LC, carbon and fabric ribbons. Compatible with TRS-80, Apple, SOL and other CPU parallel printer ports. Ship wt.: 75 lbs. Price $895.00

PACIFIC OFFICE SYSTEMS 153
2265 Old Middlefield Way Mountain View, Calif. 94043 (415) 493-7455

What's 64 characters long and disappears?

Line Loss

Hubert C. Borrmann
2840 S. Circle Drive
Colorado Springs, CO 80906

Occasionally in our daily lives we notice that something has suddenly disappeared. However much we look for it, it's gone.

Well, this is a fact of life, and we have to reconcile ourselves to the fact that this is the way it is, the way it was and the way it may always be.

I want to talk about something which is 64 characters long and suddenly disappears: A line from our TRS-80 screen.

This happens when we have to respond to a PROMPT to enter something. Whenever we type in the desired response and press ENTER, the line following the one where the cursor was, is suddenly gone.

This is a fact and we are probably willing to live with it. There is, however, something we can do about it. If we work with games or graphics, for instance, we can save the line in a string-variable. We can replace it on the screen when the program regains control after ENTER. Type in the seven-line program shown in the Program Listing. Run it and you will see the second line disappear. Now go into edit-mode and remove the quote (') from lines 40 and 60. Run it again.

Now we've solved the case of the wiped-out line.■

```
10  CLS:CLEAR 200
20  FOR N = 0 TO 255:PRINT@N, "#";:NEXT N
30  PRINT@5,"ENTER ANY CHARACTER";
40  'B$ = " ":FOR N = 64 TO 127:B$ = B$ + CHR$(PEEK(15360 + N)):NEXT N
50  INPUT A$
60  'PRINT@64,B$;
70  IF A$ = "END" THEN END ELSE 30
```

Program Listing

✓See List of Advertisers on page 306

A few enhancements to your favorite Shack utility.

Take T-Bug Higher

W. R. Stanley N4TF
City Route 13, 204 Avery Lane
La Grange, GA 30240

This article is written for those who have moved T-BUG to the high RAM range, possibly with the help of the article written by Irwin Rappaport (Get T-BUG High, p. 118) in the first issue of this magazine.

Rappaport indicated that a wholesale move of T-BUG from the vicinity of 4380h to locations starting at 7380h greatly simplifies the logistics involved.

The smallest number of instructional address bytes are to be changed only when the high-order half of the address is to be dealt with.

My TRS-80 was originally a 4K Level I machine. After getting remotely acquainted with BASIC usage, I purchased T-BUG and entered the exciting, fascinating world of machine language programming.

After my excitement died down, I realized that T-BUG lacked some operating features. Thus, this monitor program was cumbersome to use.

The two shortcomings that seemed to detract most from overall satisfaction with T-BUG were its video display approach and inability to move by choice through consecutive memory locations.

Rappaport voiced my thoughts about the address block location in the original T-BUG. I didn't devise any programmed means to move T-BUG to the high end of my then adequate memory, however. I did it the hard way, instead, by keying every byte into high memory using the T-BUG itself. Level I doesn't PEEK and POKE.

A few months afterwards the situation changed. Then came Level II and 16K! Now I can get something done!

Wait a minute. T-BUG is still in the way. So, I laboriously moved it again, this time to the memory block starting at 7380h.

Changes Needed

If you have used T-BUG much, you've probably wished it could do more, or at least do some things differently. One thing that bothered me was the video display format – not the memory address and byte display. The garbage on the rest of the screen bothered me.

Another thing I wanted T-BUG to do was complete erasure of the screen, on command. Another capability I wished the original monitor had was that of displaying bytes at the same time another machine language program was being loaded under command.

One more change I sorely needed was to backstep through memory one byte at a time. Those readers who have typing skill approaching mine can appreciate the value of that last change to T-BUG.

It is aggravating to make a mistake when entering data in the M mode. It is frustrating to have to exit the memory exam mode and re-enter it, at or preceding the address where the wrong data byte was entered – just to correct that one, last entry.

Wouldn't it be nice just to hit the space bar and back up to that last address, correct the data, and then proceed? A simple program change can allow just that.

Original Program

The original T-BUG program provided for erasure and scrolling of the 16 columns on the left side of the video display screen. Anything to the right displayed when T-BUG was brought up, remained on screen, unless a dedidicated effort was made to load 20h (ASCII space) into the appropriate video RAM locations. This erased the extraneous material.

I solved the video display garbage problem in two ways. My modified T-BUG High not only scrolls an entire horizontal line at a time, but, on command, it erases the entire screen and displays the prompt symbol at the upper left corner. Using the scrolling technique, then, the garbage was quickly shoved off the screen.

The capability to display bytes on the screen, as a system tape is being loaded by T-BUG, is not of prime importance; however, it is entertaining and useful to know. It allows you to read the file name of the tape, and with a little practice, you can identify the memory address where the first program byte is loaded. You can also see on display the execution address of that program.

Caution

A word of caution is in order. The program changes to be undertaken will, of necessity, modify the flow of events through the original T-BUG program which has been moved to the high memory area. Modifications to T-BUG High, using itself to modify its own program locations, could also result in an immediate crash.

I urge you to follow this sequence: (1) Load T-BUG High in the normal manner. Do not key /ENTER when it is loaded; instead, push RESET and return to BASIC. (2) Enter SYSTEM again and load T-BUG Low. Bring up this program in the normal manner. (3) Enter all pertinent changes to T-BUG High, including program segments above and below the original program locations. (4) Use T-BUG Low to jump to 73A0h, the starting address of T-BUG High. (5) After a thorough check of the modified T-BUG High, use it to punch a tape of itself.

Now a word to those who have relocated T-BUG to some range of addresses other than those starting at 7380h. The modifications detailed here will still produce the desired result, but only if you make the necessary address changes to suit your case.

It will be up to you to locate the bytes in your version of

✓See List of Advertisers on page 306

	Change From	To
73E5	FE	C3
73E6	46	80
73E7	CA	78
73E8	0D	00
73E9	78	00
742C	FE	C3
742D	0D	AA
742E	28	78
742F	0E	00
7554	11	00
7555	30	00
755B	10	40
7566	DD	00
7567	19	00
7573	10	40

Table 1. Starting with 73E5; changes to be made to T-BUG High.

T-BUG High to be changed, and decide the proper address locations to be used.

Modifications

Table 1 shows the location at which changes to T-BUG High are to be made. Table 2 includes the bytes to be added at the addresses at the end of the main T-BUG High program. Shown in Table 3 are the bytes to be entered in the block of addresses preceding the start of T-BUG High. The following is a description of the changes, and the logic behind them.

Since commands are being added to T-BUG, it is necessary to make room in the command table for additional function calls. Instead of inserting the additions to the table and shifting the entire remainder of the program upward in memory to make room for changes, I inserted a JUMP instruction at the head of the table.

I had to eliminate a comparison and a conditional JUMP instruction in the original table in order to make room for the inserted instruction. Dislocated table segments appear then, in the command table extension starting at 7880h.

I ran into the relocation problem again when I inserted bytes into other locations in T-BUG High to modify its performance. I thus took a logical (easy) way out. You will note that other changes detailed in Table 1 result from the need for additional program space: jumps are made to new subroutines outside the original program, and those subroutines normally return to the next logical location in the unmodified program.

The program change starting at 742Ch provides a means of including the subroutine starting at 78AAh in the routine that normally increments memory addresses when in the M mode. If the space bar is pressed, the memory address is decremented to the previous location. If data at the indicated memory location is changed, or the enter key is pressed, the memory address is incremented as usual.

The change at 755Bh provides for 64 column scrolling, instead of the 16 column scrolling in the original program.

The changes at 7566h and 7567h are necessary to permit proper operation of the memory backstep subroutine.

Additions

Following are comments on Tables 2 and 3. Note that data bytes are added in two blocks, one on either end of T-BUG High. One block of data is added following the last instruction byte at 7824h. The other data block to be added starts at 7300h and stops a few bytes short of 7380h, the first byte in the unmodified relocated T-BUG.

Readers who are curious about the discrepancy between the normal ending address of T-BUG High (7824h) and the last address reserved for T-BUG use (7980h) need not wonder any more. Portions of this memory area are used for a stack. Other segments and specific locations are used by T-BUG for temporary storage of data in housekeeping operations.

Do not relocate the data in the block starting at 7300h to any area nearer the end of T-BUG unless you have disassembled T-BUG and located all pertinent reserved memory areas.

A command table extension starting at 7880h is called indirectly whenever the prompt (#) symbol appears on the screen. (T-BUG is awaiting a command.)

Note the F command (46h = ASCII F) in this table. Remember that it was dislocated to provide room in the main program for a jump instruction. Also in this extension are hex representations for space, D and E. More on D and E later.

If the space bar is hit when in the command mode, the subroutine at 7897h–78A6h is invoked, completely erasing the screen. A jump is then made to the main program where the prompt is again displayed. T-BUG is once again in the command mode.

The subroutine at 78AAh–78C1h is called during the M mode, and allows the user to step forward or backward in memory.

If the D key is pressed in the command mode, the main program is modified. Incoming bytes from a machine language tape can then be displayed and loaded into memory. The E key restores this aspect of T-BUG High to its original configura-

7880	FE	78A2	01
	46		FF
7882	CA		03
	0D	78A5	ED
	78		B0
7885	FE	78A7	C3
	20		80
7887	CA		73
	97	78AA	FE
	78		0D
788A	FE	78AC	CA
	44		3E
788C	CC		74
	47	78AF	FE
	73		20
788F	FE	78B1	28
	45		03
7891	CC	78B3	C3
	5D		30
	73		74
7894	C3	78B6	DD
	EA		2A
	73		3F
7897	3E		78
	20	78BA	DD
7899	32		2B
	00	78BC	DD
	3C		2B
789C	21	78BE	DD
	00		22
	3C		3F
789F	11		78
	01	78C2	C3
	3C		3E
			74

Table 2. Starting with 7880; bytes added at end of T-BUG High.

7300	C5	7327	23	734D	01
7301	06	7328	23		0A
	08	7329	7C		00
7303	CD	732A	FE	7350	ED
	62		40		B0
	77	732C	20	7352	C9
7306	10		03	7353	CD
	FB	732E	21		00
7308	CD		C0		73
	10		3F	7356	C9
	73	7331	22	7357	00
730B	C1		3D	7358	00
730C	C9		78	7359	00
730D	00	7334	D1	735A	00
730E	00	7335	C1	735B	00
730F	00	7336	F1	735C	00
7310	E5	7337	E1	735D	11
7311	F5	7338	C9		82
7312	C5	7339	E6		77
7313	D5		0F	7360	21
7314	2A	733B	C6		69
	3D		30		73
	78	733D	FE	7363	01
7317	4F		3A		0A
7318	CB	733F	FA		00
	0F		44	7366	ED
731A	CB		73		B0
	0F	7342	C6	7368	C9
731C	CB		07	7369	C5
	0F	7344	77	736A	06
731E	CB	7345	23		08
	0F	7346	C9	736C	CD
7320	CD	7347	11		62
	39		82		77
	73		77	736F	10
7323	79	734A	21		FB
7324	CD		53	7371	C1
	39		73	7372	C9
	73				

Table 3. Starting with 7300; bytes added in front of T-BUG High.

tion—then there is no byte display while loading. Naturally, the monitor must be in command mode for the E key to function.

As shown by inspecting the extension, depression of the D key causes a jump to a subroutine starting at 7347h. This subroutine, from 7347h to 7352h, loads the data bytes from 7353h to 735Ch into the main program starting at 7782h. The result is the rewriting of a segment of the main program.

Now when a system tape is loaded by T-BUG High, the subroutine from 7300h to 730Ch is called periodically. This in turn invokes another subroutine, starting at 7310h, that displays the tape data bytes being loaded.

Depression of the E key calls the subroutine from 735Dh to 7368h. This rewrites into the main program the data bytes residing between 7369h and 7372h. Transfer of this block of bytes back to the main program will restore that segment of T-BUG High to its original configuration.

Operation

Little needs to be said about using the added functions and capabilities, since I'm sure that most of you will explore them as soon as the program has been modified. The following is therefore for the more patient users.

In the command mode, the space bar clears the entire screen. In the M mode, the bar backspaces the current memory address. It has no function in any other mode of operation.

In the command mode, the D key enables the tape data byte display during load operations. Incidentally, depression of the D (for DISPLAY) key results in no visible indication of a program change. It becomes visible during L, of course.

Again, in the command mode, depression of the E(ERASE) key results in no overt indication until the L mode is entered. Of course, no tape byte display will be generated.

You'll note that during a load operation where tape bytes are displayed, the display is not scrolled when the screen is filled. I couldn't get satisfactory scrolling and correct tape loading using my crude programming techniques. I simply used too much time for scrolling and, as a result, the tape loading suffered. So, the scrolling had to go. All is not lost, though.

Data bytes displayed on the bottom line are still useful, even though they appear to be all overwritten. The last bytes loaded from the tape are still on that last line. Look closely for group 78. This is a special termination code that tells the LOAD subroutine that the end of the program material is at hand. Now look at the two bytes to the right of the 78 code. These two denote the execution address (in hex) of the program just loaded.

Now look at the first line on the screen. You will see 55 as the left byte. This is a code group designating a system tape. The next six bytes are the file name in ASCII. If the name is less than six characters, the remaining six will be spaces (20h). The next byte will be 3C, indicating to the loading program that the file is to start. The second and third bytes to the right of the 3c byte will denote the starting hex memory location where the file loading begins.

Use modified T-BUG High to load T-BUG Low and decipher some of the mysterious tape format information you've heard about. Note the execution address of T-BUG Low—43A0h. The execution address of T-BUG High should therefore be 73A0h.

This article was written to share my thoughts with you, in hopes that you'll be inspired to develop and refine program techniques and share them, in turn, with us.

These modifications make T-BUG easier to work with, and help remove the last obstacle preventing some prolific minds from developing new software. We could all benefit. ■

Run this utility and it will tell you more about your programs than even you know.

Know-It-All

Alan Sehmer
150 A Lorretta Drive NW
Corrales, NM 87048

While going through my September issue of *80 Microcomputing* I came upon William Noel's VARDOC program ("Document those Variables," page 88).

I entered the program, but after a few runs I became uphappy, knowing the first occurrence, last occurrence and number of times a variable is used wasn't enough.

I started out by modifying his program, but by the time I finished the only part of the program that resembled the original was the name.

List Variables and Lines

This program will list all variables, and all lines on which a variable is used. It will also indicate if a variable is an array or not. The program will go on to list all lines pointed to by IF THENs, GOTOs and GOSUBs. It will also list the lines that contain them.

If the program bombs, it will give a rough indication why. And finally, while the program is running, it will show the BASIC line number worked on, the memory address accessed and the number of variables found thus far. The display is mainly for us nail biters who dislike to press enter and watch nothing happen but the cursor disappear.

Now for the other side of the coin, the program is slow. To analyze itself the program took 12 minutes. (The program is 2633 bytes long.)

Worst of all, you must have a 32K machine, and this program loves memory—13072 bytes of it. There are things that can be done, however. The program as listed contains about 540 imbedded blanks that can be deleted. Because the arrays are integer, the line numbers in the program analyzed must not exceed 32766. However, the arrays are probably larger than they need to be. As listed, the program can handle 75 variables, each used in 60 lines, and 100 IF THENs, GOTOs or GOSUBs.

There are four arrays used in pairs of two: A$,A and L,M. The A$,A pair is used to hold the variables and where they are used; the L,M arrays hold the line numbers of GOTOs, etc. and where they point. Since these array dimensions, from time to time, will need to be changed, I have made that easy. At the tail end of line 65020, there are three variables: A2, A3 and A4 followed by a list of which variable affects which array. If during a run of the program, one of the arrays is filled, a SORRY ARRAYS ARE FULL message will be given along with the name of the array filled. Line 65020 can then be edited to change the size of the arrays and, of course, the arrays can all be made smaller to get more free memory.

The program looks into memory to find the start of user memory so that it may be used with BASIC, BASICR and BASIC2. It also adjusts the PEEKs if it must peek into memory over 32767.

To use this program on a disk system, enter and save it with the A option (SAVE "VARDOC2",A). Load the target program, merge VARDOC2 and run 65000.

On a disk system there is a semi-painless way to pick up

```
>PRINT PEEK(16548);PEEK(16549)          <ENTER>
 233  66                                ;Save these numbers
READY
>CLOAD                                  ;Load target program
READY
>PRINT PEEK(16633);PEEK(16634)          <ENTER>
 166  69                                ;End pointer plus 2
READY
>POKE 16548,164 : POKE 16549,69         <ENTER>    (166 - 2 = 164)
READY
>CLOAD                                  ;Load VARDOC
READY
>POKE 16548,233 : POKE 16549,66         ;Numbers from first PEEK
READY
>
```

In this example 166 – 2 = 164 is easy enough, but if that 166 had been a 1, then what? Then I must borrow from the 69, this would give me a 257 68. Remember, borrowing 1 from the right number adds 256 to the left. And to clean things up 257 – 2 = 255, the first poke would be POKE 16548,255:POKE 16549,68. Nothing to it.

Example 1.

RUN BIG CPM* PROGRAMS ON YOUR MODEL 1 TRS-80*

WITH 39 or 54K of use space. Can use Mixed Drivers 5¼ or 8" (Variable Tracks) Compatable with all TRS*80 Operating Systems and all TRS*80 Programs.(usable on 48K disk systems only) Boards fit into keyboard Inclosure.

FREEDOM OPTION..............$245
MEMORY EXPANSION OPTION....$295

Send Check or Money Order to:
(MASS. RESIDENTS, PLEASE INCLUDE 5% TAX)

F.E.C. ltd. ✓141
P.O.Box 2368 • Woburn, MA. 01888
(617) 944-5329

BLACKJACK PLAYERS

A YEARS INCOME IN 4 MONTHS.

Basic Strategy Tutors, for the novice & expert alike. Use our Tutors & quickly learn how to become A WINNER. Earn $40 hour. Learn the secrets of our success. Our Tutors will teach you how to:

** BET YOUR MONEY **
** PLAY EACH HAND **
** NOT GET BARRED **
** BE A BUSINESS **

**** BEWARE ****

Our Tutors are not games!!! They are serious tools to make winners out of you. The authors have spent 7 years playing & being winners. Now so can you, with the aid of our Tutors. Our Tutors are based on the strategies developed by Lawrence Revere. More advanced strategies are available call for info. Consultation is available for you by mail or phone. From 12pm - 12am M.S.T.

PKG. 1 contains 10 Basic Tutors for Atlantic City, or Nevada Rush $25 for 16K Mod I or Mod III, cass. $30 for 32K disk Please add $2.00 for shipping.

✓205

MICRO BLAJAK SYSTEMS INC.
2800 NORTH ELLEN STREET
FLAGSTAFF, ARIZONA 86001
(602) 774-5723 or (602) 774-7561

— *Professional* —
Real Estate Software
For Apple or TRS-80

Property Management System
(32K, 1 Disk Systems)

Features:

- Tenant Information
- Late Rent Reports
- YTD & Monthly Income
- Handles —
  Partial Payments
  Returned Checks
  Advance Payments
- Prints Receipts
- 5 Digit Expense Accounts
- Building Expense Report
- Vendor Expense Report
- Income Tax Report
- All Reports Can Be Printed
- Complete Documentation
- Easy Data Entry & Edit
- 200 Units per File

Price $225.00

Real Estate Analysis Modules:
(Cassette or Disk)

1) Home Purchase Analysis
2) Tax Deferred Exchange
3) Construction Cost/Profit
4) Income Property Cashflow
5) APR Loan Analysis
6) Property Sales Analysis
7) Loan Amortization

$35 Per Module

At Computer Stores Everywhere
Or Order COD Direct
(Cal Residents Add 6% Sales Tax)
(213) 372-9419

Realty Software Company™

Dept. M 1116 "F" 8th St., Manhattan Beach CA 90266

DISASSEMBLED HANDBOOK FOR TRS-80

VOLUME 3—$18. POSTPAID

Chapter 1: Writing Disassembler Programs
Chapter 2: High Speed Disassemblers
Chapter 3: Spooling Theory & Practice
Chapter 4: Port Encoders & Decoders
Chapter 5: Writing Interrupt Programs
Chapter 6: D/A Converters & Construction
Chapter 7: A/D Converters & Construction
Chapter 8: High Speed Morse Code Program
Chapter 9: Comm. Bulletin Board Systems
Chapter 10: Radio Teletype From A to Z
Chapter 11: Self-Programmed Learning Q/A
Appendix A: Volumes 1-2-3 combined index
Appendix B: Vols. 1-2-3 Pgms. on Disk $30

VOLUME 1—$10. POSTPAID
6th printing

VOLUME 2—$15. POSTPAID
4th printing

—GERMAN & FRENCH LANGUAGE EDITIONS—

RICHCRAFT ENGINEERING LTD.
Drawer 1065, Wahmeda Industrial Park
Chautauqua, New York 14722
phone (716) 753-2654 for COD orders
(US funds: add $4.50 overseas airmail) ✓276

LEARNING TOOLS

- Spelling Primer
- Vocabulary Builder
- Crossword Generator

EACH PROGRAM FEATURES:

- HUNDREDS OF WORDS
- GRAPHIC SCORING MONITOR
- PROGRAMMABLE SPEED CONTROL

Each Program Available In 4 Levels:
Grades 1-2; 3-4; 5-6; 7-8.
(Specify Grade)

16K - LEVEL II
$ 6.95 each — any two for $ 9.95

DYNATEK INFO. SYSTEMS
586 CONCORD AVE.
WILLISTON PARK, N.Y. 11596

✓253 *"Quality Software at Affordable Prices"*

STEPWISE

MULTIPLE LINEAR REGRESSION

Fast Compiled Machine Language
Comprehensive Data Base Manager
 * Transformations
 * Lags
Designate Any Variable As Dependent At Run Time
Descriptive Statistics
Correlation Matrix
ANOVA Table
Partial Correlation Coefficients Each Step
Future Releases Are Upward Compatible And Include: Factor Analysis, Time Series, Linear Programming
For 2 Disk, 48K TRS-80®, With Line Printer
Write For Information On Other System Configurations
Price $89.00 — Documentation Only $10.00
Visa And Master Charge Accepted

Barstrann Corporation — Dept. A ✓291
P.O. Box 265 Mid City Station
Dayton, Ohio 45402

STOCKMARKET INVESTMENTS MADE EASIER

Fulfill your dream to invest in the stockmarket.
Invest as little as $1000, or as little as $100 per month.
The potential is there to double that investment within 12, 24, or 36 months. The stock selection and the buy/sell decisions are the key.
We have the financial management computer programs for you.
Through the use of your TRS-80* Level II home computer and the **COMPUVEST** programs, you can determine the stocks in which to invest and when to buy and sell.
The Graham/Rea stock selection criteria program-10 criteria for a stock buy decision........................**$10.00**
COMPUVEST Lump Sum Investment Program—Lump sum investment approach for the small stock investor **$12.50**
COMPUVEST Monthly Investment Program—Monthly investment approach for the small stock investor...**$12.50**
All Three Programs..............................**$30.00**
For information or to order cassette tape programs write:

COMPUVEST, INC.
6725 Fishburg Road
Dayton, Ohio 45424 ✓498

*A Registered Trademark of Radio Shack.

COMPUTER FORMS™
DISTRIBUTORS OF QUALITY COMPUTER SUPPLIES

S U P P L I E S

Distributor of Computer Paper Products

VISA · master charge · Check or Money Order

Box 1000 sheets 9½"x 11" (sheet size 8½"x 11") tractor feed #15 lb. economy computer paper..$11.95 + $2 shipping

Box 750 sheets 9½"x 11" (sheet size 8½"x 11") tractor feed #20 lb. economy #2 Bond computer paper . . . .$11.95 + $2 shipping

Box 750 sheets 9½"x 11" (sheet size 8½"x 11") tractor feed #20 lb. High Quality #1 Bond with Watermark computer paper. . .$14.95 + $2 shipping

Box 1000 15/16"x 3½" 2 across tractor feed 8" carrier address labels. . .$8.95 + $1.50 shipping (One across available, specify which)

Box 700 6"x 4" continuous postcard stock tractor feed Blank white, yellow, green (specify)...$13.95 + $2 shipping

Box 300 sets (900 sheets) 9½"x 11" (sheet size 8½"x 11") 3 part carbon copy style tractor feed paper . . . .$13.95 + $2 shipping (2 part also available . . .400 sets . . .$13.95 + $2 shipping)

Catalog available check(✓) reader service number

✓240 **Computer Forms • (616) 429-7922**
5588 Caribou, Stevensville, MI 49127

LISP

INTERPRETER FOR THE TRS-80*

SEE THE AUGUST, '79 ISSUE OF BYTE TO FIND HOW EASY IT IS TO USE LISP.

- INCLUDES MORE THAN 30 FUNCTIONS & PREDICATES.
- CODED FOR HIGH-SPEED OPERATION.
- FULL ARITHMETIC CAPABILITY.
- USER'S MANUAL.
- SUPPLIED ON CASSETTE FOR 16K-64K LEVEL II.

SEND $50 IN CHECK OR MONEY ORDER TO: ✓259

CYBER INNOVATIONS
WORLD TRADE CENTER
P. O. BOX 58657
DALLAS, TEXAS 75258

*TRADEMARK OF TANDY CORPORATION

10K bytes more memory, run under BASIC2 instead of Disk BASIC. Load and merge the programs as shown, then give a CMD"T and CSAVE to tape. Go back to DOS and boot up under BASIC2, CLOAD, then run 65000.

But what if I don't have a disk? Follow these instructions:

Type in VARDOC2 and CSAVE it to tape.

CLOAD the target program.

PEEK locations 16548, 16549 and write them down.

PEEK locations 16633 and 16634. This is the end of program pointer, plus two. Two must be subtracted from the left hand number, if the number is one or zero. 256 can be borrowed from the right hand number. Example 1 should make all this clear.

POKE the calculated number into locations 16548 and 16549. BASIC will now load the next program behind the first program.

CLOAD VARDOC2. If a listing is made now, the target program will seem to be lost. This is a good time to make sure the VARDOC2 program got loaded correctly. If not, CLOAD it again.

POKE locations 16548 and 16549 with the numbers from the first PEEK. Now list the program to make sure both programs made it through all this.

If all goes well, a listing should show the target program followed by the VARDOC2 program. ■

Program Listing 1. Variable Documentation Program

```
65000 '
65005 '---------- VARIABLE DOCUMENTATION PROGRAM ------
      -------
65010 ' LOAD TARGET PROGRAM, MERGE THIS PROGRAM THEN RU
      N 65000
65015 CLEAR 200 : DEFINT A,L,M : CLS : INPUT"DO YOU WIS
      H HARD COPY (Y/N)";A$ : IF A$="Y" THEN P=1 ELSE P=
      0
65020 INPUT"NAME OF TARGET PROGRAM";A$ : CLS : PRINT "V
      ARIABLES AND REFERENCES FOR '";A$;"'" : A2=75 : A3
      =60 : A4=100 'A$   -A2 A-A3 M&L-A4
65025 IF P=1 LPRINT"VARIABLES AND REFERENCES FOR '";A$;
      "'" : LPRINT CHR$(138)
65030 DIM A(A2+2,A3),A$(A2+2),L(A4+2),M(A4+2) : A1=0 :
      A$(1)="[" : M(1)=32767 : M1=0 : SW=1 : U=PEEK(1654
      8)+PEEK(16549)*256 : GOTO 65045
65035 IF U>32767 THEN F=-1*(U-65535) ELSE F=U
65040 T=PEEK(F) : U=U+1 : PRINT@ 128,"MEMORY ADDRESS ="
      ;U;"      "; : RETURN
65045 GOSUB 65035 : D=T : GOSUB 65035 : D=T*256+D : GOS
      UB 65035 : E=T : GOSUB 65035 : E=T*256+E : IF E>32
      766 THEN 65155 ELSE PRINT@ 64,"WORKING ON LINE #";
      E;
65050 GOSUB 65035 : IF T=32 THEN 65050
65055 IF T=0 OR T=147 OR T=251 THEN U=D : GOTO 65045
65060 IF T=34 THEN SW=-SW
65065 IF SW<0 GOSUB 65035 : GOTO 65060
65070 IF T=141 OR T=145 OR T=202 THEN C$="" : GOTO 6513
      0
65075 IF T<65 OR T>90 THEN 65050 ELSE C$=""
65080 IF (T>47 AND T<58) OR (T>64 AND T<91) OR T=33 OR
      T=35 OR T=36 OR T=37 THEN C$=C$+CHR$(T) : GOSUB 65
      035 : GOTO 65080
65085 IF T=40 THEN C$=C$+"(*)"
65090 U=U-1 : A1=A1+1 : IF A1>A2 THEN A$="A$" : GOTO 65
      210
65095 FOR J=1 TO A1 : IF C$=A$(J) THEN 65115 ELSE IF C$
      >A$(J) THEN 65125
65100 PRINT@ 256,"SORTING ARRAYS"; : FOR R=A1+1 TO J ST
      EP-1 : A$(R+1)=A$(R) : NEXT : A$(J)=C$
65105 FOR R=A1+1 TO J STEP-1 : FOR S=0 TO A3 : A(R+1,S)
      =A(R,S) : NEXT
65110 NEXT : A(J,0)=E : FOR R=1 TO A3 : A(J,R)=0 : NEXT
       : PRINT@ 256,"                  "; : J=100 : GOTO 651
      25
65115 FOR R=0 TO A3 : IF A(J,R)=0 OR A(J,R)=E THEN A(J,
      R)=E : R=A3+10 : J=A2+2
65120 NEXT : A1=A1-1 : IF R<>A3+11 THEN A$="A" : GOTO 6
      5210
65125 NEXT : PRINT@ 192,"NUMBER OF VARIABLES SO FAR =";
      A1; : GOTO 65050
65130 GOSUB 65035 : IF T=32 THEN 65130 ELSE IF T<48 OR
      T>57 THEN RE=VAL(C$) : U=U-1 : GOTO 65140
65135 C$=C$+CHR$(T) : GOTO 65130
65140 M1=M1+1 : IF M1>A4 THEN A$="L&M" : GOTO 65210 ELS
      E IF C$="" THEN M1=M1-1 : GOTO 65050
65145 FOR J=1 TO M1 : IF RE<=M(J) AND RE>M(J-1) THEN FO
      R Q=M1+1 TO J STEP-1 : M(Q+1)=M(Q) : L(Q+1)=L(Q) :
       NEXT : M(J)=RE : L(J)=E : J=A4
65150 NEXT : GOTO 65050
65155 CLS : PRINT"VAR","USED IN LINES" : IF P=1 LPRINT"
      VAR","USED IN LINES"
65160 FOR R=1 TO A1 : LE=0 : PRINT" ";A$(R), : IF P=1 L
      PRINT " ";A$(R),
65165 FOR S=0 TO A3 : IF A(R,S)=0 THEN S=A3 : PRINT : I
      F P<>1 THEN 65175 ELSE LPRINT CHR$(13) : GOTO 6517
      5
65170 GOSUB 65215 : PRINT A(R,S); : IF P=1 GOSUB 65220
      : LPRINT A(R,S);
65175 NEXT : NEXT : PRINT : IF P=1 LPRINT CHR$(138)
65180 PRINT"LINE #","REFERENCED BY"; : IF P=1 LPRINT"LI
      NE #","REFERENCED BY"
65185 FOR R=1 TO M1 : LE=0 : IF M(R)<>M(R-1) PRINT : IF
       P=1 LPRINT CHR$(13)
65190 IF M(R)=32767 PRINT "NONE" : END
65195 IF M(R)<>M(R-1) PRINT M(R), : IF P=1 LPRINT M(R),

65200 IF L(R)<>L(R-1) OR M(R)<>M(R-1) GOSUB 65215 : PRI
      NT L(R); : IF P=1 THEN LE=LE+LEN(STR$(L(R))) : IF
      LE>45 THEN LE=0 : LPRINT CHR$(13);"", : LPRINT L(R
      ); ELSE LPRINT L(R);
65205 NEXT : END
65210 PRINT@ 384,"SORRY ARRAY '";A$;"' IS FULL" : END
65215 PP=PEEK(16416)+PEEK(16417)*256 : IF PP-15359-INT(
      (PP-15359)/64)*64>57 THEN PRINT : PRINT, : RETURN
      ELSE RETURN
65220 LE=LE+LEN(STR$(A(R,S))) : IF LE>45 THEN LE=0 : LP
      RINT CHR$(13);"", : RETURN ELSE RETURN
```

65000–65030	House keeping
65035–65040	Peek subroutine
65045–65050	Get BASIC line number and memory address of start of next BASIC line.
65055–65065	Do I have an end of line, remark or quotes?
65070	Do I have a IF THEN, GOTO or GOSUB?
65075	Do I have a letter?
65080–65090	If letter build variable string A$.
65095–65125	Insert string A$ and line number into array sort if necessary.
65130–65135	Assemble line number after GOTOs etc.
65140–65150	Insert line numbers into arrays L&M.
65155–65175	Print variables and line numbers.
65180–65205	Print references for GOTOs etc.
65210–65220	Subroutine to get neat display.

Program Breakdown

✓*See List of Advertisers on page 306*

A mean program.

On the Average

Len Gorney
Box 91 RD 5
Clarks Summit, PA 18411

What does the word *average* mean? The arithmetic mean is not the only way by which average values can be expressed. The following tutorial concerns different ways an average value may be expressed,and includes a Program Listing, which will calculate these values with a TRS-80 Level II. Throw away your calculator, load the program, and enter the average world.

The Mean

The arithmetic mean is the method most often used to calculate the average value of a list of numbers. The following numbers represent temperature readings taken over one week: 54.0, 38.0, 36.5, 39.5, 31.5, 31.5, 10.5. Add these values to obtain the sum 241.5 and divide by 7, the number of readings. The quotient, 34.5, becomes the arithmetic mean. This is the method of determining the average most often meant when we see the word average in print. What it tells us is that, given our list of numbers, the value 34.5 would act as the balance point of the list.

The Median

Another value which can describe the average of our list of numbers is called the median, which gives us the middle value. If we sort the list into ascending order, we can easily pick out the number in the center of the list. In our example, 36.5 is the value at the center of sorted numbers! 10.5, 31.5, 31.5, 36.5, 38.0, 39.5, 54.0. The median is the number which has 50 percent of the numbers above it and 50 percent below it.

Since our list contained an odd amount of numbers, it was easy to pick out the middle value. However, when an even amount of numbers are in the list, there is no middle value to choose by counting: there are two middle values. It is necessary to take these middle values and calculate their arithmetic mean. Let us take the following list: 10.5, 31.5, 31.5, 36.5, 38.0, 39.5, 54.0, and 55.0. The middle values, 36.5 and 38.0, are added and the sum, 74.5, is divided by two, in order to arrive at the arithmetic mean of these two values: 37.25.

The Midrange

The midrange value is another measure of the average value of a number list. Take the arithmetic mean of the largest and the smallest values in the list. In our example, the largest number is 54.0 and the smallest is 10.5, which, when added, gives us the sum of 64.5. Dividing by two gives us a quotient of 32.25, the midrange of our list. Similar averages can be calculated for the midrange by disregarding the largest and the smallest values in the original list, and taking the arithmetic mean of the next largest and the next smallest. This method is used to score athletic events to smooth out biases introduced by the judges. If the amount of

```
1000 REM  AVERAGES
        LEN GORNEY, BOX 91 R.D. 5, CLARKS SSUMMIT PA
1010 DEFINT B, J, L, N, T:
     DEFSNG H, V:
     DIM V(1000), VM(1000)
1020 CLS:
     INPUT"ENTER NUMBER OF ITEMS IN SAMPLE ";N
1030 CLS:
     PRINT,"ENTER ITEMS"
1040 FOR J = 1 TO N:
       PRINT"ITEM ";J;:
       INPUT V(J):
     NEXT J
1050 CLS:
     PRINT"SORTING ITEMS NOW . . ."
1060 REM  BUBBLE SORT
1070 FOR J = 1 TO N-1
1080   B = N
1090   IF V(B) < V(B-1) THEN H = V(B):
                             V(B) = V(B-1):
                             V(B-1) = H
1100   B = B - 1:
       IF B > J THEN 1090
1110 NEXT J
1120 CLS:
     PRINT"    MEAN = ";:
     M = 0
1130 FOR J = 1 TO N:
       M = M + V(J):
     NEXT J:
     M = M / N
1140 PRINT,M
1150 PRINT"  MEDIAN = ";:
     IF (INT(N/2)*2) = N THEN M = (V(N/2)+V((N/2)+1))/2
                        ELSE M = V((N/2)+1)
1160 PRINT,M
1170 PRINT"MIDRANGE = ";:
     M = (V(N)+V(1))/2
1180 PRINT,M
1190 PRINT"    MODE = ";:
     FOR J = 1 TO N:
       VM(J) = 1:
     NEXT J
1200 JJ = 1
1210 FOR J = 1 TO N-1:
       IF V(J) = V(J+1) THEN VM(JJ) = VM(JJ) + 1
                        ELSE JJ = J + 1
1220 NEXT J
1230 M = 1
1240 FOR J = 1 TO N
1250   IF VM(J) = 1 THEN 1260
                  ELSE IF M > VM(J) THEN 1260
                                    ELSE M = VM(J)
1260 NEXT J
1270 IF M = 1 THEN PRINT"VALUE DOES NOT OCCUR": GOTO 1310
1280 FOR J = 1 TO N
1290  IF VM(J) = M THEN PRINT,V(J)
1300 NEXT J
1310 END
```

Program Listing 1

items in a list is large enough, the values should not fluctuate as widely as in a list which has few values.

The Mode

Our last average value is called the mode. This value tells us what value occurs most often in the list.

Although this average value is not as useful as the mean and median values, the mode does illustrate any bunching about an individual value in a list of numbers. To discover the mode of a list of numbers, find the value which appears most often. For example, the modal value of our list of numbers is 31.5, which appears twice in our list. When more than one modal value appears in a list, it is said to be bimodal or trimodal or n-modal, depending on how many modal values appear.

In addition to the usual description of an average, we have the median, midrange, and mode. Each value represents a different method of calculating a value. The value you choose depends on what message you want to convey about the list of numbers.

The Average Program

The program listed in this article can be used by a TRS-80 Level II system user who wants to find values for the different types of averages. In addition to values for mean, median, midrange and mode, the program will sort the input list of numbers into ascending order. This is necessary to save programming steps when the midrange and median values are calculated.

Input is simply the number of items in your list and the values in it. The program quickly calculates the mean value, then sorts the list into ascending order. Depending on how many items you input, the program may take awhile to output the median value. Midrange and mode will be output to the screen. To input another list, simply type RUN after the READY is displayed.■

✓See List of Advertisers on page 306

A game of serpentine strategy in offense and defense.

BOA

Jeff Myers
210 Park Ave.
Hillsboro, OH 45133

Boa, a graphics game, evolved from my early experiments with simulated motion by sequentially setting and resetting pixels (the smallest graphics character).

I combined this with what I'd learned about the keyboard's memory to check inputs.

Boa is a game for two and its goal is simple: to survive longer than your opponent. The two of you manipulate sets of four keys each (2, Q, W, A or I, J, K, M) in order to control your Boa "snakes" which grow and grow.

The first player to run into a wall or either snake, loses. The winner is indicated by an asterisk on the scoreboard.

Although it's simple, Boa offers the constant interplay of offense and defense that is characteristic of many good games, along with a chance to use a variety of game strategies.

You can also play Boa alone. For this solitaire version I added an indicator to show the percent fullness of the screen at the game's end. The challenge is to operate both snakes at once, striving for the highest percentage score.

In earlier versions I used single pixels rather than two pixel squares as fundamental units. In those, the horizontal movement is slower than the vertical movement, which adds an interesting facet to the game. Boa can easily be converted by altering the lines containing SET and RESET pairs. ■

```
0 '   *******  BOA  *******
1 '   BY JEFF MYERS     210 PARK AVE     HILLSBORO OH 451
    33
5 CLEAR1000:DEFINTA-Z:PO=26:CZ=0:CP=1:CN=-1:KT=15010:KN
    =15009:KF=15004:K0=15360:GOSUB9000
10 GOSUB900:GOSUB8000:GOSUB950:P=2*P0+6:Q=3*Q0+4:SET(P,
    Q):SET(P+CP,Q):R=2*R0-CP:S=3*S0+4:SET(R,S):SET(R-C
    P,S):C=2
12 P1=CP:Q1=CZ:R1=CN:S1=CZ
100 'A INPUT
120 IFPEEK(KF)AND4THENQ1=CN:P1=CZ:GOTO200
130 IFPEEK(KF)AND2THENP1=CN:Q1=CZ:GOTO200
140 IFPEEK(KF)AND128THENP1=CP:Q1=CZ:GOTO200
150 IFPEEK(KN)AND2THENQ1=CP:P1=CZ
200 P=P+P1+P1:Q=Q+Q1:IFPOINT(P,Q)ORPOINT(P+CP,Q)THEN300
210 SET(P,Q):SET(P+CP,Q):C=C+CP:GOTO500
300 'B WINS
310 BS=BS+CP:W$="B":GOTO 730
500 'B INPUT
520 IFPEEK(KT)AND2THEN R1=CZ:S1=CN:GOTO600
530 IFPEEK(KT)AND4THENR1=CN:S1=CZ:GOTO600
540 IFPEEK(KT)AND8THENR1=CP:S1=CZ:GOTO600
550 IFPEEK(KT)AND32THENS1=CP:R1=CZ
600 R=R+R1+R1:S=S+S1:IFPOINT(R,S)OR POINT(R-CP,S)THEN 7
    00
610 SET(R,S):SET(R-CP,S):C=C+CP:GOTO100
700 'A WINS
710 AS=AS+CP:W$="A"
720 'END ROUTINE
730 IFW$="A"THENP5=577 ELSE P5=641
732 PRINT@P5,"*";
740 GOSUB950
745 PRINT@769,"SCREEN";
750 PRINT@833,"";:PRINTUSING"###";INT(100*C/2530+.5);:P
    RINT"%";
752 PRINT@898,"FULL";
760 H$=INKEY$:IFH$="" THEN 760 ELSE IF ASC(H$)<>13 THEN
    760
770 RESTORE:GOTO10
900 'BOXES
902 Z$=CHR$(179)
905 A$(1)=CHR$(159)+Z$+CHR$(175):A$(2)=CHR$(149)+Z$+CHR
    $(170)
910 A$(3)=CHR$(181)+CHR$(191)+CHR$(186):B$(1)=CHR$(151)
    +Z$+CHR$(175)
915 B$(2)=CHR$(149)+Z$+CHR$(174):B$(3)=CHR$(181)+Z$+CHR
    $(190)
920 P0=11+RND(21):Q0=1+RND(3):R0=38+RND(21):S0=8+RND(3)
925 U0=P0+64*Q0:L0=R0+64*S0
930 CLS:FOR I= 0 TO 2:PRINT@U0+64*I,A$(I+1);:PRINT@L0+6
    4*I,B$(I+1);:NEXT:RETURN
950 'SCORE
952 PRINT@513,"SCORE:";
954 PRINT@578,"A";AS;:PRINT@642,"B";BS;
956 RETURN
8000 'EDGE
8010 FOR X0=K0 TO 15423:POKEX0,131:NEXT
```

Program continues

```
8012 FOR X0=16320TO16383:POKEX0,176:NEXT
8020 FORR=0TO15:POKEK0+64*R,191:POKEK0+64*R+7,191:POKEK
     0+64*R+63,191:NEXT
8030 FOR C=1TO6:POKE K0+C,191:POKEK0+960+C,191:NEXT
8040 Z=K0+64:C9=0
8042 FOR N=1TO6:Z=Z+1:READ Y:POKEZ,Y:NEXT
8044 C9=C9+1:IFC9<7 THEN Z=Z+58:GOTO8042
8050 FOR Z=705 TO 710:PRINT@Z,CHR$(143);:NEXT
8888 RETURN
8900 DATA 149,136,140,140,140,162,181,178,179,179,179,1
     84
8910 DATA 151,168,188,188,188,130,189,178,179,179,179,1
     84
8920 DATA 159,161,188,188,180,139,149,168,188,188,188,1
     28
8930 DATA 143,143,143,143,143,143
9000  'INSTR
9003 CLS:GOSUB9400:PRINT
9010 PRINT" TWO PLAYERS USE THE KEYS BELOW TO MANEUVER
     THEIR SNAKES AROUND   THE SCREEN. THE FIRST PLAYER
      WHO RUNS INTO SOMETHING LOSES."
9100 PRINT:PRINTTAB(8)"PLAYER  A";TAB(46)"PLAYER  b"
9130 U$=CHR$(91):D$=CHR$(92):L$=CHR$(93):R$=CHR$(94)
9140 F$=L$+" Q  KEY  W "+R$:S$=L$+" J  KEY  K "+R$
9160 PRINTTAB(12)U$;TAB(50)U$
9170 PRINTTAB(12)"2";TAB(50)"I"
9200 PRINTTAB(6)F$;TAB(44)S$
9210 PRINTTAB(12)"A";TAB(50)"M"
9220 PRINTTAB(12)D$;TAB(50)D$
9230 PRINT" A SOLITAIRE VERSION CAN BE PLAYED BY OPERAT
     ING BOTH SNAKES (!)  AND TRYING TO MAXIMIZE THE %A
     GE SCORE SHOWN AT THE LOWER LEFT."
9250 PRINTTAB(19)"* HIT ANY KEY TO BEGIN *";
9260 H$=INKEY$:IFH$=""THEN9260 ELSE RETURN
9400 W$=CHR$(191):V$=CHR$(189)
9500 X$=W$+CHR$(143)+V$+" "+CHR$(190)+CHR$(143)+V$+" "+
     CHR$(190)+CHR$(143)+V$
9510 Y$=W$+CHR$(142)+V$+" "+W$+" "+W$+" "+W$+CHR$(176)+
     W$
9520 Z$=W$+CHR$(188)+CHR$(159)+" "+CHR$(175)+CHR$(188)+
     CHR$(159)+" "+W$+" "+W$
9532 PRINT@PO,X$;:PRINT@PO+64,Y$;:PRINT@PO+128,Z$
9540 RETURN
```

Program Listing 1. Boa

A look at the Shack's new peripheral.

Talk to Your TRS-80

Art Welcher
360 S. Wetherly Drive
Beverly Hills, CA 90211

The Radio Shack Speech Recognition unit is now available and, judging from the demand, will prove to be very popular. It is called an Isolated Word-Speaker Trained system.

You read into the microphone, which is provided with the Voxbox, the words you want to install into memory. The box has a limited vocabulary of 32 words at one time. After reading these into memory, go into a "listen" mode via software programming. The computer then returns an index number for each word in its memory whenever it hears it. It also returns an error code when it hears a word it doesn't have in its vocabulary or when it doesn't understand the word.

From the return code index you may program the TRS-80 to do remarkable things!

These numbers can control software just as if you entered data via keyboard. And almost any program that asks for keyboard input may be changed to accept a vocal input through the microphone.

Additions

With the unit is a machine language driver program that must be read into high memory before the hardware functions. The SYSTEM command does this. Radio Shack has included three cassette tapes depending on whether you have 16K, 32K or 48K (Voxbox needs a minimum of 16K).

Also included are three demonstration programs. One is called VOXPLOT, and herein lies the key to the Speech Recognition unit. By running VOXPLOT we see a graph of the four parameters making up the identification pattern of each vocabulary word or phrase.

The first graph is the energy (dBs) of the 900 to 2200 hertz range. The second is the energy of the 100 to 900 hertz range. The third graph is the dominant frequency in the 900 to 5000 hertz band, and the fourth curve is the dominant frequency in the 200 to 900 hertz band.

The graphs are generated by reading these four parameters at 10 millisecond intervals for the duration of the spoken word or phrase up to a length of 1.5 seconds. If there is more than .1 second of silence, the computer assumes that is the end of that word/phrase.

These data are stored in a reserved position of the driver program which we played in with SYSTEM when we started. It now resides in high memory as part of the driver program. This operation is called the "training" of the computer.

When you enter a user program or one of the demo programs, you first train the computer by storing in the driver program the vocabulary that must accompany the user program.

It is not clear in the instructions that this must be done each time after powerup. Also, if you run two different programs requiring different vocabulary, you must "re-train" the computer each time. This obviously just won't do!

While the Driver Program is written for BASIC II, the instruction book furnishes information for converting programs to Disk BASIC, but there are errors. Remember, in Disk BASIC all POKE addresses higher than 32767 become negative numbers. It becomes much more convenient to use HEX numbers in the POKE,PEEK and DEFUSR(n) statements. Don't forget to use &H followed by the HEX number if you have more than a 16K machine.

The manual also includes a paragraph explaining in detail how to save the Driver Program on DISK using the TAPEDISK utility program. This works well. If you have NEWDOS plus, you may also save it on disk using LMOFFSET/CMD program and entering the correct addresses for your machine as shown under the TAPEDISK instruction in the manual.

The program included will create and save on disk a driver program complete with a set vocabulary file. Each user program may now have its own Vocab file/driver program, which will automatically load with the user program if you have NEWDOS. This means that you may enter a pre-trained vocabulary and not be required to train each time you use a program.

If you are using TRSDOS, you must make several changes in the routine for making and loading pre-trained vocabulary files.

```
line 70 delete "3. ENTER VOCAB 4. SAVE
   VOCAB"
line 80 delete 19000,19100
line 19010 add a REM to the first of the line
line 19110 add a REM to the first of the line
line 20005 delete "CMD"LOAD VDR"" and
   add
     PRINT " VOXBOX SOFTWARE NOT
             LOADED.
             RETURN TO DOS BY CMD'S'
             LOAD VDR
             BASIC* "
```

This is exactly what you must do each time. You must load the proper Vocab file/driver program before you load the Basic User program. If you have NEWDOS plus, the program will execute as shown, with nothing additional.

Vocab File

To make and SAVE a Vocab

file, you must first load the Driver Program in DOS. Next, load and RUN VOCTRAIN after you are in Basic; then return to DOS by using CMD"S" and execute a DUMP as outlined in the TRSDOS manual where the format is identical to the instuction of line 19110 following CMD". That's why we REMed it instead of deleting. It is always there to see when you go to DOS.

By placing your own vocabulary in the DATA lines 20100 to 20140 you are ready to make and save a Vocab file. The line numbers in VOCTRAIN are purposely spaced out into complete sub sections so that it may be MERGEd more easily with your existing programs without extensive line renumbering.

The manual includes a small discussion on the driver program parameters, the most important being the rejection parameter. By POKEing a different number into the proper location you may loosen or tighten the fit of a spoken word to your Vocab file. By loosening the fit, you may have more mistakes of confusing two words, but less "WORD UNKNOWN". By tightening the fit you will have less errors of confusing one word with another in the file, but more "PLEASE REPEAT".

It is a simple experiment, and when you find the proper number for you, it should be POKEd before saving the Vocab file. It will then become a permanent part of the driver portion of that Vocab file.

If you have ever been in need of a good answer to the question, "What will that computer do?" the VOXBOX may well provide it. Besides the obvious demonstration programs (they work well and are guaranteed to boggle minds), it is not difficult to write a simple calculator program. Speak in a number, say the word "plus", then another number, say "equals" and see that the answer printed on the screen will cause the non-believer to start searching the floor for hidden buttons.

Automatic Bartender

I've also thought about an automatic bartender. Say the word "Martini" and a solenoid valve is opened permitting a shot of booze to fall into a plastic tube at the bottom of which is a glass of ice. The word "dry" opens the Vermouth solenoid valve (for a short time) into the same tube . . . not too difficult to attempt.

Light control, television control, door locks, drapery motors, movie screens, all are now in the realm of voice control, it may even be possible to do bookkeeping and write checks by voice! The applications are limitless. ■

✓See List of Advertisers on page 306

Just like the real McCoy...
...except for the price and label

Super Savings on Printers for the TRS-80® New Low Price

Faster than any Radio Shack Printer

CENTRONICS 704-11
(same as 703-9)

CENTRONICS PARALLEL INTERFACE
Ideal for TRS-80

180 Characters Per Second, Bi-Directional, Logic Seeking, with Tractor
Adjustable to 16"

FORMER LIST PRICE $2975
NOW ONLY **$1695**

Disc Drive Values for TRS-80

Your Choice
5¼" Disk Drives

Shugart, MPI-51, or Tandon
w/case & Power Supply

$499 VALUE
MMM PRICE ONLY

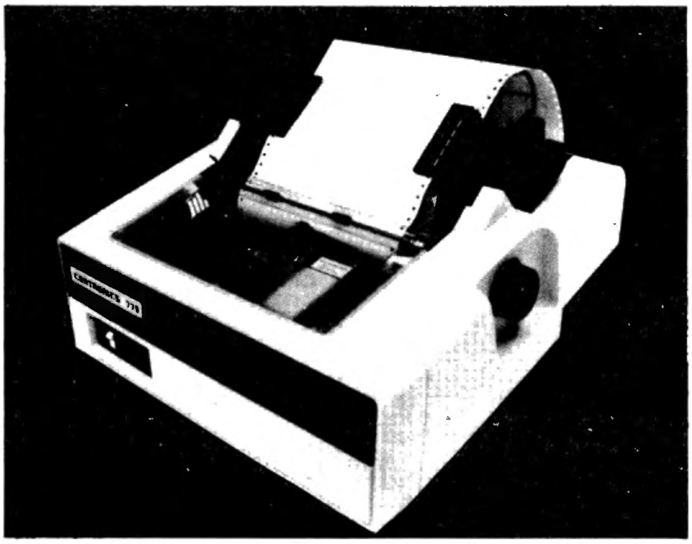
$329
2 for $638

CENTRONICS 730

Same as R.S. Line Printer II
LIST PRICE $795
(add $7.50 for shipping) NOW ONLY **$599**

CENTRONICS 779-2

Same as R.S. Line Printer I w/tractor
LIST PRICE $1350
(Ship freight collect) NOW ONLY **$799**



CENTRONICS 737

Same as R.S. Line Printer IV
Featuring Correspondence Quality and Proportional Spacing
LIST PRICE $995
NOW ONLY **$799**

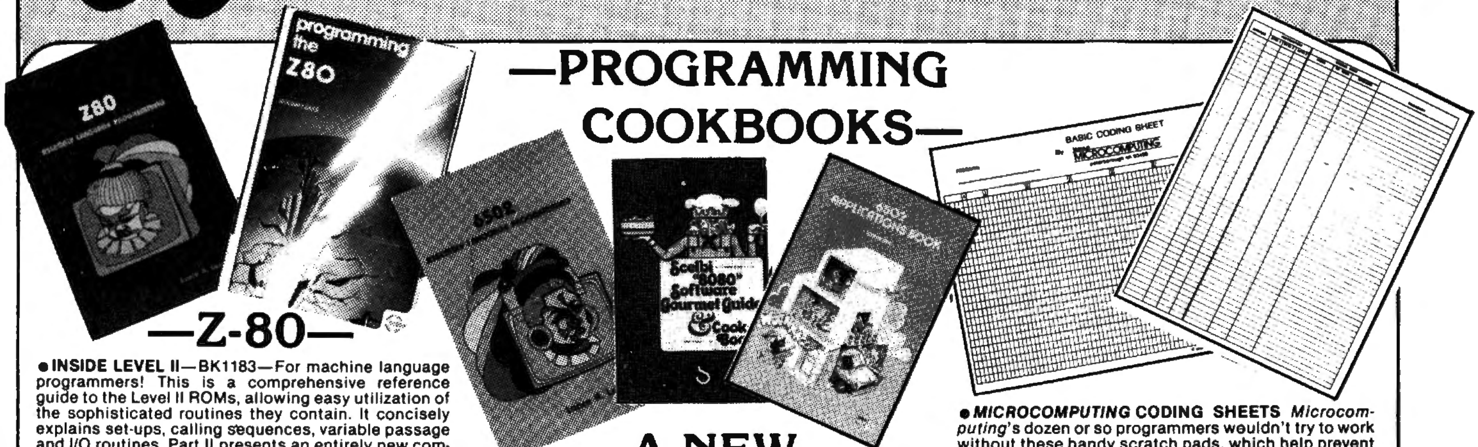
MiniMicroMart, Inc.

24

1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

80 microcomputing T.M. bookshelf

—PROGRAMMING COOKBOOKS—



—Z-80—

● **INSIDE LEVEL II**—BK1183—For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

● **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $16.99.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z-80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z-80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

TEST EQUIPMENT —LIBRARY—

● **VOL. I COMPONENT TESTERS**—LB7359— . . . how to build transistor testers (8), diode testers (3), IC testers (3), voltmeters and VTVMs (9), ohmmeters (8 different kinds), inductance (3), capacity (9), Q measurement, crystal checking (6), temperature (2), aural meters for the blind (3) and all sorts of miscellaneous data on meters . . . using them, making them more versatile, making standards. Invaluable book. $4.95.*

● **VOL. II AUDIO FREQUENCY TESTERS**—LB7360— . . . jam packed with all kinds of audio frequency test equipment. If you're into SSB, RTTY, SSTV, etc., this book is a must for you . . . a good book for hi-fi addicts and experimenters, too! $4.95.*

● **VOL. III RADIO FREQUENCY TESTERS**—LB7361—Radio frequency waves, the common denominator of Amateur Radio. Such items as SWR, antenna impedance, line impedance, rf output and field strength; detailed instructions on testing these items includes sections on signal generators, crystal calibrators, grid dip oscillators, noise generators, dummy loads and much more. $4.95.*

● **VOL. IV IC TEST EQUIPMENT**—LB7362—Become a troubleshooting wizard! In this fourth volume of the 73 TEST EQUIPMENT LIBRARY are 42 home construction projects for building test equipment to work with your ham station and in servicing digital equipment. Plus a cumulative index for all four volumes for the 73 TEST EQUIPMENT LIBRARY. $4.95.*

A NEW PROGRAMMING —SERIES—

The Microprocessor Software Engineering Series by John Zarrella provides common sense descriptions of advanced computer system topics for engineers, programmers and development managers. Each volume is a self-contained review of a software engineering topic, explaining fundamental concepts in easy-to-understand language and describing sophisticated software tools and techniques. Detailed glossary of technical jargon is included in each volume. This series will help you find the solutions to your software problems.

● **OPERATING SYSTEMS: CONCEPTS AND PRINCIPLES**—BK1193—Presents an overview of the basic operating system types, their components and capabilities. $7.95.*

● **WORD PROCESSING AND TEXT EDITING**—BK1194—Provides a firm basis for understanding word processing terminology and for comparing systems. $7.95.*

● **SYSTEM ARCHITECTURE**—BK1195 —Presents a detailed overview of advanced computer system design including object architecture and capability-based addressing. $9.95.*

—6502—

● **PROGRAMMING THE 6502 (Third Edition)**—BK1005—Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK**—BK1006—Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176—by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $16.99.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1055—by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

—8080/8080A—6800—

● **8080A/8085 ASSEMBLY LANGUAGE PROGRAMMING**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $15.99.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an in-depth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-based control system. $9.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1102—If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $9.50.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1075—Like its culinary cousin, *The 8080 Gourmet Guide*, this book by Scelbi Computing and Robert Findley describes sorting, searching and other routines—this time for the 6800 user. $12.95.*

—COOKBOOKS—

● **CMOS COOKBOOK**—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK**—BK1064—by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

● **TTL COOKBOOK**—BK1063—by Don Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

FLOPPY DISK STORAGE

CORVUS 11T 10 MEGABYTE HARD DISK Fully compatible with **TRS-80, APPLE II,** and **S-100**	*$4029
MPI MODEL B51 **TRS-80** plug-in compatible 5.25" single headed 40 track drive. Comes complete with case and power supply. **90 DAY WARRANTY.**	$321

BARE DRIVES
(No Case or Power Supply)

5" MODELS

MPI MODEL 51	40 tracks, 5 ms.	$254
MPI MODEL 52	Double headed, 80 tracks	$299
MPI MODEL 91	80 tracks	$299
MPI MODEL 92	160 Tracks, Double headed	$399

5.25" POWER SUPPLIES WITH CASE

STANDARD MINI DISK POWER SUPPLY WITH CASE	$80
DELUXE MINI DISK POWER SUPPLY w/CASE & extender card	$131
DUAL MINI DISK POWER SUPPLY w/CASE	$121
DISK DRIVE EXTENDER CABLE	$16

DRIVE CABLES

1 DRIVE TRS-80 CABLE	$24
2 DRIVE TRS-80 CABLE	$29
3 DRIVE TRS-80 CABLE	$35
4 DRIVE TRS-80 CABLE	$40

We'll make any cable! Call us with specs.

THE DOUBLER by PERCOM DATA Double density controller for **TRS-80 Model I.** With data separator. (Use your **PERCOM DISCOUNT** coupon)	$203
DATA SEPARATOR by Percom Data Stops read errors on inner tracks.	$29

SOFTWARE

NEWDOS (40 Track)	$49
NEWDOS + (40 Track)	$99
NEWDOS 80	$139
MICROSOFT FORTRAN	$85
Z80 Softcard	$305
Super Utility	$49
Make 80	$14
Electric Pencil (disk)	$130

10% off all **RADIO SHACK** software

We carry SBSG, Microsoft, personal software, Apparatt and many other brands. All at a discount.

030181

DISKETTES
(Prices per box of 10)

		VERBATIM	NEW SYNCOM	MEMOREX
5 1/4"	Single sided, single density	$31	$35	$28
	Single sided, double density	$40	$39	$32
	Double sided, single density	$48	N/A	N/A
	Double sided, double density	$48	$52	$40
8"	Single sided, single density	$41	$31	$33
	Single sided, double density	$51	$40	$41
	Double sided, single density	$61	$49	$55
	Double sided, double density	$61	$49	$50

When ordering diskettes, please give us your make and model number of computer or disk drive, or give us the brand and part number of the diskettes you are currently using.

5 1/4"	**LIBRARY CASE** holds 10 disks	$3.95
	BINDER HOLDER holds 2 disks	10 for $7.95
8"	**LIBRARY CASE** holds 10 disks	$4.95
	BINDER HOLDER holds 2 disks	10 for $19.95
FLOPPY SAVER		5 1/4" $14.95 8" $16.95 Refills also available!
16k **MEMORY KITS** 200 ns		$39.00

NEC UPD416C-2 prime dynamic RAM. Compatible with TRS-80, APPLE, or any computer specifying 4116 dynamic RAM.

HOW TO ORDER:
PHONE ORDERS: HOURS 10AM - 5PM CST MON - FRI
BY CHECK: Call in your order then send us a cashiers check or money order. This way we can have your order waiting for you when your check arrives. Personal checks require 10-14 days to clear.
BY CREDIT CARD: Just have your **MASTER CHARGE** or **VISA** or **AMERICAN EXPRESS** handy, we'll do the rest.
MAIL ORDERS: NEW ADDRESS 1467 South Michigan Ave., Chicago, IL 60605.
ALL MAIL ORDERS: Send your **NAME** and **STREET ADDRESS** (we ship UPS), **CITY, STATE, ZIP CODE AND TELEPHONE NUMBER** along with your order.
BY CHECK: Include a cashiers check or money order with your order. Personal checks require 10-14 days to clear. For your safety, mail your order via insured mail. **BETTER SAFE THAN SORRY.**
BY CREDIT CARD: Neatly type or print your **MASTER CHARGE** or **VISA** account number along with the expiration date and 4 digit interbank number (MC only).
IL. RES. ADD 6% SALES TAX

SHIPPING INFORMATION: We pay all UPS ground service charges (USA only, BLUE LABEL EXTRA) on all items except those denoted with an asterisk (*). (*) items must go airfreight or truck and can be sent freight collect.
FOREIGN ORDERS WELCOME, CALL OR WRITE.
ALL orders F.O.B. Chicago, Il.

WARRANTY INFORMATION: All Equipment carries a 90 day parts and labor warranty, unless otherwise noted. Warranty repair or replacement will be provided at no charge, excluding shipping costs. **10 DAY MONEY BACK GUARANTEE:** If within ten days you return merchandise in new and re-saleable condition we will refund your original purchase price. All refunds, exchanges or repairs include original cartons, packaging, and manuals and be in new and re-saleable condition and accompanied by your original invoice. A 10% re-stocking fee may be charged on returns at our discretion.
Not responsible for typographical errors
Prices subject to change without notice
Dealer Inquiries welcome
TRS-80™ TANDYCORP.

SAVE TIME. . .ORDER BY PHONE
TOLL FREE: 800-621-3229
IN ILLINOIS: 312-987-1024

8

MIDWEST COMPUTER PERIPHERALS

1467 South Michigan Av.
Chicago, IL. 60605

We pay UPS charges on most orders in continental U.S.

LIST OF ADVERTISERS

*This advertiser prefers to be contacted directly.

READER SERVICE

This card valid until July 31, 1981

Please help us to bring you a better magazine—by answering these questions.

I. What is your age?
- ☐ A. under 18
- ☐ B. 18-22
- ☐ C. 23-40
- ☐ D. 41-60
- ☐ E. over 60

II. What is your occupation?
- ☐ 1. Professional
- ☐ 2. Engineer
- ☐ 3. Data processing
- ☐ 4. Business
- ☐ 5. Education
- ☐ 6. Technician
- ☐ 7. Student
- ☐ 8. Other

III. What are your primary applications of your TRS-80 (check only two)?
- ☐ A. Business
- ☐ B. Games
- ☐ C. Home
- ☐ D. Education
- ☐ E. Scientific
- ☐ F. Control
- ☐ G. Music

IV. Your TRS-80, is it a
- ☐ 1. Level I
- ☐ 2. Level II
- ☐ 3. Model II
- ☐ 4. Don't own one yet

V. What peripherals do you have (check all that apply)?
- ☐ A. Expansion interface
- ☐ B. Disk
- ☐ C. Printer

VI. How much have you spent on hardware?
- ☐ 1. less than $500
- ☐ 2. $500-1,000
- ☐ 3. $1,000-2,000
- ☐ 4. $2,000-4,000
- ☐ 5. $4,000-6,000
- ☐ 6. more than $6,000

VII. How much have you spent on software?
- ☐ A. less than $100
- ☐ B. $100-250
- ☐ C. $250-500
- ☐ D. $500-1,000
- ☐ E. more than $1,000

VIII. What is your level of education?
- ☐ 1. Post-graduate
- ☐ 2. College
- ☐ 3. High school

IX. How many people read your copy of 80 Microcomputing?
- ☐ A. 1
- ☐ B. 2
- ☐ C. 3
- ☐ D. 4 or more

X. If you are not a subscriber please circle number 500.

80 Microcomputing • POB 2743 • Clinton IA 52735

Reader Service: To receive more information from any of the advertisers in this issue of 80 Microcomputing, circle the number on the postage-paid Reader Service Card that corresponds with the Reader Service number on the ad in which you are interested. You will find numbers, preceded by a ✓, near the logo of each advertiser. Complete the entire card, drop into a mailbox and in 4-6 weeks you will hear from the advertiser directly.

1	6	11	16	21	126	131	136	141	146	251	256	261	266	271	376	381	386	391	396
2	7	12	17	22	127	132	137	142	147	252	257	262	267	272	377	382	387	392	397
3	8	13	18	23	128	133	138	143	148	253	258	263	268	273	378	383	388	393	398
4	9	14	19	24	129	134	139	144	149	254	259	264	269	274	379	384	389	394	399
5	10	15	20	25	130	135	140	145	150	255	260	265	270	275	380	385	390	395	400
26	31	36	41	46	151	156	161	166	171	276	281	286	291	296	401	406	411	416	421
27	32	37	42	47	152	157	162	167	172	277	282	287	292	297	402	407	412	417	422
28	33	38	43	48	153	158	163	168	173	278	283	288	293	298	403	408	413	418	423
29	34	39	44	49	154	159	164	169	174	279	284	289	294	299	404	409	414	419	424
30	35	40	45	50	155	160	165	170	175	280	285	290	295	300	405	410	415	420	425
51	56	61	66	71	176	181	186	191	196	301	306	311	316	321	426	431	436	441	446
52	57	62	67	72	177	182	187	192	197	302	307	312	317	322	427	432	437	442	447
53	58	63	68	73	178	183	188	193	198	303	308	313	318	323	428	433	438	443	448
54	59	64	69	74	179	184	189	194	199	304	309	314	319	324	429	434	439	444	449
55	60	65	70	75	180	185	190	195	200	305	310	315	320	325	430	435	440	445	450
76	81	86	91	96	201	206	211	216	221	326	331	336	341	346	451	456	461	466	471
77	82	87	92	97	202	207	212	217	222	327	332	337	342	347	452	457	462	467	472
78	83	88	93	98	203	208	213	218	223	328	333	338	343	348	453	458	463	468	473
79	84	89	94	99	204	209	214	219	224	329	334	339	344	349	454	459	464	469	474
80	85	90	95	100	205	210	215	220	225	330	335	340	345	350	455	460	465	470	475
101	106	111	116	121	226	231	236	241	246	351	356	361	366	371	476	481	486	491	496
102	107	112	117	122	227	232	237	242	247	352	357	362	367	372	477	482	487	492	497
103	108	113	118	123	228	233	238	243	248	353	358	363	368	373	478	483	488	493	498
104	109	114	119	124	229	234	239	244	249	354	359	364	369	374	479	484	489	494	499
105	110	115	120	125	230	235	240	245	250	355	360	365	370	375	480	485	490	495	500

Name ____________________

Address ____________________

City ____________________ State ______ Zip ______

BOOKS

80 Microcomputing
Peterborough NH 03458

Please send me the following 80 Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Add $1 shipping/handling ______

Total ______

Enclosed $__________ ☐ Check ☐ M.O.
Bill: ☐ AE ☐ MC ☐ VISA

Card # ____________________

Exp date __________ Interbank # __________

Signature ____________________

Name ____________________

Address ____________________

City ____________________ State ______ Zip ______

5/81

Please allow 4–6 weeks for delivery

No C.O.D. orders accepted.

SUBSCRIPTION

80 Microcomputing POB 981
Farmingdale NY 11737

80 Microcomputing subscribers SAVE 40% Subscribe now.

Please allow 4–6 weeks for delivery

☐ New Subscription ☐ Renewal
☐ 1 year—$18 ☐ 2 years—$30 ☐ 3 years—$45
☐ Enclosed $__________ ☐ Check ☐ M.O.
Bill: ☐ MC ☐ VISA ☐ AE ☐ Me

Card # ____________________ Exp. date ______

Signature ____________________ Interbank # ______

Name ____________________

Address ____________________

City ____________________ State ______ Zip ______

Canada: $20 1 year only, US funds
Other foreign: $28 one year only, US funds

315B9

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 780 FARMINGDALE NY 11737

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing™

Subscription Dept.
POB 981
Farmingdale NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 80 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing™

Peterborough NH 03458

Att. Mail Order

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 217 CLINTON IA 52735

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

POB 2743
Clinton IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

Expand your TRS-80*

With The LOBO LX-80 Expansion Interface

Now you can realize all the power and potential of your TRS-80*, Model 1. If it's add-on memory you need, your LX-80 can accommodate up to four 5¼-inch, single- or double-density 35, 40 or 80 track mini-floppies, four 8-inch floppies (single or double sided), and up to eight Winchester fixed disk-drives (5¼". 8", 14").

LOBO's powerful new LDOS™ operating system, provided with your LX-80, allows for the use of any eight drives, in any combination, single or double density.

And there's more ... lots more. There are two parallel ports (standard), two serial ports (optional), a keyboard ROM override switch, and a 32K memory expansion (optional). Send for a free LX-80/TRS-80 cost performance comparison chart.

For the full story on how the LX-80 can expand your TRS-80, see your nearest LOBO dealer, or write or call:

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

LOBO DRIVES INT'L
354 South Fairview Ave.
Goleta, CA 93117
(805) 683-1576

15

See us at the N.C.C., booth #4211