80 Microcomputing 2/82 #26

February 1982
US $2.95
£1.80

80 microcomputing T.M.

the magazine for TRS-80* users

A WAYNE GREEN PUBLICATION

"Will Computers Replace Teachers?"

"Special Education Issue."

0 74470 65947 02

MEET THE MTI FAMILY OF LOW COST COMPUTERS

- 90 DAY WARRANTY
- LOCAL DEALER SERVICE
- MODEL III COMPATIBILITY
- CPM OPTION
- GREATER DISK STORAGE
- 4 Mhz OPTION

MOD III PLUS/140
We have taken the basic 16K Model III expanded the memory to 48K and added our MTI Double Density, Dual Disk Drive System. System is fully compatible with Radio Shack DOS and peripherals.
$1998.

MOD III/240
Same as the MOD III PLUS/140 but has double storage capacity, and 2 dual headed 40 track drives.
$2449.

MOD III/280
Has 1.5 megabytes of storage and utilizes 2 dual headed 80 track double density disk drives with DOS plus 3.3.
$2799.

MOD III/WINCHESTER
Our largest business computer system. 5.7 megabytes of storage. Includes a 5 megabyte Winchester hard disk drive and 80 track dual head disk drive used as back up and for conventional floppy disk operation.
$5399.

CPM/80 x 24 Display $699.

4 Mhz Speed-up enhancement $149.

MTI AUTHORIZED SALES AND SERVICE CENTERS

Phoenix, AZ	(602) 244-9739
Sierra Vista, AZ	(602) 458-2479
Tempe, AZ	(602) 839-0546
Tucson, AZ	(602) 323-9391
Anaheim, CA	(714) 773-0240
Covina, CA	(213) 332-4088
Goleta, CA	(805) 967-7628
Huntington Bch., CA	(714) 842-1345
Inglewood, CA	(213) 673-3295
Lancaster, CA	(805) 942-5747
Northridge, CA	(213) 886-9200
Port Hueneme, CA	(805) 985-2329
Redondo Bch., CA	(213) 370-5556
San Diego, CA	(714) 275-4243
San Jose, CA	(408) 946-1265
Santa Cruz, CA	(408) 427-0836
Vallejo, CA	(707) 694-7550
Walnut, CA	(714) 594-8311
Groton, CT	(203) 445-5166
Hollywood, FL	(305) 981-1011
Norcross, GA	(404) 449-8982
Blackfoot, ID	(208) 785-1497
Collinsville, IL	(618) 345-5068
Shreveport, LA	(318) 865-7583
Anoka, MN	(612) 427-5783
Joplin, MO	(417) 781-1748
Missoula, MT	(406) 549-9715
Raleigh, NC	(919) 755-1175
Grand Fords, ND	(701) 772-7848
Jericho, NY	(516) 997-8668
Manhasset, NY	(516) 869-8335
Troy, NY	(518) 273-8411
Maumee, OH	(419) 893-4288
Dallas, TX	(214) 247-6679
Cheyenne, WY	(307) 632-9132
Mexicali, BC	(714) 357-4717
OVERSEAS	
Australia	3877-6946
Belgium	1663-2452
Rep. of South Africa	2145-1047

MICROCOMPUTER TECHNOLOGY INC.
3304 W. MACARTHUR, SANTA ANA, CA 92704
(714) 979-9923 • TELEX 6780401 TABIRIN

Call or write for free brochure:
U.S. PRICES, F.O.B. SANTA ANA
CALIFORNIA AND MAY VARY BY AREA.

MasterCard VISA

28

TRS-80* COMPUTING EDITION

©1981 Percom Data Co., Inc. | The Percom Peripheral | 35 cents

Percom's DOUBLER II™ tolerates wide variations in media, drives

GARLAND, TEXAS — May 22, 1981 — Harold Mauch, president of Percom Data Company, announced here today that an improved version of the Company's innovative DOUBLER™ adapter, a double-density plug-in module for TRS-80* Model I computers, is now available.

Reflecting design refinements based on both theoretical analyses and field testing, the DOUBLER II™, so named, permits even greater tolerance in variations among media and drives than the previous design.

Like the original DOUBLER, the DOUBLER II plugs into the drive controller IC socket of a TRS-80 Model I Expansion Interface and permits a user to run either single- or double-density diskettes on a Model I.

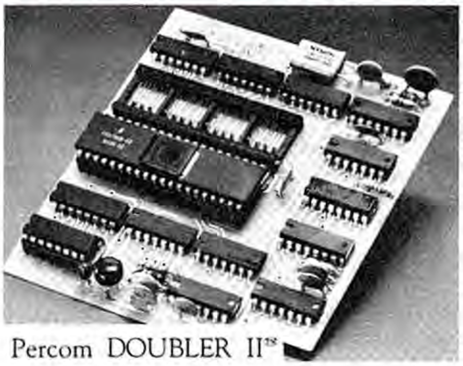
With a DOUBLER II installed, over four times more formatted data — as much as 364 Kbytes — can be stored on one side of a five-inch diskette than can be stored using a standard Tandy Model I drive system.

Moreover, a DOUBLER II equips a Model I with the hardware required to run Model III diskettes.

(*Ed. Note:* See "OS-80™: Bridging the TRS-80* software compatibility gap" elsewhere on this page.)

The critical clock-data separation circuitry of the DOUBLER II is a proprietary design *called a ROM-programmed digital phase-lock loop data separator*.

According to Mauch, this design is more tolerant of differences from diskette to diskette and drive to drive, and also provides *immunity to performance degradation caused by circuit component aging*.



Percom DOUBLER II™

Mauch said "A DOUBLER II will operate just as reliably two years after it is installed as it will two days after installation."

The digital phase-lock loop also *eliminates the need for trimmer adjustments typical of analog phase-lock loop circuits*.

"You plug in a Percom DOUBLER II and then forget it," he said.

The DOUBLER II also features a refined Write Precompensation circuit that more effectively minimizes the phenomena of bit- and peak-shifting, a reliability-impairing characteristic of magnetic data recording.

The DOUBLER II, which is fully software compatible with the previous DOUBLER, is supplied with DBLDOS™, a TRSDOS*-compatible disk operating system.

The DOUBLER II sells for ~~$219.95~~, including the DBLDOS diskette. NOW $169.95!

Owners of original DOUBLERs may purchase a DOUBLER II upgrade kit, without the disk controller IC, for $30.00. Proof of purchase of an original DOUBLER is required, and each DOUBLER owner may purchase only one DOUBLER II at the $30.00 price.

The Percom DOUBLER II is available from authorized Percom retailers, or may be ordered direct from the factory. The factory toll-free order number is 1-800-527-1222.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

258

All that glitters is not gold

OS-80™ Bridging the TRS-80* software compatibility gap

Compatibility between TRS-80* Model I diskettes and the new Model III is about as genuine as a gold-plated lead Krugerrand.

True, Model I TRSDOS* diskettes can be *read* on a Model III. But first they must be converted and re-recorded for Model III operation.

And you *cannot write* to a Model I TRSDOS*diskette. Not with a Model III. You cannot add a file. Delete a file. Or in any way modify a Model I TRSDOS diskette with a Model III computer.

Furthermore, your converted TRSDOS diskettes cannot be converted *back* for Model I operation.

TRSDOS is a one-way street. And there's no retreating. A point to consider before switching the company's payroll to your new Model III.

Real software compatibility should allow the *direct, immediate* interchangeability of Model I and Model III diskettes. No read-only limitations, no conversion/re-recording steps and no chance to be left high and dry with Model III diskettes that can't be run on a Model I.

What's the answer? The answer is Percom's OS-80™ family of TRS-80 disk operating systems.

OS-80 programs allow *direct, immediate* interchangeability of Model I and Model III diskettes.

You can run Model I single-density diskettes on a Model III; install Percom's plug-in DOUBLER™ adapter in your Model I, and you can run double-density Model III diskettes on a Model I.

There's no conversion, no re-recording.

Slip an OS-80 diskette out of your Model I and insert it directly in a Model III.

And vice-versa.

Just have the correct OS-80 disk operating system — OS-80, OS-80D or OS-80/III — in each computer.

Moreover, with OS-80 systems, you can add, delete, and update files. You can read *and write* diskettes regardless of the system of origin.

OS-80 is the original Percom TRS-80 DOS for BASIC programmers.

Even OS-80 utilities are written in BASIC.

OS-80 is the Percom system about which a user wrote, in Creative Computing magazine, ". . . the best $30.00 you will ever spend."†

Requiring only seven Kbytes of memory, OS-80 disk operating systems reside completely in RAM. There's no need to dedicate a drive exclusively for a system diskette.

And, unlike TRSDOS, you *can* work at the track sector level, defining and controlling data formats — in BASIC — to create simple or complex data structures that execute more quickly than TRSDOS files.

The Percom OS-80 DOS supports single-density operation of the Model I computer — price is $29.95; the OS-80D supports double-density operation of Model I computers equipped with a DOUBLER or DOUBLER II; and, OS-80/III — for the Model III of course — supports both single- and double-density operation. OS-80D and OS-80/III each sell for $49.95.

429

Circuit misapplication causes diskette read, format problems.

High resolution key to reliable data separation

GARLAND, TEXAS — The Percom SEPARATOR™ does very well for the Radio Shack TRS-80* Model I computer what the Tandy disk controller does poorly at best: *reliably* separates clock and data signals during disk-read operations.

Unreliable data-clock separation causes format verification failures and repeated read retries.

CRC ERROR–TRACK LOCKED OUT

The problem is most severe on high-number (high-density) inner file tracks.

As reported earlier, the clock-data separation problem was traced by Percom to misapplication of the internal separator of the 1771 drive controller IC used in the Model I.

The Percom Separator substitutes a *high-resolution* digital data separator circuit, one which operates at 16 megahertz, for the low-resolution one-megahertz circuit of the Tandy design.

Separator circuits that operate at lower frequencies — for example, two- or four-megahertz — were found by Percom to provide only marginally improved performance over the original Tandy circuit.

The Percom solution is a simple adapter that plugs into the drive controller of the Expansion Interface (EI).

Not a kit — some vendors supply an untested separator kit of resistors, ICs and other paraphernalia that may be installed by modifying the computer — the Percom SEPARATOR is a fully assembled, fully tested plug-in module.

Installation involves merely plugging the SEPARATOR into the Model I EI disk controller chip socket, and plugging the controller chip into a socket on the SEPARATOR.

The SEPARATOR, which sells for only $29.95, may be purchased from authorized Percom retailers or ordered directly from the factory. The factory toll-free order number is 1-800-527-1222.

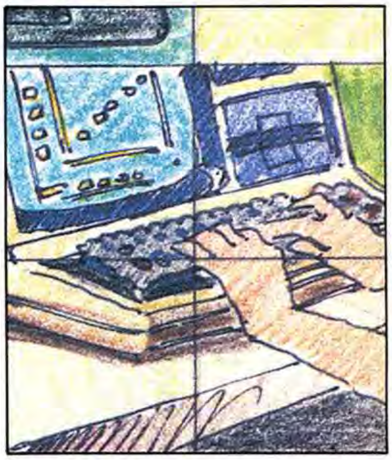
Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

1

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

PERCOM DATA COMPANY, INC. 11220 Pagemill Road Dallas, Texas 75243 (214) 340-7081

80 Contents



PUBLISHER/EDITOR
Wayne Green
EXECUTIVE VICE PRESIDENT
Sherry Smythe
ASSISTANT TO PRESIDENT
Matt Smith
ASSOCIATE PUBLISHER
Edward Ferman
ASSISTANT PUBLISHER
Jeff DeTray
MARKETING
Debra Boudrieau, Director; Ginny Boudrieau, Bulk Sales Manager
ADVERTISING
Penny Brooks, Coordinator; John Gancarz, Hal Stephens, Susan Martin: Sales. Marcia Stone, Office Manager. New England Advertising Representative: John A. Garland, Garland Associates, Inc., Box 314 SHS, Duxbury, MA 02332. 617-934-6464.
DESIGN
Clare McCarthy, Manager; Elaine Cheever, Corporate Designer. Denzel Dyer, Howard Happ, Laurie MacMillan, Joyce Pillarella, Diana Shonk, Susan Stevens, Donna Wohlfarth: Consultants

PRODUCTION
Nancy Salmon, Manager; Michael Murphy, Dennis Christensen: Assistants. Frances Benton, Fiona Davies, Sandra Dukette, Theresa Ostebo, Dianna Ritson, Deborah Stone, Irene Vail, Judy Wimberly. Patty MacKowsky/Allen, Ad Coordinator; Steve Baldwin, Bruce Hedin, Jane Preston: Advertising Production.
PHOTOGRAPHY
William Heydolph, Paul Babich, Bryan Hastings, Thomas Villeneuve
TYPESETTING
Sara Bedell, Debbie Davidson, Michele DesRochers, David Hayward, Beverly Jackson, Stephen Jewett, Elizabeth Lockhart, Anne Rocchio, Kelly Smith, Karen Stewart

Paid Audited Circulation

80 Microcomputing (ISSN -0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-9471. Second class postage paid at Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada, $27—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor: Graymar Data Services, Ltd., #4 258 E. 1st Ave., Vancouver, BC V5T 1A6. Foreign subscriptions (surface mail), $35—one year only, U.S. funds. Foreign subscriptions (air mail), please inquire. 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. 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.

Cover and illustrations on pages 52 and 94 by Jay Connelly

*TRS-80 is a trademark of Tandy Corp.

* = program is on LOAD 80 tape.

Proof Notes
the editors look at the issues

Subscriptions:
Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *80 Microcomputing*, Subscription Department, PO Box 981, Farmingdale, NY 11737.
Change of Address: Send old label or copy of old address and new address to *80 Microcomputing*, PO Box 981, Farmingdale, NY 11737. Please give eight weeks advance notice.
Microfilm: This publication is available in microform from University Microfilms International. United States address; 300 North Zeeb Road, Dept. P.R., Ann Arbor, MI 48106. Foreign address; 18 Bedford Row, Dept. P.R., London, WC1R4EJ, England.
Dealers: Contact Ginny Boudrieau, Bulk Sales Manager, *80 Microcomputing*, Pine St., Peterborough, NH 03458. (800) 258-5473.

Education—ever since persons other than monkish ascetics recognized its value, waves of controversy have surrounded it. Who should be educated? Should everyone meet some minimum educational requirement? Whose version of knowledge should be taught? These burning questions have been raised and dealt with again and again, often with different results. Twenty years ago most of us never would have dreamed the question of creationism versus evolution would raise its head again; today, the news is full of accounts of states reinstituting the teaching of creationism as a viable scientific theory.

Where microcomputers enter the realm of education we find new concerns: What teaching approach is most effective, and is CAI (Computer Assisted Instruction) really a teaching method? Can a child learn from a machine, a tape or a video tube, or only from another human being?

Most questions about education find answers through custom, the rising needs and values of each society, and the enterprising experiments of a few far-seeing individuals. Often the questions follow cycles, returning time and again to the same solutions.

Currently we are faced with a new situation: Computers have become standard equipment in most businesses. At the root of many jobs, they touch the lives of everyone in some way. They have become deeply entrenched in our society, and yet they are nearly invisible. Most of us fail to make the connection between computers and our bank balances, our grocery bills, our mail delivery, our newspapers. We do not realize that the word processors at our offices are a form, though limited, of *computer.* We do not comprehend that even the most basic of businesses, such as farming, depend on a computer at least occasionally.

Computer literacy has become a basic skill, but that truth has not reached the general population. Most importantly, it has not reached the educators.

Many school teachers are afraid of computers. They don't know how to run them and fear they will be replaced by one. Often students know more about the mysterious machine than the teachers. Documentation is written for programmers and experienced users; software doesn't fit the teacher's needs. The new school micro may sit collecting dust for a long time.

We do not believe a computer will ever replace teachers; a teacher could never be replaced by a machine, especially where younger children are involved. But a computer *can* be a strong and effective teaching tool in the hands of a competent teacher. It can provide individual instruction at any student's optimum learning rate, removing some of the disadvantages of increasingly overcrowded classrooms. It can provide advanced instruction for those students whose learning rate surpasses his or her classmates or the teacher's knowledge. It can provide instruction in topics of individual interest that might otherwise be unavailable. It can provide, most patiently, all the drill exercises a student may require. It can bring instruction to a student unable to attend class. Above all, it can provide the computer literacy our children will need to function effectively in *their* society.

John Mello has looked into education and the impact the computer has there. He is ready to share his garnerings in "The Future in Miniature."

Worlds of a different sort open for the handicapped who are equipped with computers. Finally able to communicate with relative ease, they can find and hold a useful, well-paying job. They can attend public school. They can create art. Handicaps are no longer a barrier to a normal life when a computer can fill in for physical capabilities.

Computer applications for the handicapped are in their birth stages. Kerry Leichtman has written an overview of the subject, with some suggestions for programmers wishing to add to this field, in "Making More Possible."

Debra Marshall
Managing Editor

The left bracket, [, replaces the up arrow, ↑, used by Radio Shack to indicate exponentiation, on *80 Micro's* printouts. When entering programs published in *80 Microcomputing* you should make this change.
80 formats its program listings to run 64 characters wide, the way they look on your video screen. This accounts for the occasional wrap-around you will notice in our program listings. Don't let it throw you, particularly when entering assembly listings.

Reader Service for facing page ✓4→

INTELLIGENT PRINTER INTERFACE

Free Your Computer from the Mundane Task of Printing

Imagine being able to use your computer seconds after beginning an extensive printout.

Visualize your printout with page breaks, page numbering and titles, margins of your choice, indented carryover lines, on any size paper!

Appreciate the time and money you will save by not waiting on your printer.

SooperSpooler, a buffered printer interface, maintains control over your printer while you go on using your computer for more productive activities. Eliminate waiting while your printer pecks through a long document. SooperSpooler accepts information from your computer at up to 2500 characters per second and feeds it to your printer as fast as it can handle it—without using any of your computers memory or time! As soon as SooperSpooler has stored your document in its buffer, control of your computer is returned to you.

SooperSpooler features include:

- 16K Memory—Will handle most of your printing jobs (expandable, see options)
- Buffer Status Readout—Lets you know just how much data is stored
- Space Compression—Makes the best use of memory on columnar documents
- Pagination—Eliminates printout on page perforations
- Page Stops—For single sheet printouts
- Headers and Page Numbering—Give your listing a professional look
- Indentation on Carryover Lines—Easy to find the beginning of a line
- Self Test Routine—You instantly know that all is well
- All Features Software Controllable—Your program can take over
- Plugs into Most Computer Systems—Standard cables available
- **$349.00!**—16K parallel I/O unit

Options:

- Serial Board—$95.00—Gives you the option of any combination of serial or parallel input or output. Can also be used for modem transmission.
- Memory Expansion—$159.00—Additional 46K for a total of 62K
- Cables—Available per your application.

SooperSpooler™ by Compulink—The missing link that gives your microcomputer mainframe printing.

COMPULINK
CORPORATION

1840 Industrial Circle, Dept. A
Longmont, CO 80501
(303) 651-2014
Order line: 800-525-6705

Dealer inquiries welcome

We accept
- VISA
- MasterCard
- Checks
- Money Orders
- COD

Add $3.00 per order for postage and handling
COD add $3.00
Colorado residents add sales tax

Prices and Specifications Subject to Change Without Notice

80 REMARKS
by Wayne Green

"Change means people have to cope with something new—they have to think."

While kids have no problem adapting to computers, older people, and that includes teachers, are generally apprehensive. Considering that it is a normal human reaction to resist change, this is understandable. Change means people have to cope with something new—they have to *think*. People will go to extremes to avoid having to think, thus keeping Robin Williams and the Green Bay Packers in the chips.

Kids, not having learned enough to cope even with the normal world, are more into having to think, so being faced with a computer is not much different from being faced with a car motor or a video recorder. Note that older people have one hell of a time coping with video recorders and car motors too, unless they are a member of the small minority of elderly who enjoy new things.

As a general rule I think you'll find that most teachers are particularly set in their ways. It goes along with the psychological reasons they are teachers. Some teach because they want to, others because they are used to the protective environment of academia and prefer that to trying to make a break into business. By the time one has gone through high school, college, and then on to a masters and doctorate degrees, one is so used to being in school it is usually too late to make a career change. The adventurous years have gone by, so it's easier to stay in school, teaching. This type of person, normally, is not an adventurous one and is a prime candidate for trying to resist change. . . and computers.

No Clear Role

The main use of microcomputers in schools has been merely for the purpose of "computer literacy." Indeed, in the educational environment, the microcomputer has been more of a tool looking for a use than a solution to any problems. That is why the business use of computers has been growing at a much faster rate, where there *are* important uses for the contraption, uses which save money and increase profits.

In schools there has been a lot of jawboning about the use of small computers, but the brutal fact is that there really haven't been a lot of practical applications yet. It *is* important that students get familiar with microcomputers and thus not fear them, so their use for computer literacy is a valid one, as long as the investment in capital equipment is not too heavy.

Computers *do* have some use for remedial work. I've been disappointed with the short shrift they've had in this application. I think a bunch of programs to help grade school and high school students cope with math, English, and so on would be far less expensive for parents than tutors. The computers are a lot more patient and friendly than most tutors, and could save enough in a short while to pay for themselves. Of course, if you are not a parent and have not had to write checks for tutors, you may be insulated from this problem.

Although this is a valid use for computers, it is not one which teachers are likely to embrace. First, it is not the teachers who have to pay the tutor bills, it is the parents. Second, it is often the teachers who would lose the tutoring money, giving them a strong reason not to recommend a computer as their replacement in the tutoring business.

The development of tutoring programs has been virtually nil, for obvious reasons. It would have to be these same teachers who are afraid of computers and who see them as a threat to tutoring income who would break ranks and write the needed programs. Get out the tar and feathers. Then there is the problem of selling the programs, which is not a small one. Without the recommendations of teachers, how are parents going to become convinced that a microcomputer is going to get their little darling through the course? At $2,500 in one whack, plus the needed programs, it seems easier to pay the amount in dribs (and possibly drabs), even though they end up without the computer when it's all over.

All over. Ha! There'll be more tutoring next year, dribbling more out. No, there *is* a good market there if anyone can crack the pattern.

The Spectre

Another reason for teachers' apprehension is the concept that computers are. in the end, going to take over much of the teaching role, leaving little for teachers to do other than retrain for business. Heck, if they were interested in business they would have gone that route when they were younger. On some level they realize they have traded the possibility for fame and fortune for a lifetime of penury and debts, with little thanks. But after the adventurous years, which were left behind during their post-graduate courses and extension courses, they feel it's too late to jump into the "rat race." Just as printers fought automated typesetting for a generation, teachers are gearing up to fight off computers.

You know, it wasn't until IBM came out with their Composer that the back of the typographical unions was broken. This system was so inexpensive even a small publisher could buy one and set his own type. Before that the unions protected the good old hot metal Linotype machine and all type had to be set by Linotype operators, who worked for printers. If you've never watched the operation of a hot metal Linotype machine, you've missed a wonderful part of the printing business.

Now, while I don't see a lot of use for the present generation of computers (or microcomputers) in education, other than for literacy, I do see some applications coming down the line which will, I believe, revolutionize education. There *is* a computer in the future—one which will be designed for educational use, and it will lower the cost of education tremendously—as well as making it available in depths not imagined before.

The present educational system is appallingly wasteful. We've gotten the idea that a kid's time is of no value, so we think nothing of wasting much of it. Is it any wonder kids have no concept of the value of their time and feel free to waste it further themselves? That's what they are taught in school, so it is natural.

One of the uses for schools is for babysitting. It's a place to keep the kids so they will not have to be cared for at home. At the younger ages they are put in child care

centers, where they are often put on the floor to watch television all day. . . giving them a good start toward becoming illiterate gibbering idiots. Then, when we finally put them in school, we bus them for a half hour and have them sit around in "study" rooms, or in classes where half of them are bored out of their gourds at how slow things are going, while the other half miss most of what is going on because it is too fast for them. With thirty to sixty in a class, the occasional finger from the teacher to get up in front of the group and display their ignorance strikes terror into little hearts.

The New Education

Now, let's suppose we have a video disk player which is connected to a microcomputer—or perhaps has the micro built in—which is what will really happen. (By the way, I see that Texas Instruments has announced an interface between their computer and a video disk player.) Okay, with an interactive system we can program the computer to present the material on the disk, in full color and motion, at the speed the student needs. Not so fast he misses much of it, and not so slow that it is boring. The computer can do this, much like having an individual teacher available for each student.

With the teaching system instead of a teacher, the student no longer has to be brought to the teacher, with that associated waste of time. No more threat of being beaten up in the toilets, exposure to drugs, or other growing miseries of school life. The courses can be taken at home and at the speed desired by the student. The student can go as far as he wants, with an almost infinite number of courses available.

For that matter, the student can be of any age and the courses can be in cooking, marketing, sales rep management, archaeology, cuniform, and so on. Courses available on disks would probably rent for $25 or so, bringing *anything* of interest to anyone willing to spend the time. You want to learn French? Fine, a disk from the rental store will be delivered in a few hours. We may even see the time when the information will come via radio or cable or laser via glass tubes, and go onto a read/write disk system, getting around the need for an investment in program disks.

You want to retrain for some other kind of work? No problem, just get the courses you need and get busy. With more and more manufacturing being automated and robotized, people will be able to work from home doing advertising, marketing, accounting and other white-collar functions. It is largely the problem of communication which forces workers to drive to work and waste their time in offices.

Good Lord! Do I mean that other than for making the disk programs we aren't going to need teachers some day in the future? Yes and no. . . if I may equivocate. We won't be needing them for about 80 percent of the role they've been playing for the last few hundred years. Instead of listening to a teacher who knows something about a subject, we will be able to learn from the top people in the field, people who are not only knowledgeable, but who are showmen and make the subject intensely interesting.

> *"No number of microcomputers are going to be able to simulate the learning experience of working with hardware."*

The world is changing ever faster, so it is going to take a new educational medium such as this to keep up with the changes. Our present teachers, for the most part frozen into stasis upon their own graduation, have done poorly in keeping abreast of change. . . that old enemy change.

But there is going to be a need for teachers—a serious need. I think we are going to have more need for teachers than ever before, but for a different function than most have provided so far. You see, with the movement of the regular school courses into the home via teaching machines, we are still going to be left with a need for teaching *skills.*

No number of microcomputers are going to be able to simulate the learning experience of working with hardware. If you want to learn to use a drill press, you really have to work with a drill press, not with a computer. And, at the cost of drill presses, it becomes worthwhile to bring the students to the drill press instead of the other way around.

Apply that concept to a wide range of woodworking machines, metalworking machinery, electronic repair laboratory equipment, flying, swimming, hang gliding, driving, skiing, foundry work, surveying, and so on. We will need labs with equipment if we are going to teach skills, and we are going to need experienced teachers.

During my schooling I've had courses in woodworking, metalworking, foundry, surveying, art, drafting, boxing, swimming, electronic repairs, strength of materials testing, a wide variety of chemistry and physics lab experiments, fencing, photo darkroom work, picture taking, singing, speed reading and so on. This barely touches the skill courses which are possible. In labs I've worked on motors, generators, the repair and use of all kinds of electronic equipment, antenna designing and testing, etc. I can't think of any way that computers are going to replace hands-on learning with these instruments. You really have to use a microscope, a spectroscope, an oscilloscope, and so on, to develop a skill with them.

When I talk with kids today I find they have had few of the courses I was exposed to. Something has changed in education. Heck, I remember taking a course in picture composition when I was in the fourth grade. And one in reading music at about the same time. I wonder if they are teaching those courses at those levels anywhere these days. Even in the public schools in Brooklyn they had music appreciation courses where we became familiar with and could identify a hundred pieces of classical music.

With computerized disk courses our kids could expand their worlds beyond arithmetic, geography and English in the lower grades. And they could, by going to school, learn a wide variety of skills. Why not teach juggling? Magic tricks? Swimming? Gymnastics? Bowling? Flying? Whittling? Painting? There are dozens, even hundreds of skills. Each skill learned enables one to cope better with the world and expand horizons. You learn to ride a horse by getting on one and having a damned good instructor. You aren't going to learn from a computer.

One Man's Vision

I think it's clear that computerized and automated education is coming. It is going to be vastly better than the present system—far more efficient, and fun. The key to education is that it *must* be fun. The

courses have to be presented by enthusiastic people and made interesting.

It is also clear that though this is going to bring about some big changes in education, we are still going to need teachers and schools. In fact, the investment in schools is going to be more than ever, as we stock up on the hardware needed to teach skills. We are going to have to develop some systems for getting students to the hardware with efficiency so a drill press will be used by students as long as it takes to be learned. In cities this is no problem, but in suburban areas we may want to move the hardware around from town to town to spread out the training. This could take the form of mobile hardware units—labs, if you will.

It is far too expensive for one town to have all the electronic test equipment it takes to cope with television or computer repairs, but if the equipment is built into a mobile lab it could be taken to a school and used for a few weeks for that group of students and then moved to another school, so more students could use the equipment. The teachers would travel with the equipment. This would keep the costs for elaborate set-ups down, spreading them over a wider range of students. Indeed, the labs could be part of extension courses in the evenings and not just used days, thus getting about 14 hours per day use from them, further reducing the cost per student for the lab.

When you get into medical labs this gets even more important, for the costs of some of the new medical electronic equipment is astronomical. It's something to think about.

When?

Are we going to wait for the slow process of evolution to bring about the changes, or are we going to start pushing for them to happen faster? I'm a rabble-rouser myself, so you can be sure I will be out there pushing.

In line with that, I have been working with the FCC to set up a national goal of getting both ham clubs and computer clubs started in every high school in the country. By doing this we will expose more of our impressionable kids to the fascination of electronics, communications and computers, and a rather high percentage of them will then be attracted into careers in these fields. Fine. . . but what about the colleges? They are ill-equipped to provide a modern education in these fields right now. By putting pressures on the colleges through turning out hundreds of thousands of kids interested in more education, I think we can bring about the needed changes. We'll see.■

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™

MODELS I & III . . . $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

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!**

MTC CALCS-IV™ $39.95
For Models I & III $39.95
For Model II $59.95

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 Models I & III $19.95
For Model II $29.95

THE COMPLETE MTC AIDS-III™ PACKAGE

Includes MTC AIDS-III™ CALCS-III™ and MERGE-III™

A comprehensive system at a competitive price!

MODEL I & III $99.95
MODEL II $149.95

Add $25 for CALCS-IV™

MTC CALCS - III™

Models I & III $24.95
Model II $39.95

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

- **Columnar Headings**
- **Optional Indentation**
- **User-specified Columnar Totals**
- **Columnar values computed using constants and/or column values**
- **Balance forward calculations**
- **Use for accounting, inventory, financial and other numeric-based systems.**

EPSON PRINTERS

DISK DRIVES

DISKETTES

BOOKS and more!!

AIDS/P™ IS COMING!

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.

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

PRICES IN EFFECT THRU **February 28, 1982**
Prices, Specifications, and Offerings subject to change without notice.
8202

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 6.5% sales tax.**

80 INPUT

"There is more to life than TRS-80 hardware and software."

Extra DOSPLUS

Your October issue review of DOSPLUS prompted me to write concerning what I consider to be a highly undesirable business practice concerning that product.

LNW Research refuses to sell their LNW80 computer without also forcing the purchaser to pay for a copy of DOSPLUS. As a result I will soon have over $300 of unneeded and unwanted software littering my office (when my third LNW80 arrives).

I agree with your reviewer that DOSPLUS is a superior operating system, certainly by comparison with TRSDOS. However, I am strongly of the opinion that for any application which involves extensive file manipulation, NEWDOS80 simply has no competition.

It is true that for a time DOSPLUS was the only serious choice if you were operating double density. With NEWDOS80 Version 2.0 that is no longer the case, nor is Model I-Model III compatibility a problem.

My business applications programs are all written using the variety of file handling procedures which NEWDOS affords, and I simply refuse to use field-item files when there is any way not to. As a result I have no need for DOSPLUS, and I object strongly to being forced to buy it.

Protests to LNW have produced no result. Maybe shouting louder will help. Anybody want a copy of DOSPLUS, Cheap? Or two. . . or three. . . ?

Bob Penny
Small System Design
Boulder, CO

LNW Responds

I certainly hope that our policy of supplying a DOSPLUS 3.38 Operating System Disk absolutely free with our LNW80 Computer System has not caused you much grief. I would like to point out that we do not mention in our advertisements or our published literature that the DOSPLUS is part of the complete Computer System price. We supply DOSPLUS for the following three reasons:

First, to eliminate possible support problems when LNW80 owners attach non-LNW Research peripherals to the LNW80 and use non-LNW Research supplied software. We needed to eliminate at least one unknown (software).

Second, to supply an excellent quality operating system that takes advantage of our 4 MHz CPU speed, five inch or eight inch single/double-density disk controller, built-in RS232, etc. Although NEWDOS80 2.0 also satisfies our requirements and is certainly an excellent system, we cannot recommend it for a first-time computer owner. DOSPLUS is by far the simplest operating system to use.

Third, as a gift for those who have purchased the LNW80 Computer System.

You have, Mr. Penny, caused us to take a second look at our practice of supplying the DOSPLUS for free. We do, however, stand by our original decision. If you have any other comments or suggestions we look forward to hearing from you.

Kenneth M. Woog, President
LNW Research Corporation
Tustin, CA

Brown Off Base

I read every word of your editorials and think you bat 1000. Concerning your technical editor, Chris Brown, I would rate him a generous 2½ for his editorial article, "Computerization of the Workplace" (*80 Microcomputing*, November 1981). What is worse is that it rated top billing on your Contents page.

First, there is nothing in it that concerns the TRS-80. Second, friend Brown devotes a lot of space to the Air Traffic Controller's plight and suggests that they are examples of "Computer Types." They are Radarscreen specialists and the computer only serves as a back-up. The stress they complain of is due to the Civil Service system where you get no credit for a good job but can be criminally prosecuted for a goof.

If friend Brown had done his homework he would know this. Instead he listened to a couple of PhD. types from M.I.T. and the Ford Foundation.

I agree that the unions are going to successfully recruit the data processors and programmers because they are prime targets. First, they are specialists on a dead-end street. Few if any will be picked for top communication skills as evidenced by their futile attempts at documentation. But more important, they are a breed that talks technically down to their bosses. Top management regards the data processors and their staff as only technicians who know how to accumulate data but not how to employ it.

Going back to Chris Brown's editorial-type article, take out the misinformation about the air traffic controllers and what has he said? Not a damn thing and surely nothing that involves the TRS-80. Hopefully you will delegate articles such as Brown's to File 13. It is a great magazine and please keep it that way.

Harry Wenger
San Luis Obispo, CA

Chris Brown Replies

Hey Wenger, what's your beef? There's nothing wrong with an occasional "editorial-type" article in a 400-page computer magazine, especially if it gets a guy like you to look beyond his nose and think about some of the effects computers are having on society. (And in this case, you have to admit it worked!)

As far as air traffic controllers go, it was not my intention to imply that they had a working knowledge of computers. What I tried to show was that, although they use many of the tools of the new technology in their work, these guys aren't all that happy.

Finally, there is more to life than TRS-80 hardware and software. Sure, we give you more information on these subjects than anyone else and are proud of how we do it, but you can always count on getting more from us than program listings and schematics. We also try to provide perspective and whether it's robotics, AI, business or education, 80 Microcomputing *will always make an attempt to provoke thought as*

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

Columnar Headings

Automatic Page Numbering

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™

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

PRICES IN EFFECT THRU **February 28, 1982**
Prices, Specifications, and Offerings subject to change without notice.
8202

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.

well as program listings. What differentiates the user from the machine is the ability to think. Let's not lose it.—C.B.

Forum-80

Fred Blechman's article, "Adventures in Modemland," (*80 Microcomputing*, October 1981) was very interesting and informative. However, since considerable time has elapsed since Mr. Blechman gathered the data for the article, there were some inaccuracies in the information regarding the Forum-80.

The Forum-80 system is now available as a complete, fully documented, ready to run package for $350 (not $150 as stated in the article).

The Forum-80 Users Guide offer was withdrawn in June 1981. However, due to the immediate response to the mention of the Users Guides in the article, I have made arrangements to supply them for a limited time. The postage rates indicated were at the old rates. Using the current rates, interested persons should send a self addressed business size envelope with three ounces of first class postage affixed, to Forum-80 Headquarters.

Bill Abney
Forum-80 Headquarters
7600 East 48th Terrace
Kansas City, MO 64129

Accel 2 Update

My review of Accel 2 (*80 Reviews*, November 1981), although accurate when written, was of an early version which has since been updated. The new release of Accel 2 has fixed the problem with Data and Read Statements. It now correctly compiles programs that use these commands. Compile times are also now somewhat less. This version has been out since February.

Bruce Douglass
Vermillion, SD

More on SuperSoft's Lisp

I want to thank you for publishing Gary McGath's review of SuperSoft's Lisp for TRS-80, Models I and III, in your October issue.

In this review, he raised a couple of points which the author, Ron Nicol, has responded to. They are as follows:

Mr. McGath's implication that he has found two bugs in the interpreter is false. The fact that an atom must be followed by a special character is not an error. As noted on page 2 of the documentation: "An atom is ended when a special character is encountered. Normally a space is used to signify the end of an atom." The other Lisp special characters are listed in the documentation. Mr. McGath points out that if an atom is given the property NIL it can only be removed by modifying the property list. NIL in Lisp signifies the empty list. NIL should never logically be placed on a property list since it indicates the absence of a property. If a property is to be removed the Lisp function REMPROP should be used.

It is true that the cassette version of the interpreter does not have printer commands. The design criteria for the cassette version was to produce the most powerful interpreter which could be used effectively in a 16K machine. It was decided that the printer functions could be deleted. The disk version, however, contains very powerful printer functions which, in fact, are much more useful than their Basic counterparts—Ron Nicol

Once again, keep up the good work.

Herb Schildt, President
SuperSoft Associates
Champaign, IL

Letter-Writer Update

Thank you for your excellent article on word processing ("A Fast Round Up" *80 Microcomputing*, November 1981) by Gordon McComb. We have added several more features to the current edition of our Letter-Writer since Gordon wrote his article. These include: tabs; page centering (whole page and part of a page); option to justify; calculating option (adds and subtracts columns of numbers for doing budgets, taxes, etc.); line numbering option (numbered printed lines for legal papers and proofreading); and direct access disk files (retrieves/writes any block of text. If only one line in a street address on a file of labels needs to be changed, only that one line needs to be accessed. You do not have to read/write a whole file).

The Letter-Writer by Astro-Star Enterprises (5905 Stone Hill Drive, Rocklin, CA 95677) is now priced at $23.99 for tape and $37.99 for disk (Model I or III)—prices include shipping and tax charges. As noted in Mr. McComb's article, customers should try to use various systems to find the one that is best for them. We have made testing our system easy. Customers can try our Letter-Writer for three months—where it counts, in their office, home or classroom. If not satisfied, return it and we will refund all but $3.50 to cover our handling and shipping costs.

Craig Wood
Astro-Star Enterprises
Rocklin, CA

Softswap

Several months ago we sent an announcement about SOFTSWAP, our exchange of teacher-written public domain educational software, to *80 Microcomputing*. Included in the announcement was the following sentence: "New disks are listed in the CUE NEWSLETTER, or send $1 for ordering/exchange information to: Ann Lathrop, Library Coordinator, SOFTSWAP, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063.

Unfortunately, *80 Microcomputing* omitted this important information and, instead, listed the announcement as a free item on their Reader Service card. This has caused us a great deal of difficulty, as we run the SOFTSWAP as a largely volunteer activity and do not have personnel or funds to print, address or mail free catalogs and brochures. We have been flooded with hundreds of requests for a "free" item that was never intended to be free. We have tried to send postcards giving the correct information to as many people as possible. For those of you who did not receive any response, please accept our apologies. It is our intention to make our software as freely available as possible, and at the lowest possible cost (it is absolutely free to anyone who visits our Center and copies the programs onto their own disk or tape). We regret any delays or irritations that may have been caused by this breakdown in communication.

Ann Lathrop
Library Coordinator
San Mateo County Office of Education
Redwood City, CA

More on the E-Z Loader

I want to respond to a statement made by R. B. Shreve in his article "The Wave Shaper" (*80 Microcomputing*, September

1981). Mr. Shreve states that the E-Z loader, a cassette pulse re-shaper for the TRS-80 designed by myself and Dave Miller, is more complicated than it has to be. I would like to take this opportunity to restate the design features of the E-Z loader and let your readers decide for themselves.

Unlike most pulse re-shapers, the E-Z loader attacks the cassette interface problem from three angles simultaneously. The incoming signal is first transformer-coupled and stepped up to 10–20 volts. At this point the pulses are *peak detected*, and used to trigger ½ of a 74LS123. This peak detection is the first important feature of the E-Z loader. If simple threshold detection were employed, as the recorder's volume increased from zero a point would be reached where the unit would trigger correctly. As the volume continued to increase, a second point would be reached where not only valid pulses would be capable of triggering the input, but also any secondary pulses and noise that happened to be on the tape would trigger the input and "blow" the load. The peak detection scheme used in the E-Z loader, although simple, effectively sets the trigger threshold a fixed amount below the pulse peak and maintains that relationship regardless of the actual pulse amplitude, even if it is changing rapidly.

The second section of the E-Z loader is similar to Mr. Shreve's device and most other pulse re-shapers, creating an output pulse of fixed duration and amplitude for each input trigger. However, the E-Z loader goes a step further. It is fairly common to see multiple pulses recovered from the tape for each single pulse recorded. This sometimes is more than a simple device can handle, since to the input, the "extra" pulse looks like any other pulse. The E-Z loader uses the other half of the 74LS123 to disable the pulse shaper input for approximately one-half the period between successive pulses, eliminating the possibility of a false trigger from a double pulse, noise, and so on.

A further problem with input schemes such as that used on Mr. Shreve's device is it recognizes only the positive-going component of the signal. Although the output from the TRS-80 is a symmetrical square wave centered on zero volts, my CTR-41 normally produced negative-going pulses, and another recorder I used produced positive pulses. Although an input that recognizes a rising voltage as a trigger will work with negative pulses due to the capacitive coupling used, it will not always be triggering reliably due to the inversion. The TRS-80 and the E-Z loader both recognize either polarity, and as explained above, the E-Z loader looks for the peak of positive or negative pulses.

We feel the E-Z loader goes several steps further than other pulse re-shapers and comes as close as possible to eliminating cassette load problems. The E-Z loader was designed taking more than one potential problem into account, and its slightly more complex circuitry is the result. Since a circuit board is available and parts count is actually not much higher than Mr. Shreve's device, I feel the slightly more complex E-Z loader presents no problem.

"Is the E-Z loader, a cassette pulse re-shaper for the TRS-80, more complicated than it has to be?"

For more information, see the September 1979 issue of *73 Magazine* or send a SASE to : Paul Goelz, 2228 Madison Pl., Evanston, IL 06202.

Paul Goelz
Evanston, IL

RSBasic

A few weeks ago I purchased the new Radio Shack Compiler Basic written by Ryan McFarland. I had been waiting patiently for over a year since Tandy first advertised the compiler. With a wait like that, I should have known there were problems.

I should not have wasted my money. The compiler is slow. The internal editor (a part of RSBasic) is slow and very limited. The loader for the text file is extremely slow.

The external editor (BEDIT/CMD) provided with the compiler package has several bugs. I don't understand why, since it seems to be based on the old Microsoft Edit-80 supplied with Radio Shack's Fortran and Macro Assembler. Although that package also had a few bugs, they were not the same ones I encountered with BEdit.

The data files created by the Basic Compiler are *not* compatible with any data files created by Radio Shack's Level II Disk Basic or Fortran. The result of this incompatibility is that any existing data base is worthless if I want to use compiled Basic. I cannot see any easy way to convert files from one format to the other, either, and no conversion program was supplied. I suspect a similar situation exists with Radio Shack's Cobol package, also authored by Ryan McFarland.

I wrote to Radio Shack to complain, and they responded promptly. To their credit, they authorized my return of the software. To their discredit, I don't think they fully understood the nature of some of my complaints. I also do not think they realize the major problem caused by file structure incompatibility. If they do recognize it, they clearly do not care.

Paul Jaeger
New York, NY

Complaints About RSBasic

When I purchased Compiler Basic I expected easy integration of machine language and possibly Fortran programs with Basic; speed computation coupled with ease of program creation; and the ability to create programs that would operate in a stand-alone environment.

Ten General Complaints

First, the Compiler, RSBasic, needs a run-time system (Radio Shack's, not mine) for the compiled program. It does not create a stand-alone (/CMD) program.

Second, Fortran and the Assembler create relocatable files then linked into the finished program. RSBasic-compiled programs cannot easily be integrated with machine language code. In this respect it is as awkward as Level II Basic.

Third, RSBasic file structures are *totally* different from your Disk Basic and Fortran files, forcing a monstrous conversion process if I want to use existing data files.

Forth, different file structure also creates the inability to use both Disk Basic/Fortran and RSBasic programs on the same data base. Programming flexibility is thereby non-existent. Frankly, I was dumbfounded and totally outraged when I discovered the incompatibilities.

Fifth, unlike the Level II editor, the internal editor for RSBasic is worthless for anything more than very minor changes. Editing source text must usually be done by BEdit. That entails saving the source file on disk, leaving RSBasic and going to DOS, running BEdit, loading the text file, editing the text, saving the source file, going to DOS, reloading RSBasic, and re-

loading the source file. If you are out of breath reading that sentence you can imagine my weariness at creating and debugging an RSBasic program.

Sixth, RSBasic is very, very slow in loading a source file from disk. This adds to the aggravation of item five above. Perhaps the program must do a lot more than just get the text into RAM, but it appears the loader software is just not well constructed.

Seventh, I can find no way to freeze or terminate a listing of text under RSBasic. This is very aggravating when the program is 400 lines long.

Eighth, the compilation process is very slow, with the disk constantly starting and stopping as the source text is read in and compiled.

Ninth, some of the language structure is strangely different from Level II. For example, Print Using in Level II has a semicolon following the format item, whereas RSBasic requires a comma. Perhaps that difference is necessary, but if it is not, I would only call the difference stupid.

Tenth, there is no way to stop a compilation. If the system was fast (using the Fortran compiler as a comparison) I would not complain. But since the compilation process is so slow, I have time to think, and sometimes I will catch errors or make changes in my head before compilation is done. Then I have to sit and wait for the process to finish even though I don't need it.

I also encountered several bugs causing either keyboard freeze or system reboot. I have been able to document and repeat the following three problems using TRSDOS 2.3B:

First Bug

First, execute BEdit and insert some text. Second, exit the insert mode and give the [incorrect] command L (for List, since we just got through using the RSBasic editor). Third, to the query about appending the new file, forget that we are supposed to hit Break (still thinking we are in the RSBasic editor) and respond N. Now, as the system attempts to load a non-existing file, we have lost the text in the buffer, but the worst is yet to come. Fourth, attempt to insert some text. The editor will respond 'No room between lines'. Fifth, insert at the *end* of the text (command I*). Wonderful. We can put in that line. It is numbered one higher than the last (non-existing) line that we lost after the L. Sixth, hit Enter to terminate that line and enjoy a dead keyboard!

Second Bug

First, enter RSBasic and create a new source file using AUTO. Second, save the file on disk: SAVE filename. Third, while still in the compiler, do some editing. Fourth, save the file on disk again: SAVE. Fifth, when the keyboard freezes up we remember that the compiler only remembers the name of the working file if it was previously loaded under an old command.

Third Bug

First, create a program with BEdit, making liberal use of the Tab(shift→) character. Second, give the command to Compile filename, filename. Third, watch the compiler fail as it encounters the unknown ASCII character 09 hex.

Paul Jaeger
New York, NY

"There is no way to stop a compilation."

Radio Shack Responds

Let me address your specific complaints one for one:

First, I know of no compiler Basic which generates true "native code." All require a run-time—Microsoft accomplishes this by appending the run-time system to the linked modules (which adds 8–12K to each /CMD file you create).

Second, RSBasic was never intended to produce or use relocatable code.

Third, RSBasic has always been advertised as different from our current interpretive Basic and intended for program development, not conversion of existing applications.

Fourth, TRSDOS had to be changed to correctly support variable length records. Compiler Basic was never specified or intended to be used with interpretive Basic and Fortran.

Fifth, this process is essentially the same as developing a program with any other compiler. RSBasic provides the ability to develop a program interactively prior to compiling which others do not.

Sixth, RSBasic does perform many operations when loading source code but it is very well constructed.

Seventh, to freeze the listing of a program under RSBasic, simply use shift @ as in Level II or Disk Basic.

Eighth, RSBasic is designed to compile in a disk-to-disk environment. This provides you the ability to write source code much larger than would fit into available memory if needed.

Ninth, different programs use different syntax.

Tenth, this is a good suggestion and I will be sure it is passed to systems software.

I have also passed your bug list to our software development group and asked them to verify them and take the appropriate action. If you cannot live with the "problems" in the Basic Compiler, please consider this authorization to get a refund from your local store.

Bill Wash, Director
Radio Shack Computer Customer Service

Please see this month's review pages for a review of RSBasic.—Eds.

The following is an open letter to equipment manufacturers and software publishers—Eds.

Educational Materials Wanted

We are a group of inmates confined at Folsom State Prison who are seriously interested in becoming computer programmers and system analysts during our period of confinement. We are attempting to locate software applications and programming tools for use with our TRS-80 Model I (48K, two disk system), helping us to learn programming techniques and program development, as well as increase our knowledge of microcomputers.

There are about 200 inmates here at the prison interested in programming and eventually we hope the Education Department will be able to offer comprehensive programming courses. There are no funds available at present to obtain equipment and materials nor does it appear there will be any in the immediate future. In response to a number of letters sent to equipment manufacturers and a letter sent to the editors of a popular computing magazine, we have received several pieces of equipment and a number of computer textbooks and some software materials. However, we are still far short of instructional materials for use by the large number of men interested. We have a real need for instructional application software materials and programs.

An awful lot of real talent is locked up here at Folsom and it is my firm belief that much of this talent could be utilized in a constructive way if given a proper direction. While not everyone may have the

Continued on page 26

DATA RESOURCES

QUALITY SOFTWARE FOR YOUR TRS-80® DISK SYSTEM

TO ORDER CALL TOLL FREE 800-525-7958

Silver Edition Software Series

Data Resources is pleased to continue its commitment to professional quality TRS-80 software by introducing The Silver Edition Software Series. This collection of programs represents the best contributions from the most talented and popular authors available.

DATA RESOURCES NEW SUBSYSTEMS FOR AIDS III

AIDS DISK SORT

Designed for files that are too large to sort "in memory" with AIDS or for when you don't want to wait while AIDS loads the records. Records are physically rearranged on disk.

MODEL I **$24.95**

AIDS FILE EDITOR

Allows you to update and edit records in AIDS data files without loading the file into memory. Now you can change a single record in a mailing list quickly and easily with random access selection.

MODEL I **$24.95**

AIDS DATA ENTRY MODULE

Designed for inputting data directly into AIDS files. Fields may be fixed to repeat the same data. The key-in routines are improved in speed — no more over keying. Data can be added to the end of an existing file or a new file can be created. Files are automatically opened and closed so even if the computer is turned off in the middle of work, no more than 255 keystrokes can be lost.

MODEL I $24.95

AIDS SYSTEMS BY METS TECHNOLIGIES

MTC AIDS III

This easy to use system allows even a novice TRS-80 user to create data files custom configured for many applications, from lists for mailing and cataloging to financial reporting and time scheduling. It requires **NO PROGRAMMING** and is complete with features for adding, deleting, sorting updating and printing.

MODEL I **$69.95**

MTC CALCS III

Performs numeric calculations of data contained in AIDS files. Ideal for financial applications.

MODEL I **$24.95**

MTC CALCS IV

The same features as CALCS III with more powerful formulas and the ability to store report formats.

MODEL I **$39.95**

MTC MERGE - III

Combines up to 14 AIDS data files into a single file. Duplicates may optionally be purged, and sorted order of records is maintained.

MODEL I **$19.95**

SPECIAL

AIDS III SUPER SYSTEMS

AIDS III & CALCS III & MERGE III

MODEL I **$99.95**

AIDS III & CALCS IV & MERGE III

MODEL I **$109.95**

CUSTOM MADE CABLES

$7.00 per Connector
$1.00 per Foot of Cable
We will make Printer and Drive Cables for you to your specifications. 10% discount for 10 or more.

RIBBON CABLES

Model I or III Printer	$18.95
2 Drive Disk Cable	$19.95
4 Drive Disk Cable	$29.95

B.T. ENTERPRISES

UNITERM/80

By Pete Roberts

This is the state of the art in communications software. It configures itself for either Model I or Model III and can be used with any standard modem, both RS-232 and Bus-Decoding. Especially designed to use the extended commands in NEWDOS/80, but fully compatible with all major DOS systems.

For Model I and III **$89.00**

THE MICRO ACCOUNTING SYSTEM

This is the finest accounting system available for your TRS-80. Ideal for small business, this Intergrated General Ledger, Accounts Receiveable, Payable System includes a Checkbook manager.

We offer something no other general business package can give you-a money back guarantee. If you are not totally satisfied return the package within 60 days and receive a full refund. Write for sample reports and full descriptions.

MAS/80 complete **$489.00**

THE FLOPPY DOCTOR

By Dave Stambaugh

FLOPPY DISK/MEMORY DIAGNOSTIC

Computer "professionals" have long known the importance of regular use of diagnostic software in verifying the integrity of computer hardware. The TRS-80 is no exception; good diagnostics are a must. The MICRO CLINIC offers two programs designed to thoroughly check out the two most trouble prone sections of the TRS-80, the disk system (controller and drives) and the memory arrays.

MODEL I ..... **$24.95** **MODEL III** **$29.95**

NEWDOS/80 Version 2.0 for MODEL I and III $149.00

The hottest Disk Operating System is now available in its latest version. This is the ONE from Apparat, Inc., the people whose systems have made the TRS-80 a reliable computer. This system includes:

- Double Density support on the Model I
- New Superzap with expanded capabilities
- Diskette compatibility between Model I and III

GENERIC

SOFT SECTORED 5¼" DISKETTES

PLASTIC BOX OF 10 ...... $22.95
BULK PAC OF 100 .... $200.00
50 DISKS WITH A FLIPSORT $119.95
HEAD CLEANING KITS $24.95

VERBATIM

SOFT SECTORED 5¼" DISKETTES
DOUBLE DENSITY
DATA LIFE $26.95
1 BOX WITH A FLIPSORT $43.95
FLIPSORT
STORAGE CASE FOR 50 DISKS $19.95
10 TO A BOX 40 TRACK

DYSAN

SOFT SECTORED 5¼" DISKETTES
WHEN YOU CAN'T AFFORD ANYTHING BUT THE VERY BEST.
SINGLE DENSITY $36.95
DOUBLE DENSITY $39.95
10 TO A BOX — 40 TRACK

MODEL I DOUBLE DENSITY PACKAGE NEWDOS/80 Version 2.0 and the LNDOUBLER

Everthing you need to convert your TRS-80 Model I to run double density. Complete with software, hardware, and instructions, — installs in minutes with **NO SOLDERING, WIRING OR CUTTING** **$229.95**

LNDOUBLER Alone **$149.95**

✓540

MASTERCARD AND VISA WELCOME
We also accept
- COD • CHECKS
- MONEY ORDERS

ALL SOFTWARE SUPPLIED ON DISKETTE.
ORDERS IN U.S.A. SHIPPED U.P.S. **FREE.**
Colorado residents add 6½% sales tax.

ATTN: DEALERS - We offer attractive pricing and terms to small computer retailers. Write for information on our dealer program and discounts in small quantities.

Data Resources Corporation

Business & Professional Center 8000 East Quincy Ave.
Denver, Colorado 80237 (303) 773-6665

We take the nail-biting out of mail-order shopping.

If the idea of mail-order shopping makes you nervous, you're in for a pleasant surprise.

Nail-biter #1: I need to talk to someone before I buy it.

When you call Alpha Byte you won't talk to an order-taker. Our people are state-of-the-art experts who live and breathe microcomputers. If you're not sure about *exactly* what you need, or you'd like to discuss the pros and cons of a particular piece of equipment, call us. We love it.

Nail-biter #2: It'll take forever to arrive.

Not from Alpha Byte. An order placed today gets shipped tomorrow. If an item is temporarily out of stock, you won't be charged until stock is replenished and your order is shipped.

Nail-biter #3: What if it's still not right once I get it?

No problem. Return it and we'll happily give you a complete refund. And, of course, we'll pay the shipping charges.

Still biting your nails? Here's the clincher, our guarantee:

We guarantee *everything* we sell for thirty days. If *anything* is wrong, just return the item and we'll make it right.

Put us to the test. You won't be disappointed.

PRINTERS

ANADEX DP 9500	1295.00
ANADEX DP 9501	1295.00
C-ITOH 25 CPS PARALLEL	1440.00
C-ITOH 25 CPS SERIAL	1495.00
C-ITOH 45 CPS PARALLEL	1770.00
C-ITOH 40 CPS SERIAL	1870.00
EPSON MX-80	$CALL
EPSON MX-80 F/T	$CALL
EPSON MX-100 GRAPHIC	$CALL
EPSON GRAFTRAX	90.00
IDS-445G PAPER TIGER	779.00
IDS-460G PAPER TIGER	945.00
IDS-560G PAPER TIGER	1195.00
MALIBU 200 DUAL MODE	2695.00
NEC SPINWRITER 3510 S. RO	2195.00
NEC SPINWRITER 3530 P. RO	2195.00
NEC SPINWRITER 7710 S. RO	2645.00
NEC SPINWRITER 7730 P. RO	2645.00
NEC SPINWRITER 3500 SELLUM	2295.00
OKIDATA MICROLINE 80	389.00
OKIDATA MICROLINE 82A	569.00
OKIDATA MICROLINE 83A	799.00
OKIDATA MICROLINE 84	1199.00

16K RAM SPECIAL........13.95

Set of 8 NEC 4116 ns with instructions. Guaranteed one full year.

MODEL I, III SOFTWARE

LAZY WRITER MOD I	169.00
LAZY WRITER MOD III	169.00
MICROPROOF SPELLING CHECK	84.95
PROSOFT NEWSCRIPT MOD I, III	99.00
PROSOFT MAILING LABELS MOD I, III	29.95
PROSOFT NEWSCRIPT/LABELS MOD I, III	115.00
SPECIAL DELIVERY MOD I, III	119.00
X-TRA SPECIAL DELIVERY MOD I, III	179.00
TRACKCESS MOD I	24.95
OMNITERM SMART TERMINAL MOD I, III	89.95
MAXI-MANAGER	89.95
DOS PLUS 3.3	89.00
LDOS	149.00
CALCSTAR MOD I, III	129.00

DISKETTES

ALPHA DISKS....................21.95

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

VERBATIM

MD 525-01, 10, 16	26.50
MD 550-01, 10, 16	44.50
MD 557-01, 10, 16	45.60
FD 32 or 34-9000	36.00
FD 32 or 34-8000	45.60
FD 34-4001	48.60

DYSAN

5¼, SS/DD, BOX OF 5	24.50

DISKETTE STORAGE

5¼'' PLASTIC LIBRARY CASE	2.50
8'' PLASTIC LIBRARY CASE	3.50
PLASTIC STORAGE BINDER WITH INSERTS	9.95
PROTECTOR 5¼''	21.95
PROTECTOR 8''	24.95

TRS-80 MOD I HARDWARE

PERCOM DATA SEPARATOR	27.00
PERCOM DOUBLER II	159.00
TANDON 80 TRACK DISK DRIVE	429.00
TANDON 40 TRACK DISK DRIVE	289.00
LNW DOUBLER WITH DOSPLUS 3.3D	159.00

ISOLATORS

ISO-1 3-SOCKET	53.95
ISO-2 6-SOCKET	53.95

MODEMS

NOVATION CAT ACOUSTIC	145.00
NOVATION D-CAT DIRECT CONNECT	165.00
NOVATION AUTO-CAT AUTO ANSWER	219.00
UDS 103 LP DIRECT CONNECT	175.00
UDS 103 JLP AUTO ANSWER	209.00
D.C. HAYES STACK Smart Modem	249.00
LEXICON LX-11	109.00

CP/M® SOFTWARE FOR MOD II

MICROSOFT

MICROSOFT BASIC-80	299.00
MICROSOFT BASIC COMPILER	319.00
MICROSOFT FORTRAN-80	369.00
COBOL 80	599.00

MICRO PRO

WORDSTAR	310.00
MAILMERGE (REQUIRES WORDSTAR)	110.00
SPELLSTAR (REQUIRES WORDSTAR)	199.00
SUPERSORT	195.00
DATASTAR	245.00
CALCSTAR	239.00

PEACHTREE

GENERAL LEDGER	595.00
ACCT. RECEIVABLE	595.00
ACCT. PAYABLE	595.00
PAYROLL	595.00
INVENTORY	595.00
PROPERTY MGMT	799.00
CPA CLIENT WRITE-UP	799.00
MAGIC WAND	299.00

OTHER CP/M® SOFTWARE

SPELLGUARD	239.00
SUPERCALC	229.00
COMMX TERMINAL SOFTWARE	75.00
dBASE II	599.00
P & T CP/M® MOD II TRS-80	175.00

SUPPLIES

AVERY TABULABLES

1,000 3½ x 15/16 .... 8.49
3,000 3½ x 15/16 .... 14.95
5,000 3½ x 15/16 .... 19.95

FAN FOLD PAPER (Prices F.O.B. S.P.)

9½ x 11 18lb WHITE 3,000 ct 29.00
14 7/8 x 11 18lb WHITE 3,000 ct 39.00

CORVUS

TRS-80 MOD I, II, III

Controller, Case/P.S., Operating System, A&T
5 MEGABYTES 3245.00
10 MEGABYTES 4645.00
20 MEGABYTES 5545.00
MIRROR BACK-UP 725.00

SPECIALS

SPECIAL NO. 1

TRS-80 DISK AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 44.50

SPECIAL NO. 3

NEWDOS/80 2.0, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 149.00

SPECIAL NO. 4

MICROSOFT BASIC COMPILER, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 179.00

SPECIAL NO. 5

MICROSOFT BASIC DECODED AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 49.95

DOUBLE DENSITY SPECIAL

PERCOM DOUBLER II, NEWDOS/80, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 279.00

COMMUNICATION SPECIAL MOD I & III

NOVATION MODEM, OMNITERM TERMINAL SOFTWARE.
ACOUSTIC SPECIAL (CAT) 225.00
DIRECT CONNECT SPECIAL (D-CAT) 245.00
SERIAL CABLE 25.00

DISK LIBRARY SPECIAL

RACET DISK-CAT CATALOGING PROGRAM, FLIP SORT (50 DISK CAPACITY), 50 STICK-ON DISK LABELS 59.95

TWO NEW 'OTHER MYSTERIES' BOOKS

THE CUSTOM TRS-80 .... 24.95
MICROSOFT BASIC FASTER AND BETTER 24.95

TRS-80 GAMES

TEMPLE OF APSHAI	34.95
HELLFIRE WARRIOR	34.95
STAR WARRIOR	34.95
RESCUE AT RIGEL	24.95
CRUSH, CRUMBLE & CHOMP	24.95
INVADERS FROM SPACE	17.95
PINBALL	17.95
STAR TREK 3.5	17.95
MISSILE ATTACK	18.95
STAR FIGHTER	24.95
Z-CHESS III	24.95
ADVENTURE NO. 1, 2, & 3	34.95
ADVENTURE NO. 4, 5, & 6	34.95
ADVENTURE NO. 7, 8, & 9	34.95
DUEL-N-DROIDS	17.95
STARFLEET ORION	21.95
INVASION ORION	21.95
OLYMPIC DECATHLON	24.95
MONTY PLAYS MONOPOLY	31.95
SARGON II	31.95
BLACKJACK MASTER	27.95
ROBOT ATTACK	17.95
GALAXY INVASION	17.95
SUPER NOVA	17.95
TUESDAY MORNING QUARTERBACK	26.95
LUNAR LANDER	18.95
THE MEAN CHECKER MACHINE	21.95
GAMMON CHALLENGER	18.95
SPACE ROCKS	18.95
PIGSKIN	17.95
ZOSSED IN SPACE	18.95
ARCADE-80	21.95
COMBAT	18.95
SPACE INTRUDERS	17.95
SILVER FLASH	17.95
MORTON'S FORK	26.95
PROJECT OMEGA	21.95

CP/M is a registered trademark of Digital Research.

Alpha Byte is proud to feature the finest business software available for the TRS-80 from Occupational Computing Co., Inc.

OCC A/R WITH BILLING AND INVENTORY CONTROL

One step posting prints an invoice, reduces inventory, posts to customer ledger and accumulates sales analysis. Can be set up as Open Item or Balance Forward system.

Model I,III 750.00
Model II 1495.00

OCC PAYROLL

Includes: Multiple Pay Frequencies, Salary or Hourly, Automatic Deductions (Federal, State, FICA, Disability Insurance), Commissions, Bonuses, Tips, Complete Journal, Check Writer, Check Register, Quarterly 941A and Yearly W-2 Reports.

Model I,III 350.00
Model II 495.00

OCC UNION REPORTING

Features: Interfaces with OCC Payroll (Data entry is from employee time card), Flexible Local Set-up (Union Codes and Classifications), Computation and reporting of Various Funds (Vacations, Dues, Pensions, Health & Welfare, Industry, Training, Promotion, Contract Administration).

Model II 495.00

OCC ORDER ENTRY

Features: Allows On-Line Entry and Maintenance of Customer Orders. Supports Credit Memos, Standard & Non-Standard Discounts and Total Back-Order Control. Automatically Interfaces to OCC A/R, Billing and Inventory Control.

Model II 995.00

OCC MANUFACTURING INVENTORY CONTROL

Features: Bill of Material 'Explosion' maintains inventories in up to 10 levels of sub-assemblies and/or components. Allocates and relieves raw materials based on input of production schedules and completions thru the bill of Materials Explosion Update and Prints Bill of Materials Cost Report. All or selected ranges of assemblies are 'Exploded' into the lowest level components; quantities and unit prices are extended for each component to give a unit cost for the entire assembly. Prints Inventory Exception Report and Suggested Purchase Orders. Automatically interfaces to OCC A/R , Billing and Inventory Control and OCC Order Entry Systems.

Model II 995.00

All OCC Software is supplied with user oriented manuals that include easy user installation instructions. OCC Software is written in Radio Shack Basic and runs under TRSDOS. OCC Software uses Keyed Sequential Access Method (KSAM) File structure, Binary searches and fast Machine Language Sorts to insure the fastest possible execution. OCC's Exceptional Screen Control Format makes data entry as easy as filling out a form. Related OCC Software packages interface to form complete MENU DRIVEN Accounting systems.

If you have a serious business application then you should consider OCC Software. Call for a complete description of any of OCC's Business Packages.

Alpha Byte COMPUTER PRODUCTS

We built a reputation on our prices and your satisfaction.

We guarantee everything for 30 days. If anything is wrong, return the item and we'll make it right. And, of course, we'll pay the shipping charges.

We accept Visa and Master Card on all orders; COD orders, up to $300.00.

Add $2.00 for standard UPS shipping and handling on orders under 50 lbs. delivered in continental U.S. Call for shipping charges over 50 lbs. Foreign, FPO and APO orders, add 15% for shipping. Californiana add 6% sales tax.

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

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

31245 LA BAYA DRIVE, WESTLAKE VILLAGE, CALIFORNIA 91362

80 AID

Merge Malady

I was all excited over the article "Merge for Level II" by William J. Dalesandry, Jr. (*80 Microcomputing*, November 1981) until I tried to run Program Listing 2 on my 16K Model III.

On entering Merge (to append the second program), the cassette does not turn off until tape ends, then the computer resets itself. Hitting Break to stop the cassette prints only the first line of the second program, then garbage.

A friend had identical results with his independently prepared program on his Model I, Level II.

Would it be possible to obtain a correction?

Monroe Mayer
2104 Dalloz Road
Columbia, SC 29204

TWOHAF Aid

In the July 1981 issue of *80 Microcomputing*, you printed a program written by Gil Spencer called TWOHAF. Since I do not own an editor/assembler, I tried POKEing the object code into memory. That worked, but how do I activate the program? The USR and SYSTEM commands did not work. Please tell me how to get this program running. Are there any revisions to the program?

Anthony Wenzel
14537 Perrywood Drive
Burtonsville, MD 20866

Access Time

I have just upgraded my Model III from a 16K tape to 48K with two disk drives. I saved a considerable amount of money by answering ads in *80 Microcomputing*, and by buying non-Radio Shack disk controller board and drives. Everything works except TRSDOS. The problem is, my drives are Shugarts model 400L. They have a track-to-track access time of 20 milliseconds. TRSDOS is written to have a six millisecond access time. Can anyone help me with patching information? Calling the customer service in Texas only got me, "We didn't write the program, so we don't know how to change it."

Dick McKenna
10231 Greenbrook Court
Indianapolis, IN 46229

Date?

I am looking for information on a patch to Model III TRSDOS 1.3 that will disable the Date and Time prompts on power-up. The AUTO function engages only after these two are taken care of. I use my TRS-80 for educational purposes. Why have a student answer these questions of no significance to the task at hand? It would make more sense to turn the computer on and have the desired software executed immediately. Radio Shack claims that they can offer no help.

Tony Nardo
845 Nancy Street
Niles, OH 44446

Time Delay Relays

We are planning on using a TRS-80 to do some in-house testing of eight time delay relays at one time. We wish to determine the length of time each particular relay takes from activation to close. I have a definite need for this on-line equipment as a means of testing products we manufacture.

Gerald H. Servos, President
Instrument and Control Systems Inc.
520 Interstate Road
Addison, IL 60101

NEWDOSPLUS Graphics

I have a Microline-80 printer which can print TRS-80 graphics, and NEWDOSPLUS converts all graphics symbols to periods. Does anyone have a fix that will permit the screen-print to send the graphics codes to the printer unaltered?

Jerry M. Lentz
P.O. Box 5153
Falmouth, VA 22403

ROM RND Fix

This letter is in response to Tom Jones' random number problem ("ROM RND Hangup," *80 Aid*, November 1981). The secret to random number generating is to adjust the seed with each number generated. While the routine shown in Program Listing 1 may not be the most elegant, it does the job in 29 bytes.

By the way, if you want the RND(0) function, simply delete lines 210-270, and line 300. In either case, the result is left in 4121H-4124H. Using the full routine, the integer result is also in HL.

The ROM entry points used are from

Continued on page 26

```
00200 RND  CALL  14F0H      ;RND)0)
00210      CALL  9A4H       ;PUSH WRA1 TO STACK
00220      LD    HL,100     ;RND(100)
00230      LD    (4121H),HL ;PUT LIMIT INTO WRA1
00240      CALL  OACCH      ;CONVERT TO SINGLE PRECISION
00250      POP   BC         ;GET OPERATOR FROM STACK
00260      POP   DE
00270      CALL  847H       ;MULTIPLY
00280      LD    HL,(4121H) ;ADJUST RANDOM NUMBER SEED
00290      LD    (40ABH),HL
00300      CALL  0ABAH      ;CONVERT TO INTEGER
```

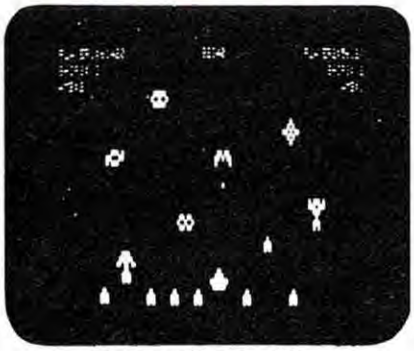
Program Listing 1.

WHY IS THE ALPHA JOYSTICK SUCH A SUCCESS ?

Because of games like these.

THE LATEST BLAST FROM BIG FIVE...

NEW! **DEFENSE COMMAND**



Actual unretouched photos

ALL GAMES:
16K Level 2, Mod 1 + Mod 3 Cassette: $15.95
32k Level 2, Mod 1 + Mod 3 Diskette: $19.95
10% discount for 2 games, 15% for 3 or more
Games may be played with or without joystick.

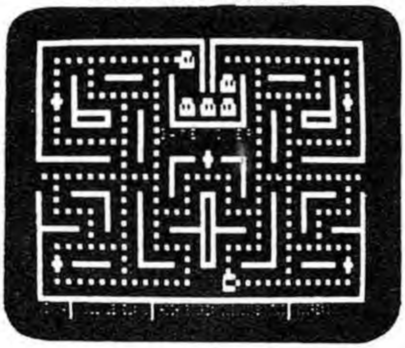
DEFENSE COMMAND

NEW!

Big Five has done it again! Now the most popular arcade game of all time has a fascinating new twist. The Invaders are back! You are alone, valiantly defending the all important nuclear fuel cannister stockpile from a convoy of thieving aliens who repeatedly break off and attack in precision formations. An alien passes your guard, swiftly snatching up a cannister and flying straight off. Quick! you have one last chance to blast him out of the sky. Great action and sound!

BEST SELLER

SCARFMAN



SCARFMAN

BEST SELLER

THE LATEST ARCADE CRAZE now runs on your TRS-80.

It's eat or be eaten. You control Scarfman around the maze, gobbing up everything in your path. You attempt to eat it all before the monsters devour you. Difficulty increases as game progresses. Excellent high speed machine language action game. From The Cornsoft Group. With sound.

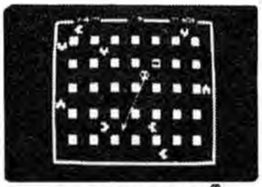
CAUTION: Played with the Alpha Joystick, Scarfman may become addictive.

SUPER NOVA®

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

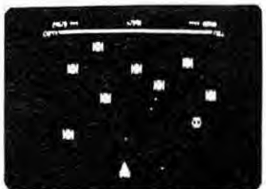
LUNAR LANDER

As a vast panorama moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored — if you can land safely. You control LEM main engines and side thrusters. Absolutely the best use of TRS-80 graphics we have ever seen! From Adventure International. With sound.



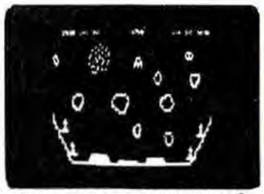
ATTACK FORCE®

As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the "Flagship"... With sound effects!



COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them, too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner... With sound!



METEOR MISSION II®

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

THE ALPHA JOYSTICK: REAL ARCADE ACTION

ONLY $39.95 (MODEL I)

- Features the famous ATARI JOYSTICK
- Works with any Level II or disk system
- Plugs directly into KB or E/I (next to printer port)
- Compatible with any other TRS-80 accessories
- Saves your keyboard
- Fun to experiment in BASIC. Simply use A = INP(0) to read the joystick
- Fully assembled, ready to plug in and use.

Price includes Atari Joystick + Alpha Interface + instructions + demo program. The Alpha Joystick is backed by an unconditional money back guarantee. If you are not delighted with it, return it within 14 days for a prompt and courteous refund.

(Model I version shown)

NEW!

MODEL III JOYSTICK ONLY $49.95

Toll Free Order Line
800-221-0916

Hours: 9AM–5PM, E.S.T.
For Info Call (212) 296–5916

ADD $2.00 PER ORDER FOR SHIPPING AND HANDLING
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
C.O.D. ADD $3.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
NY RESIDENTS ADD SALES TAX.
OVERSEAS ORDERS ADD 10%

ALPHA Products

85-71 79th Street, Woodhaven, N.Y. 11421 (212) 296-5916

210

80 AID

Continued from page 24

Inside Level II by Mumford Micro Systems. By using these, the number type flag at 40AFH is finessed, so you do not have to monkey with it.

Now, for my problem: Radio Shack's Disk Editor/Assembler package includes the Fortran subroutine library, which the manual says contains a number of useful arithmetic routines. But your $100 investment is not good enough to get the whole story. In addition to the arithmetic routines, there are many double-precision trigonometry and LOG routines, along with some disk I/O routines, which are *not* documented.

If somebody out there has figured out how to access any of them, I would like to swap notes. In the meantime, I will be digging through them for goodies to pass along.

G.E. Waldron
2357 East Mall Drive
Apartment 305
Fort Myers, FL 33901

RND Fix

Mr. Jones' problem with the RND function in machine language (*80 Aid*, November 1981) is one that I had faced and conquered. The program that does what he wanted is shown in Program Listing 2.

The important line in here is CALL 0A7FH, which converts the number in WRA1 (single precision, double precision, or integer) to an integer.

Do not try ROM calls from Microsoft's Z-Bug. While the Editor/Assembler Plus is an excellent program, its monitor (Z-Bug) plays havoc with the NTF (number type flag)! This could be part of Mr. Jones' problem.

Has anyone found a way to get the expression evaluation routine at 2337H to work? Roger Fuller's Supermap offers no advice, and even *Microsoft Basic Decoded* does not help much. I have followed all the advice, yet all I get is "?OV ERROR", "?OM ERROR", "?SN ERROR", and so on. Any ideas?

David Librik
2211 East Reservoir
Peoria, IL 61614

```
LD    HL,50
LD    (4121H),HL   ;Save parameter (50) in WRA1
LD    A,2
LD    (40AFH),A    ;Store "integer" in NTF
CALL  14C9H        ;Do a RND—single precision result
CALL  0A7FH        ;Convert WRA1 to INTEGER
LD    HL,(4121H)
LD    (RVAL),HL    ;Store random number in RVAL
```

Program Listing 2.

Output Driver for IBM Selectric

I am interested in hearing from anyone who has written an output driver routine to allow a Selectric terminal of the IBM 2741 or GTEIS/NOVAR 5-50 variety to be used as an output printer for Scripsit from the serial port.

William C. Bennett
3270 Melendy Drive
San Carlos, CA 94070

Microfile Sort Wanted

Has anyone been able to add a sort routine to Radio Shack's Microfiles program? I need a routine than can sort one or more fields at one time. Profile's pitifully slow sort routine gives the disk drive a real workout.

William Greenwood
149 Piccadilly Place, Apt E
San Bruno, CA 94066

Continued from page 20

need or desire to obtain a BS or BA degree, nearly all of us in here do have the need—if not always the desire—to have gainful employment upon release at the end of our sentence. If we can learn about computers, programming, applications, and system development, enough to attract prospective employers—then one major goal of our group would be reached. It is to this extent that our programming studies are being directed.

Any consideration that your firm may be willing to give in obtaining program instructional materials for use with our TRS-80 would be deeply appreciated. You can be assured that the materials would be constructively utilized.

If the possibility exists that you may be willing to send us copies of your programming materials and should you require additional information before doing so, I would be pleased to answer any questions that I possibly can.

We do realize that software materials are expensive to develop and produce, and that you are in business to turn a profit. But if you can see your way clear to send us any of your programs or publications they would be most welcome and sincerely appreciated.

Gottfried R. von Kronenberger
PO Box B49542
c/o Mr. R. E. Miller,
Supervisor of Education
Folsom State Prison
Represa, CA 95671

Screen Printing with NEWDOS80 V2

In the November *80 Input* column, Herbert S. Dubois presented zaps for screen printing from NEWDOS + and NEWDOS80 VER 1.0. The following zap is for those of us that have NEWDOS80 Ver. 2.0. Execute Superzap, and use the DFS option to load FILESPEC-SYS3/SYS. Enter the find command as follows: F,3E,2E. The flash cursor will stop at byte C1H of DRS 184, FRS 4. Enter "MODC1". Change the "3E2E" bytes to "C620". This stops NEWDOS80 V2.0 from replacing characters above the system AX = PARM with a period. It now adds 20H to put the character up the MX-80 graphics range.

Larry Eckenrode
Berkeley Heights, NJ

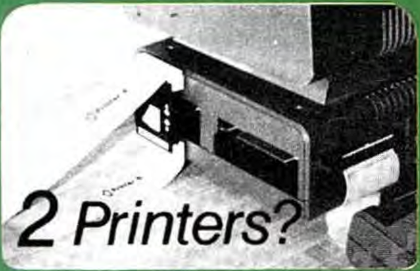
Captran Modifications

The Captran program (Capitals to lower-case, Buzz Gorsky, June 1981) is

Continued on page 28

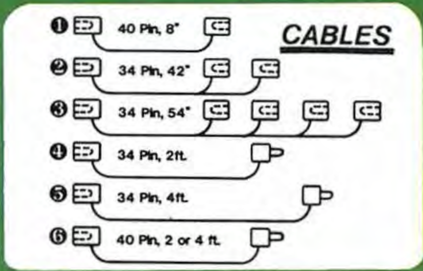
NOW MODEL I AND MODEL III!

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



PRINTSWITCH

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

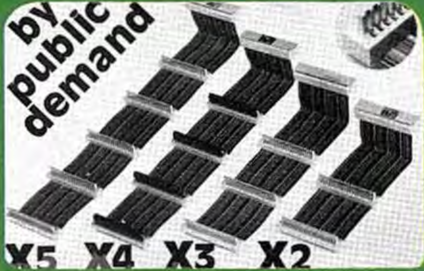


SUPERIOR QUALITY REPLACEMENT & EXTENSION CABLES

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

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

Custom cable configurations are also available. Call us.



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

ANALOG-80: A WORLD OF NEW APPLICATIONS POSSIBLE.

8 DIGITAL MULTIMETERS PLUGGED INTO YOUR TRS-80!!! Measure Temperature, Voltage, Current, Light, Pressure, etc. Very easy to use: for example, let's read input channel #4: 10 OUT 0,4 'Selects input #4 and also starts the conversion 20 A = INP(0) 'Puts the result in variable ''A''. Voila! Specifications: Input range: 0-5V. to 0-500V. Each channel can be set to a different scale.
Resolution: 20mV (on 5V. range). Accuracy: 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.

TIMEDATE 80: REAL-TIME CLOCK/CALENDAR MODULE

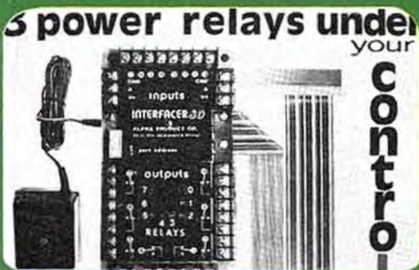
Keeps quartz accurate time for 3 years on 2 replaceable AAA batteries (not included). Gives MO/DATE/YR, DAY of WEEK, HR:MIN:SEC and AM/PM. Features INTELLIGENT CALENDAR and even provides for Leap Year. This compact module simply plugs into rear of Keyboard or side of Expansion Interface (may be slipped inside E/I). Includes cassette software for setting clock and patching to any DOS (including NEWDOS 80, 2.0). Optional ''Y'' connector allows for further expansion. For Model I. Fully assembled and tested. Complete with instructions and cassette. ONLY $95.00. ''Y'' option add $12.00.

DISK DRIVE EXTENDER CABLE, FREE YOUR MINI-DRIVES.

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

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

Get one for each drive . . . . ONLY $8.95



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

GREEN SCREEN WARNING

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

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

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

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

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

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

ALPHA Products ~210

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

80 DEBUg

Basic Disassembler Fix

While running a dump of my Model III ROMs using Basic Disassembler from your July 1981 issue, I encountered two small problems

First, I discovered that two op codes are not supported: 34H-INC (HL), and 35H-DEC (HL). This is easy to fix. Insert in line 9008 after "INC SP": I$(52) = "INC (HL)":I$(53) = "DEC (HL)":. This will give you the proper printout.

The second problem was that while trying to run that long a dump all at once, I got an OS error after about 2,500 bytes. To fix this change the CLEAR1000 in line 1 to read CLEAR3000. In a 16K machine this will not leave any memory for the machine language program, but it will work for a ROM dump. If you have 32K or more, this will work all the time as long as the machine language is in high memory. This also has another advantage in that it speeds up the program slightly, cutting down on the need for the garbage collection process.

Jay Hayes
331 14th Street
Montara, CA 94037

Baudot LPRINT Bugs

There is an error in my article, "Baudot LPRINT" (*80 Microcomputing*, November 1981).

In Program Listing 1 location 7F13 should be 6C instead of 5C. The look-up table was omitted from the listing. The look-up table is locations 7E00–7E7F. If the reader desires to type in the Assembly version, the code can be obtained from Listing 2, lines 1000–1022. Notice that the mnemonics are not accurate, as they were printed with a Baudot printer.

A reminder about the Basic version. The right bracket in line 50 is an up arrow on the keyboard, and remember to protect memory before loading the program.

I will be happy to assist anyone who has a problem using this routine.

Winford Rister
Route 10, Box 33
Lake City, FL 32055

Line 1070 was missing from the program listing "The Pacemaker," November 1981, page 222. It should have appeared as follows:

```
1070 S = 1:TF#(X) = 0
```

UCLBAS Fix

I have the Radio Shack Lowercase Driver Mod installed in my Model I, Level II and use their 26-1104 "Lowercase Driver" (ULCAS). Recently, when entering a fairly large program, the keyboard suddenly produced only one graphic character—no matter what key I hit.

ULCAS can be loaded at any time. It loads in at 7000H and, after the *? / Enter routine, it relocates itself to just below any previous limit set by the memory size response. It also resets the memory size pointer to protect the driver routine from Basic.

In the R/S program, one address is not corrected during the transfer and a large Basic program will overwrite this information. The fix is simple. Enter ULCAS in the normal fashion. Immediately enter and run this short program below.

Nate Salsbury
608 Madam Moore's Lane
New Bern, NC 28560

```
10  Y = PEEK(16561) + 256 * Peek(16562) +
20  H = INT(Y/256)
30  L = Y - 256 * H
40  POKE Y + 64, L
50  POKE Y + 65, H
```

Program Listing

Continued from page 26

designed for Mr. Gorsky's personal word-processing program. It can be used with the Electric Pencil by altering the buffer location from 5500H to 6C02H. My system has the memory size of 65000 so I change the buffer size (BUFSIZ) from A000H to 91F0H, allowing large files to be translated.

To use the program (with TRSDOS 2.3) Pencil is entered and the text to be translated loaded. Then with control "O" return to DOS from pencil. The text remains in memory. Run CAPTRAN/CMD, and the program returns to DOS. Now type Debug, then the Break key. When in Debug type G5C61, which brings up the Electric Pencil with the new (lowercase) text in the normal operating condition.

Kelly Swainson
Sarasota, FL

Superlist Modifications

Here are a few changes to the excellent Superlist program published in the November issue.

To allow a different ORG use:

```
120             ORG     wherever
130             LD      (STACK), SP
140             LD      SP, STACK
. . . . .
3610    BUFF    DEFS    164H
3620    STACK   EQU     $
3630            END
```

To use Superlist with Disk Basic use:

```
300             LD      IY, (40A4H)     ;40A4H Points
                                        ;to beginning
                                        ;of BASIC
2630            LD      DE, (40A4H)
```

To solve the problem created by linefeeds (the down-arrow key = 0AH) in a Basic line use:

```
632             CP      0AH             ;check for crlf
634             JR      A, A2           ;if so, ignore it
```

T. J. Reibold
Superior, WI

"Symbolic Dump Printout"

Mr. Beasley's note (*80 Microcomputing*, November 1981) interested me. I had tried a patch for hard copy but could not get it to work. When I saw Mr. Beasley's note I tried it. It works on a 16K version, although the last five lines (7F87–7F8B) are superfluous. However, Mr. Beasley's comments about values for other versions of RSM are just plain wrong. First, for a 32K version the addresses start with B, and for a 48K version they start with F. In addition, some of the values to be inserted are also different for 32K and 48K versions. Finally, the print routine for a 48K version will be in the FF80 range, and cannot be saved. It must be located at a lower address. Table 1 gives the correct values.

Maynard B. Neher
Columbus, OH

Scripsit-NEWDOS80 Zaps

If you have tried running the Radio Shack disk based Scripsit program under NEWDOS80 you may have discovered as I have that the cassette operations no longer function properly. Although the lack of cassette capability is not critical with a disk based system, it is convenient to be able to make a cassette back-up of important documents.

The problem can be noted in either of two ways. If the cassette recorder is connected to the expansion interface port 1, it will not start at all when you issue the tape commands. If the recorder is connected directly to the keyboard jack, the recorder starts and begins loading, but an error message (Tape loading error) soon appears.

Fortunately, all of these problems are easily overcome with a few additional zaps, Apparat's euphemism for patches, applied directly to the disk file. The procedure is as follows. First, make a back-up copy of the original Scripsit program and work from the back-up, rather than the original. Next apply the first four zaps from chapter 13 of the NEWDOS80 manual per the directions supplied by Apparat. *Do not apply the zap five.* Instead, apply the four zaps shown in Table 2.

These patches set the cassette latch in the expansion interface to port 1 on power-up and disable the interrupts only during cassette operations. The interrupts will be enabled at all other times and thus allow use of the interrupt driven features of NEWDOS80, such as JKL, DFG, and so on. These features will not be active during cassette operations, however.

R. Lee Holstege
Millersville, MD

Location			Current Value	New Value		
16K	32K	48K		16K	32K	48K
606E	B06E	F06E	F1	C3	C3	C3
606F	B06F	F06F	77	81	81	F0
6070	B070	F070	F5	6F	BF	EB
6081	B081	EBF0	00[a]	F1	F1	F1
6082	B082	EBF1	00	D3	D3	D3
6083	B083	EBF2	00	F8	F8	F8
6084	B084	EBF3	00	F5	F5	F5
6085	B085	EBF4	00	C3	C3	C3
6086	B086	EBF5	00	71	71	71
6087	B087	EBF6	00	60	B0	F0

(a) Depending on previous programs, there may be non-zero values from here on.

Table 1

```
SCRIPSIT/CMD,00,C3
Change 57 F3 ED to 57 FB ED

SCRIPSIT/CMD,00,03
52 18 3D 43 4F 50 59 52 49 47 48 54 20 31 39 37 39 20
52 3E 00 32 E4 37 C3 3F 52 F3 C3 94 66 F3 C3 14 63 20

SCRIPSIT/CMD,25,59
Change 65 14 63 2A to 65 0C 52 2A

SCRIPSIT/CMD,25,6A
Change 67 94 66 94 66 CB to 67 08 52 08 52 CB
```

Table 2

Goldflam Sort

I created a sort routine (see Program Listing 1) that is easier than using a Shell-Metzner or bubble sort. When sorting (especially with a large number of items) my subroutine beats the S.M. method by a considerable amount of time. The only time my method is slower is when the items to be sorted are already in order, and then it still beats the S.M. method, but loses to the bubble method by one or two seconds.

Michael Goldflam
Massapequa Park, LI, NY

```
1 Input "How much memory would you like to set aside for items"; B: CLEARB;
  Input "How many items to sort"; A: dimA$(A): For X = 1 to A: inputA$(A):

2 For x = 2 to A: For X1 = 1 to X - 1: if A$(X)•A$(X1) Then 4 else Next

3 Next X: Goto 5
4 B$ = A$(X): For X2 = X to X1 + 1 step - 1: A$(X2) = A$(X2 - 1): Next: A$(X1) = B$: Goto 3

5 Cls: ?"Done": Input "Hit ENTER to continue"; B: For X = 1 to A: ?A$(X);" ";: Next
```

Program Listing 1

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

"Not surprisingly, tax preparation software headed the list."

The year 1981 is fast coming to a close. For accountants this is especially significant: It is projection time. Needless to say, projection time is always quite valuable but this year the passage of the Tax Act of 1981 has created unique situations. The mechanics of the law and the effect of taxable income have received wide publicity. Woe to the accountant who fails to advise clients of possible benefits or pitfalls!

With our increased sensitivity to client needs we looked for tax projection software for our Model II. Our search was fruitless: We found only a package for the Apple prepared by our old friends Aardvark Technical Services. When we contacted them, they kindly sent us a well-written manual and indicated that in the future a CP/M compatible version might be available.

We could have used one of the tax preparation programs for projections, but these programs are oriented towards a final return. Accordingly they do not prepare the comparative projections offered by the Aardvark package. For this year, at least, we will use a combination of service bureau packages and the old pen and pencil. Hopefully next season Aardvark and others will provide tools for Model II users.

During our search for Model II tax software packages, we polled our professional staff about what software would be most valuable for their future desktop computers. Not surprisingly, tax preparation software headed the list. It seems that most dream of plugging in their client's data and having their micro pop the return out of a slot, ready to mail.

Also high on their lists was software to help assemble client tax data. Many complained that the hazards of data collection waste many valuable hours. All had "war stories" of completing a complex return only to find a small fact inadvertently omitted. The resulting recalculation and repeat preparation is doubly frustrating because the problem always happens when filing time is short.

From an economical point of view, software which reduces costly last minute preparation to an absolute minimum is worthwhile. The best approach to achieving this goal is to develop software accumulating the taxpayer's information during the year. Finalizing the return is then simple. The resulting data, summarized in machine-readable form, can be input directly into one of the more sophisticated tax preparation packages available from established service bureaus.

Because taxpayers will never get data to the accountant on time, we still have to process an enormous amount of information in a very short time. The best use of microcomputers in the tax preparation process is in the data entry phase. Of all data processing devices currently available, micros are least capable of handling large volumes of printing and processing. The average configuration is usually equipped with slow speed disk storage and character printers. Even a fast character printer has rates under 100 lines per minute.

"If something breaks (or Murphy joins your staff) you will have some reserve capacity."

If your firm is planning to prepare tax returns in-house on your microcomputer this season, plan to double up on your data processing equipment before the end of March. Rent or arrange for use of additional equipment. If something breaks (or Murphy joins your staff) you will have some reserve capacity. In addition, consider preventive maintenance for your present installation. Have your drives serviced now and lay in a good supply of printer ribbons and disks. Hang the cost and select a conservative approach—you will not be sorry.

Data Accumulation

Although current micro configurations are not well suited for volume tax preparation, they are ideally designed for data accumulation. The limited disk capacity so constrictive for general business applications is just right for accumulating data on a single client's financial activity. Unless the client is very unusual, 500K of disk storage is more than adequate. With the low cost of disks, devote a separate disk to each client, and store each client's activity disk in a loose-leaf binder. As you receive data, enter it into the appropriate file on the disk and retain it in the loose-leaf binder.

At some time during the year, run a projection program and compare the results to a summary of the previous year's actual and estimated tax activity. The source of the data for this projection can be the actual activity files and your best educated estimates. Discuss the results of this projection with your client; it can serve as a feedback loop to identify missing data. Then update the activity files and schedule the final touches to be applied at year's end.

A system with decentralized files such as this can be difficult to control. With each client's file maintained just in a notebook, management of a firm's tax practice is impossible. As growing firms acquire a large volume of returns to process, they find a "tax control" facility necessary to ensure that client returns do not "drop through the cracks."

Here's a Project For You!

To control the decentralized file structure, consider linking each desktop micro together in a network. In such a network each local unit communicates with a control program located in the tax controller's computer each time a local client file is opened. The control program indicates processing status and activity of client files to the tax controller. Absence of activity at a local unit is as significant as high volume. The same network transfers data to a larger facility for final processing.

The hardware required to implement

this system is already available. Based on the current Radio Shack users' newsletter, Tandy is planning to offer ARCNET (Datapoint's attached resources network) compatibility to Model II users this year. In addition, the IBM/3270,3780 and ReForMaTTer software permit the Model II to be used as an input device in many IBM mainframe systems. In short, the Model II is as close to being a universal input device as any piece of data processing equipment. Unfortunately the only missing links are the local microcomputer programs required to implement the system design.

If you plan to implement such a project, keep the following elements in mind: You should develop a taxpayer profile to maintain personal data about the taxpayer and his or her dependents; arrange the program so potentially taxable events relating to the taxpayer are highlighted whenever you execute the program. An extension to the personal data program should deal with your client's employment history and salaried earnings. Since most salaried individuals have an earning pattern, you could apply regression analysis techniques to project potential earnings even if you lack specific information.

Another important component of a micro tax data acquisition system is a securities trading analysis package. This program would maintain an inventory of capital assets and summarize transactions likely to appear on schedules B and D. Trading analysis packages are starting to appear for investors. In general these are difficult to apply to accounting needs; they are designed to evaluate portfolio performance and accordingly do not have the accounting detail required.

For other business activities usually listed on Schedule C, you can use current financial software. Even the least complicated general ledger system that I have seen maintains a listing of receipts and disbursements. You can use the resulting income statements as a summary for the return. You will need special software packages to manage tax shelter recordkeeping.These packages will vary depending on the nature of your firm's clientele.

In setting up system design parameters, make a provision for all software to have a VisiCalc interface. In addition, you should be able to link all data files into tax projection and tax preparation programs.

The Estate Planning Model

While tax data accumulation would be the primary mission of a desktop computer, you could use this resource for other client support tasks. For example, the new ERTA tax act provides opportunities for significant family tax savings through intelligent estate tax planning. Fortunately software has already been developed for this area.

The software, the Estate Planning Model (EPM), is available from the Beard Software Development Company, 59 Skyline Avenue, Canfield, OH. The proprieter, Ralph A. Beard, is an attorney. EPM, written in Radio Shack Compiler Basic, is supplied in compiled form.

Compiler Basic differs from the Basic supplied free with the Model II in several respects: It has a different syntax and can be compiled to operate more quickly than the normal Basic. The same software firm that supplied the Cobol Development system developed this version of Basic. Like the Cobol system it uses a run time module to execute the compiled source code.

If you already have Compiler Basic, you own the run time module. In this case, Beard Software Development will sell you the code alone for a license fee of $475. If you do not own Compiler Basic, the run time module is available from either Radio Shack or Beard for only $30.

"...apply regression analysis techniques to project potential earnings..."

The EPM system is supplied with an excellent manual. This manual provides a tutorial on the system use, and frankly discusses the assumptions made during the system design. Potential purchasers should read it. Beard offers a copy without the software, for $50. You can apply this cost to the purchase of the software.

It is apparent that the system's author is not only an experienced estate planner but a data processing professional. The documentation's clarity and the smoothness with which the software operates is a testimony to many hours of careful preparation and thought. The professional presentation of a complex subject (estate planning using a computer) is well expressed in the manual.

The manual is organized into four main chapters, a table of contents and an index. The first two short chapters are devoted to an overview and the mechanics of getting started. The third chapter, a familiarization tutorial, uses a test case as an example of the system's features. The fourth chapter contains technical details such as program size, specifications, limitations and troubleshooting.

System operation is designed to be simple. You insert the disk and satisfy the normal Tandy initialization requests, and the system automatically calls up the Run-Basic module and executes the main menu. The system assumes no data processing expertise, but the user should have experience and estate tax planning. The manual, program menu captions and output reports refer to sections of the code and the estate tax return familiar to those specializing in this area.

Once in the system, you enter client and spouse financial inventory data. If a computer is not available at the time of the client interview, an estate planning worksheet is provided. The system includes a special program to print as many pro forma copies of this worksheet as desired. The worksheet serves as an inventory of the client's assets: Use it to calculate a tentative gross estate.

In addition to a tentative gross estate, the worksheet provides for accumulation of additional data (such as Section 2039 benefits not included in the gross estate, joint property, life insurance and entered marital bequests). Because the planned-for event is, hopefully, some time in the future, a provision allows you to apply a growth rate to estate assets. The growth rate provision is an extremely important factor in establishing the estate plan. Even a modest potential estate can grow to taxable proportions in a very short time with today's high interest rates. Provisions are also made for charitable bequests, gifts after 1976 and the exemption used in the period 9/8 to 12/31/76. You complete the worksheet by entering a presumed date of death for the client and spouse.

After data entry, the program stores all variables on the disk. At this point, you can generate an estate tax projection, however, EPM makes several assumptions. These include an estimate of an expense factor of 5 percent of the gross estate and the use of Ohio estate tax rates in the state tax projection.

Alter these defaults to more nearly approximate the estate planner's local situation. The 5 percent expense figure was intended to apply only to attorney and executor's fees. This percentage was based on Beard's observations for estates in excess of $250,000. You should recognize the effect of debt by offsetting it against the assets scheduled in the inventory. To utilize a different state tax amount EPM offers two other alternatives: a state tax

80 ACCOUNTANT

amount equal to the credit allowed on the federal return or a general state tax assumption based on a formula. Unfortunately New York levies a tax considerably in excess of the federal allowance, consequently we had to alter the default to the general assumption. Fortunately changing the defaults resulted in a change of the disk data. The new values became the defaults and did not have to be reentered.

The resulting projections of estate tax are presented with values calculated using six different assumptions. These assumptions range from a simple will to a calculated optimum marital deduction formula and bypass trust. Because of screen size limitations, all calculations cannot be presented at once on the monitor screen. However using the 132 character width of the printer, you could prepare a complete comparative analysis of all alternatives.

The printing program allows control over such factors as page number, top of form and a selection of a fancy report title page. The printing program also displays our firm's logo on each page. The printing menu allows reentry of variables if you should notice an error or require a different set of assumptions. If desired, you can switch the dates of death of the client and spouse and print a new schedule.

The program and the resulting reports are intended for the professional estate planner. Therefore, if data is inserted without expert knowledge, the resulting reports will be of little use and could prove misleading. Beard indicates that a first-time user should test the system by entering known data and comparing previously prepared projections to the system's output reports. This procedure, always wise when using a new system, is especially valuable in this case since understanding assumptions and system defaults is vital to proper use.

The ERTA tax act of 1981 constitutes a revolutionary change in taxing philosophy concerning the transfer of wealth. By means of memoranda and seminars we alert our saff and clients to the opportunities created by this new legislation. Software such as the EPM system will greatly simplify this task. ■

Disk Storage: It's No Place to Compromise.

A dependable disk system means everything to the high-performance operation of your computer. At Percom, we know this. And we do something about it.

For example, the connectors on our TFD drive systems are gold plated.

Why? Because solder-plated contacts oxidize, forming an insidious insulation that can cause erratic operation. And then total failure.

Gold-plated connectors are just one way we make better drive systems than the competition.

Now #1 for the Model III

Quality Percom TFD drives provide more features, cost less. Add-on drives start at only $399.00. Complete First-Drive Systems start at only $599.00. Features: **"Flippy" Capability** — Record your programs and data on either side of a diskette. **Greater Storage Capacity** — Store 180 Kbytes — formatted — on one side of a 40-track TFD drive. Store over twice as much on a TFD 80-track drive. **Think Megabytes** — Your TFD drive controller will handle up to four drives. You can access almost 1.5 million bytes of on-line program and data files. **Either Add-in or Add-On** — Your TFD First Drive System can be **either** internal (add-in) or external (add-on). In either case, you get a complete system including the disk controller, drive, power supplies and disk-operating software. **BASIC-Language DOS** — Included on diskette with each First Drive System is Percom's OS-80/III™. This easy-to-use disk-operating system loads into 7 Kbytes of RAM and then frees the drive for other use. With OS-80™ programs, you can have full, read-write interchangeability between Model I and Model III diskettes. **Or Model III TRSDOS*** — TFD 40-track drives work with Tandy's Model III TRSDOS, without modification. For 80-track operation, TRSDOS can be easily modified with diskette patches supplied.

Still #1 for the Model I

Besides greater storage capacities, more quality control measures and lower prices, all Percom Model I drives are rated for double-density operation.

Plug a DOUBLER™ in your Expansion Interface and enjoy the same double-density disk storge capacity as Model III owners. Included with each DOUBLER is a TRSDOS compatible double-density disk-operating system. DOUBLEZAP programs are available for upgrading other popular DOSs for DOUBLER operation. And our double-density version of OS-80™ costs just $49.95.

Of course you don't **have** to upgrade your Model I for double-density operation to use Percom disk drives. But it's nice to know you can.

Percom TFD drives for the TRS-80* Model I are available in 40-, 77- and 80-track versions, in 1-, 2- and 3-drive configurations. Prices start at $399.00.

System requirements: Model III TFD drives work with a 16-Kbyte system (min) and Model III BASIC. The initial drive must be a first-drive system. An optional interconnecting cable is available for expanding with external drives #3 and #4. Model I TFD drives work with a 16-Kbyte system (min) equipped with an Expansion Interface, Level II BASIC and DOS software, and an interconnecting cable. Two- and four-drive interconnecting cables are available from Percom.

To order products, or for the name of your authorized Percom retailer, call toll free:
1-800-527-1222

PERCOM

PERCOM DATA COMPANY, INC.
11220 PAGEMILL RD. DALLAS, TX 75243
(214) 340-7081

Yes . . . I'd like to know more about the best for my TRS-80 computer. Send me free literature about quality Percom products.
☐ Model I ☐ Model III

Send to
PERCOM DATA COMPANY, Inc., 8-C
11220 Pagemill Rd. Dallas, TX 75243

name
street
city state
zip phone number

MAIL TODAY!

™trademark of Percom Data Company, Inc.
*trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

Maximize the potential of your TRS-80*

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 cannot realize the full potential of the TRS-80*.

Our response to the clamor for additional information is the *Encyclopedia for the TRS-80**, a ten-volume reference work of 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* Each book is full of material on programming techniques, business, language, hardware, games, tutorials, education, utilities and interfacing.

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, approximately six weeks apart. This means that each new volume will reflect the latest developments and discoveries, making this a living encyclopedia for TRS-80* users.

The first four volumes are being issued during 1981. The remaining volumes will be issued during the first half of 1982. The deluxe COLLECTOR'S EDITION has a handsome green and black hard cover with a dust jacket.
A soft cover edition is also available.

DEALERS Please request discount information and catalog when ordering. Mail Dealer orders ATT: Wayne Green Books Dealer Sales.

WAYNE GREEN BOOKS
A division of Wayne Green Inc
Pine Street
Peterborough, NH 03458

Encyclopedia Loader

The editors at Wayne Green Books want to help you use the programs in your *Encyclopedia for the TRS-80**. So to help you maximize the use of your microcomputing time, we created **Encyclopedia Loader™.**

By a special arrangement with Instant Software™, Wayne Green Books can now provide you with selected programs contained in each volume of the *Encyclopedia for the TRS-80** on a special series of cassettes called **Encyclopedia Loader™**. Your encyclopedia provides the essential documentation but now you'll be able to load the programs instantly.

With **Encyclopedia Loader™** you'll save hours of keyboard time and eliminate the aggravating search for typos.

Save money with this introductory offer. **Encyclopedia Loader™** for Volume 1 of *Encyclopedia for the TRS-80** which will normally cost $14.95 is available **for a limited time only at the introductory price of $12.95.**

To order specify EL8001 $12.95.

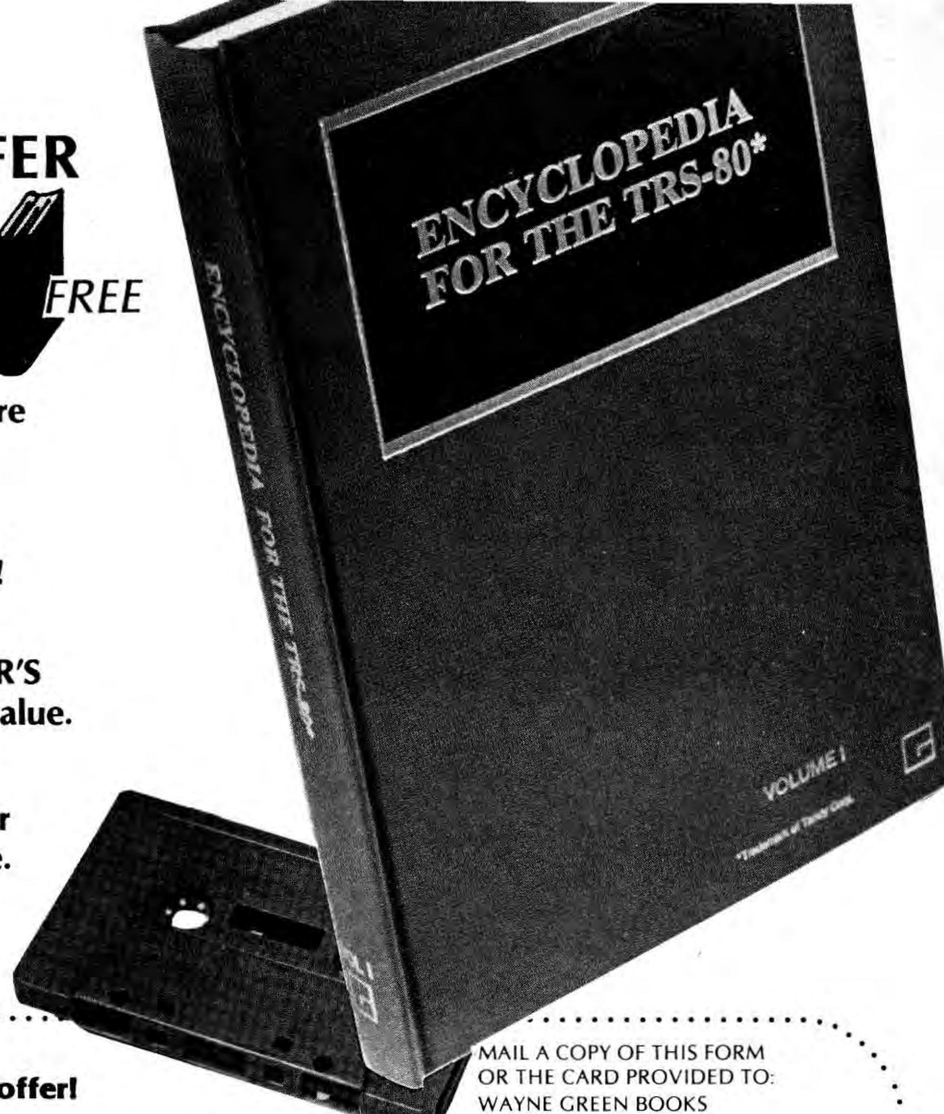
The Encyclopedia for the TRS-80 is a Wayne Green publication. Encyclopedia Loader is manufactured by Instant Software, a division of Wayne Green Inc.
* TRS-80 is a trademark of Radio Shack division of Tandy Corp.

SPECIAL OFFER

If you order the entire ten volume set of the Encyclopedia for the TRS-80,* you'll receive the tenth volume FREE!

10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value.
$164.00 EN8100 postpaid

10 Volume soft cover edition, a $109.50 value.
$83.00 EN8080 postpaid



YES, I want your special offer!

____10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value. EN8100 $164.00

____10 Volume soft cover edition —a $109.50 value. EN8080 $83.00

MAIL A COPY OF THIS FORM OR THE CARD PROVIDED TO:
WAYNE GREEN BOOKS
SALES DEPT., PINE ST.
PETERBOROUGH, NH 03458
OR CALL **TOLL FREE**
1-800-258-5473

To order single hard cover volumes of the *Encyclopedia* please specify:

____	Volume 1	EN8101 $19.95*
____	Volume 2	EN8102 $19.95*
____	Volume 3	EN8103 $19.95*
____	Volume 4	EN8104 $19.95*
____	Volume 5	EN8105 $19.95*
____	Volume 6	EN8106 $19.95*
____	Volume 7	EN8107 $19.95*
____	Volume 8	EN8108 $19.95*
____	Volume 9	EN8109 $19.95*
____	Volume 10	EN8110 $19.95*

To order single soft cover volumes of the *Encyclopedia* please specify:

____	Volume 1	EN8081 $10.95*
____	Volume 2	EN8082 $10.95*
____	Volume 3	EN8083 $10.95*
____	Volume 4	EN8084 $10.95*
____	Volume 5	EN8085 $10.95*
____	Volume 6	EN8086 $10.95*
____	Volume 7	EN8087 $10.95*
____	Volume 8	EN8088 $10.95*
____	Volume 9	EN8089 $10.95*
____	Volume 10	EN8090 $10.95*

To order **Encyclopedia Loader™** cassettes please specify:

____	Volume 1	EL8001 $12.95*
____	Volume 2	EL8002 $14.95*
____	Volume 3	EL8003 $14.95*
____	Volume 4	EL8004 $14.95*
____	Volume 5	EL8005 $14.95*
____	Volume 6	EL8006 $14.95*
____	Volume 7	EL8007 $14.95*
____	Volume 8	EL8008 $14.95*
____	Volume 9	EL8009 $14.95*
____	Volume 10	EL8010 $14.95*

PAYMENT ENCLOSED______ PLEASE CHARGE TO MY ____VISA ____MASTERCARD ____AMEX

CARD # ________________________ INTERBANK #__________ EXPIRES__________

NAME ____

ADDRESS ____

CITY ______________________ STATE__________ ZIP__________

SIGNATURE ____

*These prices do not include shipping and handling. Please add $1.50 per item in the U.S. and Canada. Allow 4-6 weeks after publication date for delivery. (All volumes will be shipped UPS if the complete street address is provided; otherwise shipment will be 4th class book rate.) Foreign orders please add $10.00 for airmail per item or $5.00 for surface mail per item. Delivery outside the U.S. varies. Prices are subject to change without notice. No C.O.D.'s.

8002

SOFT BITS
a basic/assembly column

by Roger Fuller

"I did not set any more machine language for nearly a year.

During a phone conversation about a short piece of code I was assaulted by malapropisms spoken in a vain attempt to impress me with the caller's expertise. Ignorance is a prerequisite for learning. To feign knowledge you do not have is to prepare a cream pie for later application to your face. This bit of philosophy precedes a discussion of a few words used and misused in reference to assembly and machine language programming. While I cannot make the issues black and white, I can reduce the amount of greyness associated with their use.

Memorize This List!

A *pointer* is an address which points to a specific location in memory. This location (but not necessarily the pointer) is in read/write memory. An example of a pointer is address 40A7H–40A8H, which contains the LSB and MSB of the keyboard buffer location. ROM uses this address to locate character storage during Input or List. A pointer does not direct program flow.

A *vector* is an address which directs program flow. For example, the vector at 401EH–401FH contains the address of the video driver. ROM uses this address in processing a video output operation. Since control is directed to the driver, this address is a vector. By changing this address you can redirect program flow to another driver. This is how a lowercase driver can be implemented in a TRS-80.

A *flag* or *switch* represents a condition. An example of a flag is a traffic light. A motorist should make a decision on the basis of the light's condition but does not have to obey. The byte at address 411BH is a flag byte. It is used by ROM in processing the tracing function: If it is zero the trace is off.

"...I can reduce the amount of greyness associated with their use."

Similar to flags or switches are *control bytes*. A control byte will force an operation to occur. In Basic, the control byte 0AH causes a line feed.

All the above items can be arranged in memory in the form of *areas, regions, blocks,* or *tables.* All the information, pointers, vectors, flags, switches and control bytes are called the *reserved area* in the TRS-80. A *table* usually contains a well-ordered group of similar items. Basic variables are stored in the variable table.

A pointer used to reference a table is called an *index* if the pointer is relative to the table only. The IX and IY registers are not index registers but are used in an indexing scheme. The index itself is called an *offset*. In the Z80 the IX and IY are used as pointers and their associated offsets are the real indexes. *Displacement* is another word for index. Think of a pointer as an index to all of read/write memory. While that makes the IX and IY indexed pointer registers, they are commonly referred to as index registers.

Intro to Machine Language

I bought my TRS-80 in the summer of 1978. After a few months I upgraded to 16K Level II and had a TTY for a printer. The TTY was connected via a cassette port interface. The interface required a software driver, which printed brackets rather than the TRS-80's up arrows. Because I wanted the TTY printing to match the computer type, I decided to fix the driver program.

A disassembled listing of the driver and the RSM2 monitor program made it easy. First I figured out how the driver worked by reading the comments in the listing and looking up what the accompanying mnemonics meant in the editor/assembler manual. I concluded I only needed to substitute the up arrow TTY code for the TRS-80 character. I inserted 254,91,32, 2,62,94 into the data statements of the "POKE it in from Basic" driver. This is the assembly code:

```
CP  91
JR  Z,$+2
LD  A,94
```

That took five hours. I did not write any more machine language for nearly a year. My next attempt was to produce a driver for a correspondence code Selectric printer. This took a total of fifty hours; I had to

```
0  ' USR INPUT ROUTINE BY ROGER FULLER       PUBLIC DOMAIN
1 '
10 CLEAR 1000
20 GOSUB 65000
30 TEST$=STRING$(10,32)
40 IF PEEK(16396)-195 THEN U=USR(VARPTR(TEST$))
       ELSE U=USR1(VARPTR(TEST$))
50 PRINTCHR$(13)  "==========>"  TEST$
60 GOTO 30
65000 U=PEEK(16551) + 256 * PEEK(16552) +192
      : FOR B=0 TO 67 : READ BYTE : CKSUM=CKSUM + BYTE
      : POKE U+B, BYTE : NEXT
      : IF CKSUM<>6823 THEN PRINT "CHECKSUM ERROR" : STOP
65010 IF PEEK(16396)=195 THEN DEFUSR1=U : RETURN
        ELSE POKE 16526, U AND 255 : POKE 16527, U / 256 : RETURN
65020 DATA 205,127,10,229,70,72,35,94,35,86,235,62,14,205,51,0,205,73,0,254
,32,48,21,254,13,40,31,254,8,32,241,120,185,40,237,62,8,205,51,0,4,43,24,22
8,95,120,183,40,223,123,119,205,51,0,35,5,24,214,62,15,205,51,0,121,144,225
,119,201
```

Program Listing 1. User Input Routine.

If you *liked* MICROPROOF, then you'll *love* SON OF MICROPROOF:

ELECTRIC WEBSTER T.M.

The ultimate spelling checker.

IMPROVED:

- One-step proofing and correcting.
- Lists errors to screen or printer.
- Can display errors in context.
- Can display dictionary to locate correct spellings.
- New precise symbolic dictionary will not miss an error.
- Remarkably compact (50,000 word dictionary will fit on one 5 inch disk).
- Even FASTER than MICROPROOF (formerly the fastest available).
- Simple Grammatical Checking (Optional Feature).
- Hyphenates automatically (Optional Feature available for some Word Processing programs).

SELECT APPROPRIATE RESPONSE:

CORRECT MISSPELLED WORD:	ENTER CORRECT WORD
LEAVE WORD "AS IS":	HIT <ENTER> KEY
DISPLAY WORD IN CONTEXT:	?
DISPLAY DICTIONARY:	@
ADD WORD TO DICTIONARY:	+
EXIT:	!

WORD: (Your error)
RESPONSE:

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 ELECTRIC WEBSTER proofreads your document, displaying misspellings and typos on the screen. Then correcting ELECTRIC WEBSTER can display each error separately, requesting you to enter the correct spelling for each. You are also given the option of displaying errors in context or adding words to ELECTRIC WEBSTER's 50,000 word vocabulary. If you do not know the correct spelling you may also ask ELECTRIC WEBSTER to look it up for you and display the dictionary. Finally, ELECTRIC WEBSTER corrects your document. All in less than a minute.

LOW PRICES: Standard MICROPROOF is available for $69.50 (TRS-80). Standard ELECTRIC WEBSTER is available for either $89.50 (TRS-80 Model I or III, Apple) or $149.50 (CP/M, TRS-80 Model II and all others). The optional correcting feature can be added at any time for an additional $60. Correction feature can be ordered with patch to integrate with your word processing software. For each patch, optional Grammatical Checking feature, or optional Hyphenation feature, add $35. (Integration patch not necessary for Wordstar.TM)

SPEED is the single most important factor in a dictionary program. All dictionary programs will find your potential errors but if the program is too slow, you are not likely to use it. ELECTRIC WEBSTER's speed is outstanding. It can proofread a several page letter in 20 seconds.

ELECTRIC WEBSTER's FULL 50,000 WORD VOCABULARY saves you time and allows you greater confidence in the lists of potential errors that ELECTRIC WEBSTER identifies. The mini-dictionary programs, with their 10,000 and 20,000 word vocabularies, have many correctly spelled words omitted from their vocabularies. Consequently, they identify as potential "errors" many words that are actually spelled correctly; five to ten times as many such words as does ELECTRIC WEBSTER. So, when you use ELECTRIC WEBSTER, you will have far fewer extra words to evaluate, a major time savings. There will be less need to look up words in order to verify that they are in fact spelled correctly. The extra 30,000 words in ELECTRIC WEBSTER's vocabulary assures you confidence in the error lists that Electric Webster generates.

HERE'S WHAT THE REVIEWERS HAVE TO SAY ABOUT MICROPROOF:

"I have already found that the use of (MICROPROOF) has greatly enhanced the quality of my letters and reports. This is a very useful product and should be obtained by anyone who uses a word processor."

Michael Tannenbaum, CPA
80 Microcomputing, August 1981

"The summary review of this program? One word — Excellent. I highly recommend it for anyone using a word processor for any need — articles, manuals, reports, and even letters of substantial length."

A. A. Wicks - Program Previews
Computronics, September 1981

In a comparative review of proofreading programs (with smaller dictionaries), MICROPROOF was found to be considerably faster than all the others, when tested against a 400 word sample document.

Phillip Lemmons
BYTE Magazine, November 1981

"(MICROPROOF) operates with good speed and efficiency. A 1500 word document took 26 seconds to load, process, and proof when the program was run on a TRS-80 Model II under CP/M."
"Once the program is integrated, it is very friendly and any person able to use a word processing program can master it in moments."

Frank Derfler
Info-World, January 1982

See your local microcomputer dealer or write to:

NOW AVAILABLE FOR APPLE!

CORNUCOPIA SOFTWARE

378

Post Office Box 5028, Walnut Creek, California 94596 • (415) 524-8098 learn how to use the editor/assembler and the RS-232 interface. A short time later I sold this driver; it was installed into a DOS written in Basic by none other than the legendary Jesse Bob Overholt.

Z80 Loads

One of the most common Z80 instructions are load instructions. (In fact, 19 percent of the Z80's instructions are loads.) Below is a list of functions and capabilities which apply to all 132 load instructions.

```
LD   DESTINATION , SOURCE
```

- All load instructions follow the above format.
- Pointers are always enclosed in parentheses.
- Source and destination cannot both be specified by pointers.
- The source is never destroyed by a load.
- A register pair to register pair load is impossible except:

```
LD   SP,HL
LD   SP,IX
LD   SP,IY
```

Pointers can be either a register pair or an immediate address contained in the instruction itself. Pointers can be for source or destination. If a source is specified by a pointer it is pronounced "Load Destination from where Source." If a destination is specified by a pointer it is pronounced "Load where Destination with Source." For this scheme to work, an immediate address used as a pointer is considered to point to the location in read/write memory specified by that address. See the examples in Fig. 1.

```
LD    A,(HL)      —load A from where HL
LD    (HL),A      —load where HL with A
LD    A,B         —load A with B
LD    (IX+4),A    —load where IX offset 4 with A
LD    A,(IX+4)    —load A from where IX offset 4
LD    DE,HL       —load DE with HL
```

Fig. 1.

Loads involving immediate addresses can confuse DE with 0DEH and BC with 0BCH. This is no problem for disassemblers because they work backwards from the instruction code itself. It is no problem in an assembler since hex numbers must always begin with a numeric digit. To avoid such confusion in speaking, use the suffix "hex" after an immediate address. Do not forget that immediate addresses point to a specified memory location, not the location contained in the instruction (like register pair). See Fig. 2.

```
LD    DE,(42E9H) —load DE from 42E9 hex
LD    (42E9),DE  —load 42E9 hex with DE
LD    DE.42E9H   —load DE with 42E9H hex
```

Fig. 2.

By now you should have a good grasp on how loading works in the Z80. As a test of your understanding, what is wrong with "load 42E9 hex from where DE?" If you can't explain you had better read the lesson over.

Four instructions in the Z80 involve loads but don't conform to the above rules. Think of these four instructions (block moves LDD, LDDR, LDI, and LDIR) as macro instructions. That is, they can be made from a group of other less powerful instructions. For example LDIR could be made as in Fig. 3.

```
              00100 ; USR INPUT SOURCE CODE  BY ROGER FULLER   PUBLIC DOMAI
N
              00110 ;
7F00          00120        ORG     7F00H          ;ARBITRARY LOCATION
              00121 ;
              00122 ;      ROM ROUTINES  MODEL I AND MODEL III COMPATIBLE
              00123 ;*****************************************************
0033          00130 DSPLY  EQU     0033H          ;VIDEO DISPLAY
0049          00140 KEYBD  EQU     0049H          ;INPUT A CHAR FROM KEYB
OARD
              00141 ;*****************************************************
              00150        ;
7F00 CD7F0A   00160        CALL    0A7FH          ;GET STRING DESCRIPTOR
ADDRESS
7F03 E5       00170        PUSH    HL             ;SAVE LENGTH POINTER
7F04 46       00180        LD      B,(HL)         ;GET STRING LENGTH
7F05 48       00190        LD      C,B            ;C=INPUT BUFFER LENGTH
7F06 23       00200        INC     HL
7F07 5E       00210        LD      E,(HL)         ;GET LSB OF STRING ADDR
ESS
7F08 23       00220        INC     HL
7F09 56       00230        LD      D,(HL)         ;GET MSB OF STRING ADDR
ESS
7F0A EB       00240        EX      DE,HL          ;ADDRESS TO HL
7F0B 3E0E     00250        LD      A,0EH
7F0D CD3300   00260        CALL    DSPLY          ;TURN ON CURSOR
7F10 CD4900   00270 NPUT   CALL    KEYBD          ;GET A CHAR
7F13 FE20     00280        CP      20H
7F15 3015     00290        JR      NC,ASCII       ;IF A PRINTABLE
7F17 FE0D     00300        CP      0DH
7F19 281F     00310        JR      Z,EXITCR       ;IF A CARRAIGE RETURN
7F1B FE08     00320        CP      08H
7F1D 20F1     00330        JR      NZ,NPUT        ;IF NOT A BACK SPACE
7F1F 78       00340        LD      A,B            ;   ELSE GET INPUT LENG
TH
7F20 B9       00350        CP      C              ;COMPARE TO ORIGINAL LE
NGTH
7F21 28ED     00360        JR      Z,NPUT         ;IF NOTHING IN BUFFER
7F23 3E08     00370        LD      A,08H
7F25 CD3300   00380        CALL    DSPLY          ;   ELSE BACKUP ON SCRE
EN
7F28 04       00390        INC     B              ;INCREASE ROOM LEFT IN
BUFFER
7F29 2B       00400        DEC     HL             ;BACKUP IN BUFFER
7F2A 18E4     00410        JR      NPUT
7F2C 5F       00420 ASCII  LD      E,A            ;SAVE CHAR
7F2D 78       00430        LD      A,B
7F2E B7       00440        OR      A              ;CHECK FOR ROOM IN BUFF
ER
7F2F 28DF     00450        JR      Z,NPUT         ; IF NONE LOOP BACK
7F31 7B       00460        LD      A,E
7F32 77       00470        LD      (HL),A         ;PUT CHAR IN BUFFER
7F33 CD3300   00480        CALL    DSPLY          ;   AND ON SCREEN
7F36 23       00490        INC     HL             ;BUMP BUFFER POINTER
7F37 05       00500        DEC     B              ;DROP ROOM LEFT IN BUFF
ER
7F38 18D6     00510        JR      NPUT
7F3A 3E0F     00520 EXITCR LD      A,0FH          ;TURN OFF CURSOR
7F3C CD3300   00530        CALL    DSPLY
7F3F 79       00540        LD      A,C            ;GET ORIGINAL LENGTH
7F40 90       00550        SUB     B              ;REMOVE REMAINING LENGT
H
7F41 E1       00560        POP     HL             ;RETRIEVE STRING DECSRI
PTOR BLOCK
7F42 77       00570        LD      (HL),A         ;SET LENGTH OF STRING I
NPUTTED
7F43 C9       00580        RET
0000          00590        END
00000 TOTAL ERRORS
```

Program Listing 2. Source Code.

An easy way to remember this instruction is its function (a block move of at least one byte), and its setup (involving three register pairs). You would need to know where the code to be moved is now, where it is going, and how much to move. The HL (High Low address) is the source, the DE (DEstination) is where it's going, and the BC (Byte Count) is how much is moved. The LDI instruction transfers only one byte. Produce LDDR by decrementing DE and HL instead of incrementing them. LDD moves only one byte. In memory the LDIR moves the old block to the new location bottom up. The LDDR moves the old block top down.

Three more instruction types (Bit, Set, and RES) take up 35% of the instructions.

```
LDIR    LD      BC,BYTCNT     ;BYTCNT = number of bytes
to move
                              ;0000H = 64K
        LD      HL,SOURCE     ;SOURCE = source address
        LD        DE,DESTIN     ;DESTIN = destination
address
        PUSH    AF            ;save register pair
LOOP    LD      A,(HL)
        LD      (DE),A        ;transfer byte
        INC     HL            ;bump source
        INC     DE            ;bump destination
        DEC     BC            ;drop count number
        LD      A,B
        OR      C             ;see if BC = 0000H
        JR      Z,LOOP        ;if not loop til done
        POP     AF            ;restore register pair
```

Fig. 3.

Bit tests a single bit in a byte specified by a register or pointed to by HL, IX + offset, or IY + offset. The zero flag contains the results of the test. Thus,

```
LD     A,01H
BIT    0,A
```

would reset the zero flag since bit 1 in register A is not zero. The carry flag is not affected by this instruction. Set and RES does not affect any flags in setting or resetting the specified bit in the target byte. See Fig. 4.

```
BIT    1,A     —test bit 1 of A
BIT    1,(HL)  —test bit 1 of where HL
SET    0,B     —set bit 0 of B
RES    7,(IX + —reset bit 7 of where IX offset 23
```

Fig. 4

These four types of instructions (Load, Bit, Set and RES) cover 54 percent of the Z80's mighty instruction set.■

THE EXCLUSIVE ORACLE

by Dennis Kitsz

"Take a walk to your dining room and turn off that light dimmer."

Q. In the October 1981 issue of *80 Microcomputing* an article appeared describing a screen twitch, reminding me of a problem I have. Mine is not a twitch, but more of a small wave which travels slowly up the video display, disappearing at the top and reappearing at the bottom.

It seems to have no more effect than to annoy me over the fact that it spoils an otherwise almost perfect computer setup. I have a Model I, Level II, 48K, two-disk system with an Epson MX70 printer, and the problem has been with me since my 4K Level I days. I suspect a component in the video display, but my knowledge of electronics is scant.

Raymond T. Schreiner
Tuckerton, NJ

A. Take a walk to your dining room and turn off that light dimmer, Ray. The plastic case and unshielded cable used in the Model I monitor made it susceptible to electrical interference, which from your description appears to be the electronic dimmer syndrome. If it's not that, check for flukey fluorescent lights or even see if any of the neon pilot lights on your hi-fi have an effect.

Q. I tried to load programs from cassette. I let the recorder run for a few minutes and never saw an asterisk appear on the screen. I know there is something on the cassettes, because I can hear it when I take out the ear plug. I ran this program, which always printed 255:

```
10 OUT255,0:PRINTINP(255):GOTO10
```

I ran the program with the cassette cable connected and disconnected from the computer. The relay works and I can generate music. What is the problem? Can I fix it?

Ken M. Hickey
Redwood City, CA

A. I'll assume that the volume control of your cassette recorder was turned up to the right level. That would point to a failure in the cassette input circuitry; the most likely component to give up is Z4, an LM3900 operational amplifier. However, CR4, CR5, CR6 and CR7 might have been destroyed by too much input, part of the Z24 (74LS132) input latch might be damaged, one gate of Z44 (74LS367) might be out of commission, or any one of a host of small parts in the cassette input circuitry might be open or shorted.

You'll need a copy of the *Technical Reference Handbook* and some electronics skills (or a friend with either or both) to do the repairs. Try replacing Z4 first, then the diodes. If you have access to an oscilloscope, hook up the tape recorder and look for a point along Z4 that the signal disappears. If it looks good straight through to the input at pin 9 of Z24, then replace Z24 and/or Z44. Check sheet 2 of the TRS-80 schematic for details.

If you get everything back in working order, be careful not to overload the input. Some tape recorders (especially Panasonic and Sony) provide a very loud output that can wipe out low-level analog integrated circuits like Z4. Also, make sure R67 (a 100-ohm resistor) is soldered across the cassette input jack; this resistor keeps most of those super-loud outputs from damaging the system.

Q. Can the Micro Front Panel circuit (May) be modified to display in either hex or decimal on seven-segment LED displays?

Paul Rivera, Jr.
Rego Park, NY

A. It's great to have a short question for a change. The Micro Front Panel may be modified, but I suspect its usefulness would be seriously limited by this change. Here's why. Say the computer was working vigorously around addresses 4406 and 4222. If you used seven-segment displays, you would get the first "4" okay, but the second "4" and "2" would combine to look like an "8", the "2" and "0" would look like an "8", as would the "6" and "2".

However, the binary readout would show alternating 0100 0100 0000 0110 and 0100 0010 0010 0010. The leading 0100 would be bright, the combined 0100 and 0010 would result in a dimmer 0110, as would the dimmer 0000 + 0010 and the dimmer 0110 + 0010 with the third digit emphasized. From that you could make a closer guess—either 4222 and 4406 or 4422 and 4206 was being displayed. The relative intensity of the binary digits can tell a great deal which a randomly flickering batch of Arabic numerals might not.

If you insist on changing to hex digits, look into the Texas Instruments hexadecimal seven-segment displays with built-in binary-to-hex decoding; you can get them at surplus for around $4 each.

Q. What can be done to give the Model I's five-volt power supply a higher current rating? Is there a regulator that can be substituted for the present chip? If not, can another regulator circuit be added in parallel with the first?

Secondly, I plan to upgrade my Z80 CPU for a new Z80B, but I know that at six MHz, my 80's keys may bounce right off their circuit board. Do you know of a clock-independent bounceless switch circuit that may work? Would a circuit similar to Radio Shack's cassette fix work here, only allowing the keyboard buffer to open for a specific time period? Or would it be easier to reprogram all my programs by increasing their timing loops?

William T. Faulkner
Marshfield, WI

A. I'd be curious to know why you would need to extend the TRS-80 power supply capacity. If it is to drive more chips, you might be relieved to know that the computer's power supply has a built-in reserve of about 150 mA—enough to piggyback an extra dozen LS TTL integrated circuits, or 16K more memory. With a bigger heat sink attached to the pass-transistors Q2, Q3 and Q6, plus perhaps a fan for extra ventilation, you can pull nearly half an amp more (though I wouldn't recommend that).

If you are serious about the added current, I would suggest a

✓See List of Advertisers on page 290

second power supply designed similar to the present Model I supply. You will get 1.5 amps or more of clean current, and the Model I "black box" transformer is inexpensive. Don't merely parallel the extra circuit with the first, though, because one supply will always try to do all the work, overloading itself. Instead, power the basic TRS-80 with its own supply, and power any peripheral or piggyback boards with the new supply. Keep the ground leads thick and short, and make sure that both supplies share a common ground nearby the regulators. Lastly, you will need additional ventilation.

Your second set of questions leads me to suspect that you're trying to create a "Super-80" from your trusty Model I. If you plan to install the Archbold Electronics speed-up kit, okay, but just upping the clock rate and getting a faster CPU isn't the answer for a homebrew speed-up. You'll have to consider:

- Do you want the 500-baud cassette rate? If not, do you have the XRX-II cassette mod? It will have to go.
- How fast are your memories? You will have to install a completely new memory refresh/select circuit using (preferably) a digital delay line. If you have a buffered cable, noise problems will seriously increase.
- Are you using disk? Can your DOS handle high speed, only low speed, or a mixture of speeds? Can it boot at high speed?
- Will you be switching back and forth between speeds? How?

As for the key-bounce question, high speed will make it much more prominent. The easiest solution is the Radio Shack Hall-effect (ALPS) keyboard, a $75 (but permanent) fix, completely assembled. Other options include purchasing a Cherry or similar high-quality keyboard, or perhaps an IBM pullout—those System 6 keyboards are solid and bounceless. Otherwise, you'll be forever afflicted with having to clean contacts or change all of your key-bounce delay routines.

A timed window like that used in the cassette mod isn't a viable solution unless you are a predictable and rather slow typist. You would shortly get out of sync and miss letters here and there. Even key debounce routines can slow up some typists; a fast typist can take 50 mS or less on common words like "the."

One option remains. You can use the keyboard column connections (see the *Technical Manual*) to trigger a one-shot flip-flop (such as a 74LS121) which will in turn activate a complete set of open-collector buffers—one for each key. Not a very economical solution.

• • •

I would like to share with you the insights of two *80 Microcomputing* readers—Art Brothers, a maverick telephone baron in Utah, and Richard L. Brocaw, for whom no amount of flickering lights can be too many.

Art (whose pioneer independence recently brightened the "Today" show) writes from Salt Lake City to expand on data transmission problems experienced in small or badly maintained telephone systems:

"A pair of wires stretched for miles and miles (over three) has unacceptable losses to high frequencies; the audio roll-off filtering to higher frequencies in the audio is unacceptable. Capacitive reactance (Xc) is added to the cable. To get rid of it, or balance it out, you have to add load coils, which provide inductive reactance (X1). The result is a bandpass to audio with sharp roll-off above about 3.5 KHz. It is easy to check this if you have someone somewhere in the system connect an audio oscillator to his phone through a 600-ohm or 900-ohm repeat coil, after having dialed you up. Then you connect a dB meter (HP 400 works fine, for sale at surplus for $35 to $50; or borrow some voltmeter with a dB scale or which will read down to a half volt or less, etc.); then send a tone at 500 Hz, again at 1000, and then at 2000 or 3000 and see what you get. If you get lots of roll-off at 2 KHz, you've got troubles. If you are friendly with the telephone company, and they are bigger than mine, you can get someone at the test desk to send you the tones. . . and you measure."

Thanks, Art. By the way, Art tells me he'll string wire on anything—even a cow if it will stand still long enough—to get a working telephone to his customers.

Finally, Richard Brocaw (Bosque Farms, New Mexico) loves micropyrotechnics. His letter:

"I thought the Micro Front Panel would be an interesting project. After all, who ever heard of a *real* computer without flashing lights? The big drawback of the circuit as presented is that the lights must be placed fairly close to the computer or perhaps plugged into the expansion port. I wanted to have the lights at some distance so they could be easily seen (see photos). But noise problems can really get you when you arbitrarily run an unterminated extension line about five feet long.

"With my LNW computer, the problem of placing the lights was greater than with the TRS-80. I used a piggyback board inside the computer containing all the ICs and a connector going

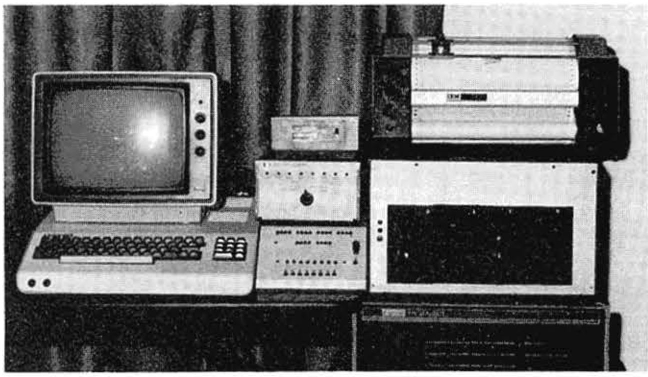


Photo 1. My complete system: The front panel is at the right of the CPU. My front panel also has a general-purpose eight-bit input and output port. Above the front panel is an RS-232 monitor and a running-time meter. To the right of that is a home-grown cabinet containing one 77-track and three 35-track drives. On top of that, an X-Y plotter.

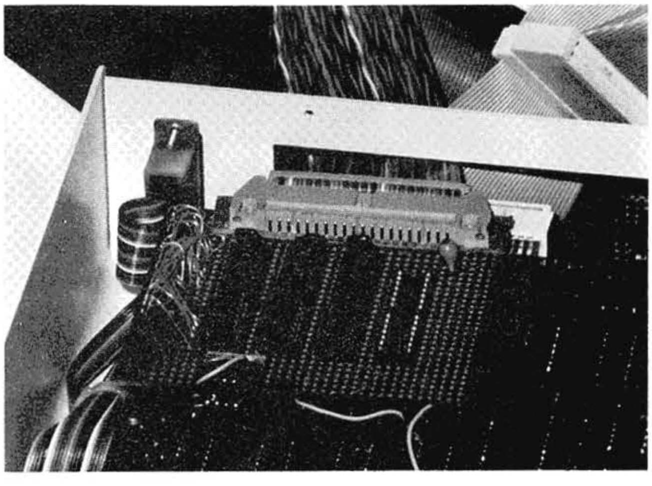


Photo 2. The piggyback board. The flat ribbon on the left is for the RS-232 signals from the LNW. The cable from the back of the board goes to the light box.

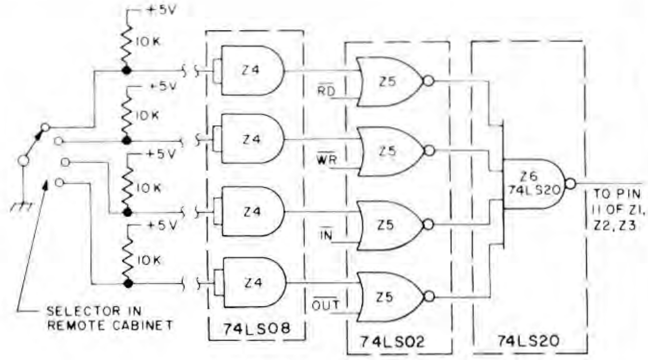


Fig. 1

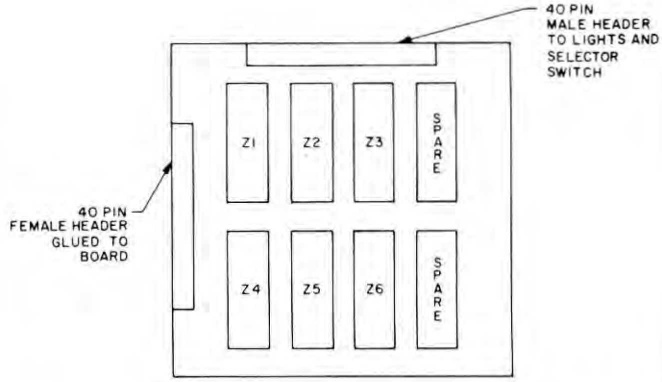


Fig. 2. A 40-pin, right angle, dual-row strip line converter was soldered to the expansion connection on the LNW. This board plugged in and is supported on the right by a ½-inch rubber grommet.

to the display. But what to do with the selector switch? I did not want to run important signals like RD* and WR* any further than necessary. My solution is a hardware switch remotely selected by TTL signals (see Figs.).

"My system consists of the 48K LNW computer, Doubler II and an NEC monitor. The front panel is to the right; it also has a general-purpose eight-bit input and output port. Above the front panel is an RS-232 monitor and running-time meter; to the right is a homemade cabinet with one 77-track and three 35-track disk drives. On top of that is an X-Y plotter."

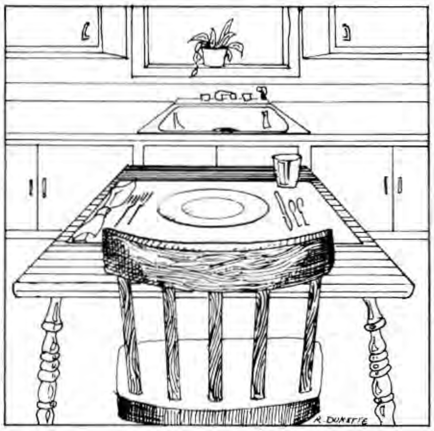
Well, Richard, your system looks like real battleship computing (whatever happened to the "micro" in Micro Front Panel?). Seriously, this is an excellent solution to a general-purpose remote-switching problem, and seeing Richard's system should give readers a surprising glimpse of a complete non-TRS-80 TRS-80-based system. ■

Desperate? Send your questions on TRS-80 Model I or Color Computers to Dennis Kitsz, Roxbury, VT 05669. Please describe your complete system, and the problem in as much detail as possible, including all tests you have made and their results. I'll try to give you a personal reply if you enclose a self-addressed, stamped envelope (two international postal reply coupons outside U.S., Canada or Mexico).

✓See List of Advertisers on page 290

News From KITCHEN TABLE SOFTWARE, INC.

by David Busch



Data storage technology took several quantum leaps forward when Kitchen Table Inc. introduced the world's first half-inch hard disk drive with 7500 megabytes of on-line storage.

At a press conference at KTI's sprawling corporate headquarters in suburban Atwater, OH, the firm unveiled its diminutive drive to be sold to customers for $950; to OEMs buying 10 or more for $57.49.

Regular readers of *80 Micro* will recall KTI is a fictitious supplier of non-existent space age products. These include the TLS-8E microcomputer, DROSSDOS 7.3, and OPENSESAME—the world's first voice-actuated garage-door opener.

The company provided both reporters attending the session with working prototypes of the KHD-7500 drive attached to a souvenir keychain.

An intense question and answer period unearthed the following tidbits:

- The KHD-7500 measures 1 by 3-1/2 inches and will fit a standard 40-pin DIP socket. Unlike larger drives, the KTI unit may be installed inside any TRS-80 Model I's expansion interface, Model III's keyboard, TLS-8E models I–IV or Sinclair ZX81.

It is rumored KTI is working on incorporating the drive in a slightly larger version of their slow-selling Pockets Computer. If the plan comes to fruition, the PC will be possibly the only hand-held computer in the world able to store 7500 megabytes of information.

- The KHD-7500 was conceived by accident. Kitchen Table's U.S. management sent a cablegram to the firm's research facility in Sri Lanka asking it to design a 3.5-inch drive compatible with Sony's mini-mini-floppy system. But when the first prototypes were shipped to the United States earlier this year, KTI officials realized the "3" had been deleted by mistake from the cable.

- Installing the KHD-7500 is child's play, although most adults can also do the job. Pry loose the 1771 disk controller in your TRS-80 or TLS-8E and replace it with the KTI hard disk device. The old controller may be sold to any friend who was careless when installing his or her data separator.

"The non-erasability of the media should be welcomed by programmers."

The KHD-7500 includes a "piggyback" 40-pin DIP socket so the drives may be stacked up to the space limits in your computer. If you own a TLS-8E with the improved 1250-watt power supply, you may add eight or more drives to your machine. However, TRS-80 owners, after piggybacking three or four drives, will probably blow a fuse. Kitchen Table also cautions there is a small chance the heat generated by a multiple configuration may melt a few non-critical solder runs.

- Once data has been written to disk, it is impossible to erase the information. A rachet mechanism prevents the write head from returning to a recorded track. The read head, on the other hand, can access any track. Once the disk has been filled, you must expand your system by adding another drive.

However, KTI points out that with 7500 megabytes of storage, you could fill a megabyte of space daily and not deplete the KHD-7500 for more than 20 years.

Such storage capabilities create the possibility of building respectable microcomputer data bases. Those of you who have mailing lists of every household in mainland China must swap disks like crazy every time you sort or update. No more. The capacity of the KHD-7500 is so large, KTI included as a free demo program all the U.S. census data from 1890 to date.

The non-erasability of the media should be welcomed by programmers. Usually, halfway through the 15th enhancement of a complex piece of software, the programmer realizes the program has taken a wrong turn and he or she needs to return to an earlier version of it.

The KTI operating system (DRECKDOS 1.0), introduced especially for the KHD-7500, automatically appends an "invisible" numeric extension to files. By entering "LOAD filename/ext PREVIOUS," the computer will search and find the most recent version of a file.

It required sophisticated technology—most of it proprietary—to develop the KHD-7500. Kitchen Table engineers have provided me with detailed information on the drive, but I have sworn to keep it a secret. I am honoring my pledge by sharing the information with only *80 Micro* readers, so if David Ahl calls any of you, just politely hang up without telling him anything.

The key to the ultra-high density of the KHD-7500 is the low flying height—measured in tenths of an angstrom—of its write head. This creates the problem of large air molecules being too big to squeeze between the head and the disk. Carbon monoxide (CO) is okay but carbon dioxide (CO_2) can cause some horrible head crashes.

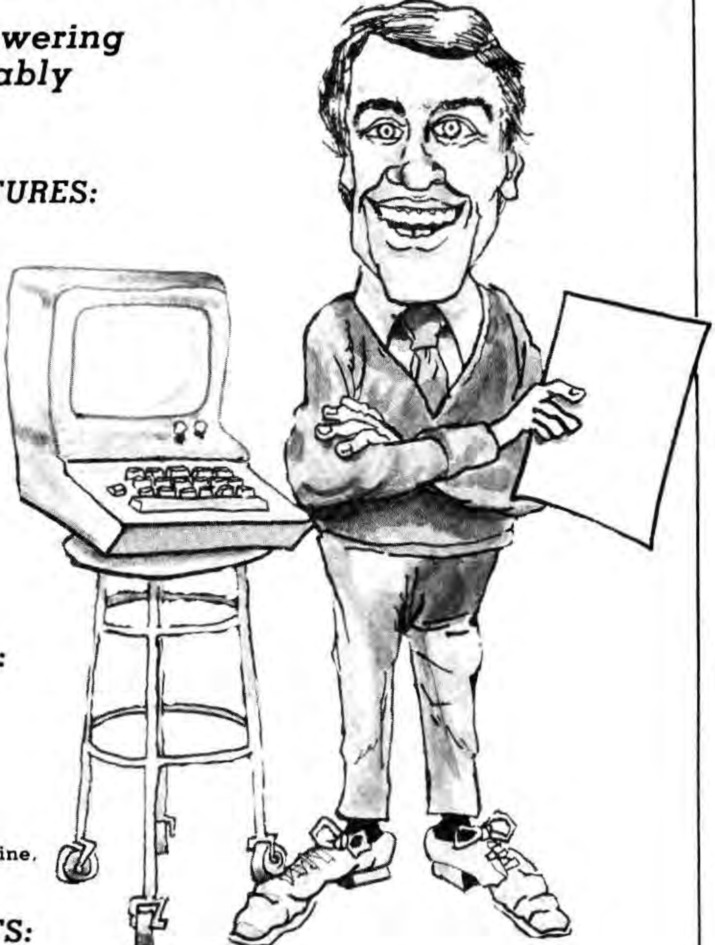
Kitchen Table's solution to that problem was to create a near-perfect vacuum in the disk chamber. The disk "floats" on a tiny pad of Teflon II and is driven at 13,000 rpm by an electromagnetic field that "grips" the disk's center but does not leak over to spoil the data.

This unique process has not been patented by KTI. Instead, each KHD-7500 is rigged to explode when dismantled or x-rayed. (Don't take your computer through an airport security area!)

HAPPINESS IS...NEWSCRIPT™

THE WORD PROCESSOR FOR BUSINESSMEN AND PROFESSIONALS

We offer you ongoing support by answering your questions and providing reasonably priced enhancements.



A FEW OF NEWSCRIPT'S STANDARD FEATURES:

*Form Letters with merging of names and addresses
*Gives superb appearance to your final documents
*Comprehensive manual with hundreds of examples
*Centering, top/bottom titles, indents, pagination
*Underlining, boldface, double-width, italics†
*Sub-scripts, super-scripts, proportional pitch†
*Generates Table of Contents, sorted Index
*Allows block graphics, special symbols†

*Search and replace globally or within a range
*Block move, copy, delete, insert from other file
*AUTOSAVE, WHOOPS, DIRECTORY, KILL, REPEAT

*Based on IBM's "EDGAR" and "SCRIPT" systems
*Supplied ready-to-run on "tiny" DOSPLUS
*Easily transferred to NEWDOS, NEWDOS/80, LDOS, TRSDOS
*Includes handy Quick Reference Card

NEWSCRIPT CONTROLS THESE PRINTERS:

*EPSON MX-80, MX-100: all 12 fonts, plus underlining and block graphics; italics with GRAFTRAX

*Centronics 737, 739; Radio Shack L.P. IV, L.P. VIII, Daisy Wheel II, NEC PC-8023A, TEC 8500R, C.ITOH 8510 Right-justified true proportional printing (ledding)

*Good support for: Diablo, Spinwriter, Starwriter, QUME, Microline, Anadex, modified Selectric

TYPICAL USER AND REVIEWER COMMENTS:

the manual:
"It definitely rates the first '10' given to any documentation reviewed in this column." (A.A. Wicks, COMPUTRONICS, October, 1981)

the software: "An excellent Word Processor" (D.H.); "Absolutely fantastic" (S.E.S.); "You have features that I cannot duplicate on my $14,000 system" (J.B.)

the support: "Your phone information system and the prompt and courteous staff that you provide to help your clients...are worth the cost of the system." (V.H.H.)

REQUIRED CONFIGURATION: 48K TRS-80 with one disk drive. Specify Model I or Model III.

† Some features work only if your printer has the mechanical capability.

NEWSCRIPT:	**$99.95**
MAILING LABELS OPTION:	**29.95**
Special: NEWSCRIPT + LABELS:	**115.00**
"PENCIL" to NEWSCRIPT Convertor	**14.95**
Manual Only (180 pages):	**30.00**
Reference Card Only:	**1.50**
MICROPROOF + CORRECTION FEATURE:	**149.95**
Full DOSPLUS *3.4* Operating System:	**149.95**

TO ORDER, CALL NOW, TOLL-FREE:

(800) 824-7888, Operator 422
CALIF: (800) 852-7777, Oper. 422
ALASKA/HAWAII: (800) 824-7919
FOR TECHNICAL INFORMATION CALL:
(213) 764-3131, or write to us.

Order from your Software dealer, or from:

PROSOFT™

DEPT. C, BOX 839 • NO. HOLLYWOOD, CA 91603

TERMS: We accept VISA, Mastercard, checks, Money Orders, C.O.D. and even cash. We pay shipping via surface UPS inside U.S.A. Please add $3.00 for Blue Label, 6% tax in California, and 15% outside North America (air shipment).

KITCHEN TABLE SOFTWARE, INC.

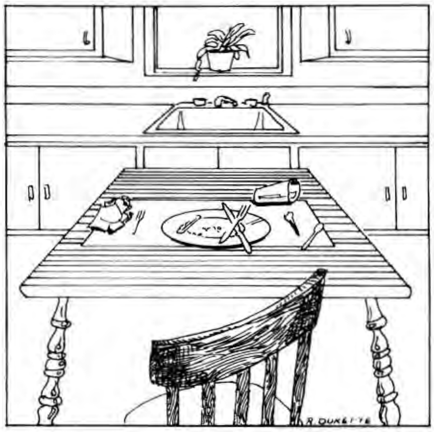
Since most microcomputer users will be unable initially to purchase two KHD-7500s, how can the single-drive user back up 7.5 gigabytes of data conveniently? The TLS-8E's eight-track tape system or the cassette port of the TRS-80s may be used for disk dumps to cartridge or cassettes.

I pointed out to KTI's head engineer that at 500 baud it would take 174 days to dump 7500 megabytes onto nearly 4200 C-60 cassette tapes. The engineer chided me for resorting to a worst case example. It was more likely, he said, a user would dump only a megabyte or two at a time and for that size dump, a few times a week would be sufficient.

As a higher-speed option, KTI offers an interface allowing disk dumps at 18,000 baud to a VHS or BETA videocassette system. It dropped plans for generating print-image tapes for a computer-output microfilmer when its research revealed few microcomputer users own or have access to these $75,000 devices.

Wayne Green Inc. is very excited about the possibilities of the new half-inch drives. Three of them will hold the subscription lists for *80 Micro, 73 Magazine* and *Kilobaud Microcomputing*—with room left over for *Desktop Computing*.

And Wayne Green is busy working on a plan for marketing the devices under the brand name Superload 80. Sold for $995, each drive will be loaded with every Z80 compatible program ever written. How's that for instant software? ■



PROFESSIONAL

INCOME TAX SYSTEM

FOR TRS-80* MODEL I, II or III

For four successive years!

Our system, running on the magnificent line of TRS-80 computers, has prepared thousands of lightning-fast, error-free tax returns a day in accounting offices nationwide.

Now we introduce!

SYSTEM THREE!

SYSTEM THREE is an improved, still more interactive version — Totals W-2's and computes FICA overpayment — Includes "QWIKTAX," a fast tax calculator for tax planning — Has automated Client Billing — More!

And, as always—

1. Full interactive user control, in tax-form language only, line-by-line.
2. Screen display of full 1040 and all schedules, prior to printout.
3. All formats IRS and State approved.
4. Schedule amounts automatically entered on Form 1040.
5. Your Preparer's Information automatically printed at bottom of Page 2, Form 1040.
6. Built-in Validation Check tests entire system, hardware and software.
7. Special printer adjustment routines: Margin Offset, Text Position, etc.
8. Fills in pre-printed forms (we supply) or use overlays. Your choice.
9. AUTOMATICALLY COMPUTES: Tax — Earned Income Credit — Maximum/Minimum Tax — Least Tax Method — All Percentage of Income Limitations — All Fixed Limitations.
   DOES INCOME AVERAGING IN EIGHT SECONDS!
10. Full support through the tax season, no charge.
11. Inexpensive yearly updates in accordance with tax-law changes.

SYSTEM THREE comes in four modules. You select the ones you need. There is no system like it, for any computer, and the cost is literally nothing. In one season, our users have paid, out of savings, for the software ***and* the computer.**

35-page Descriptive Manual - $7.50
(Refundable on order)

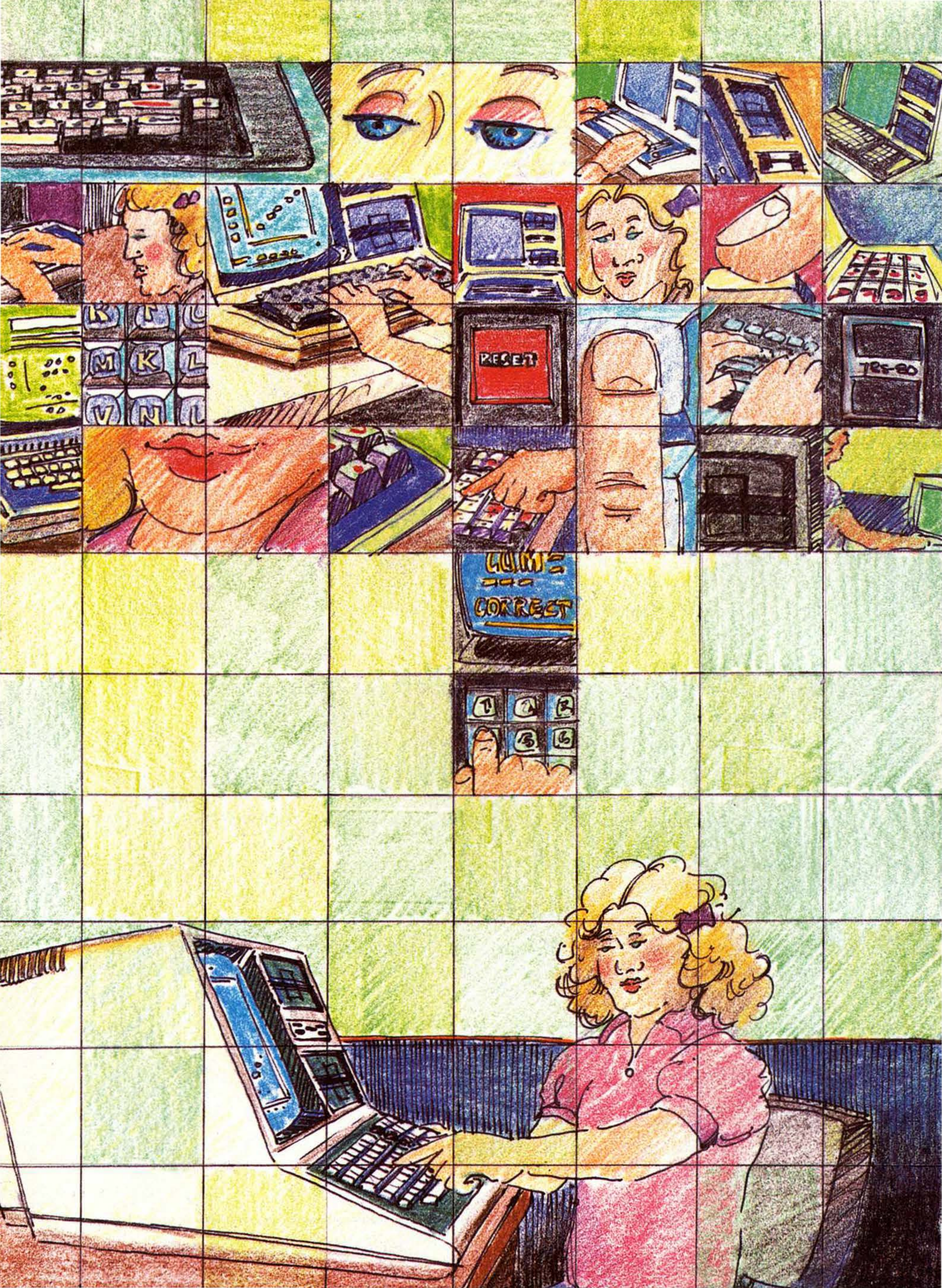
MINIMUM SYSTEM REQUIRED: Model I, 48K, 1 Disk Drive

CONTRACT SERVICES ASSOCIATES ✓10

706 SOUTH EUCLID • ANAHEIM, CA 92802 • TELEPHONE: (714) 635-4055

*TRS-80 is a trademark of Tandy Corp.

✓See List of Advertisers on page 290


RESET
TRS-80
CORRECT

The micro in the classroom is more than a microcosm.

The Future in Miniature

by John P. Mello Jr.
80 Micro Staff

The schools of the country are its future in miniature."
—Tehyi Hsieh

Call it what you will: The information explosion, the post-industrial revolution; it's here, it's growing, and where it and its standard-bearer, the computer, take our schools will be where they take our future.

The digital generation is upon us. A generation of school children have already been taught to accept the computer, according to a report released by the Office of Technology Assessment of the U.S. Congress. It observed: "The first generation to grow up with computers is now reaching maturity. These 'computer literate' young adults accept computers as a natural part of their world."

Make no mistake about it. Computers are in the classroom now and more will be there in the future. A survey last year of 15,442 school districts showed 42 percent of them use instructional computers. More and more schools will be buying computers selling for less than $10,000, according to Market Data Retrieval, which conducted the survey. The Westport CT firm predicted the school computer market, which was $35 million in 1980, would explode to $145 million by 1985.

"People are still going to buy building blocks and erector sets and coloring books."
—Alan Chernoff, Mego Toys

Computers are not only used to teach children in school, but out of school, as well, through educational games like the Children's Discovery System by Mattel and Speak and Spell by Texas Instruments.

Questions have been raised about the value of these games, which represent an $85 million market. In a report in the *New York Times*, Ken Komoski—an associate professor at the teachers' college at Columbia University and executive director of the Education Products Information Exchange, an independent consumer agency that has completed a study of learning aids for the Ford Foundation—observed:

"One generalization is that while most of these games are awfully good practice for what a youngster already knows, unless there is an adult willing to work with the youngster to create a set of activities, they will not learn very much that is new. These products are valuable for practice devices for what is already in the repertory."

At Texas Instruments, they are careful what they say about their learning games, Director of the Learning Center at TI, Dr. Ralph Oliva, told the *Times*: "We are careful not to overclaim. Our claim is that our products are valuable aids for drill and practice. They are an enrichment tool using new technology for parents and teachers, not a replacement for a classroom."

"It is very hard to grab a child's attention and keep it," Milton Bradley Senior Vice President of Marketing George R. Ditomassi, Jr., said in the *Times* report. "Electronics has allowed us to do this because responses keep coming from the product and it offers a tremendous amount of variety. Electronic technology dramatically extends a child's attention span."

Michael Katz, vice president of marketing at Coleco, added in the report, "Our philosophy is that the magic of electronics—the sounds and lights—can put fun into education."

Does this bells and whistles approach to education—an approach akin to the one used in the Public Broadcasting System's "Sesame Street"—help a child in his or her studies? The answer isn't in yet, and the Office of Technology Assessment report asks a similar question: "Has Sesame Street improved the educational level of its viewers, or has it conditioned young children to a short attention span and to an orientation toward learning as entertainment, unsuitable for serious academic work?"

"Instead of believing they were stupid, the children understood and learned from their errors; the teachers could begin to build better ways to teach."
—John Seely Brown, Xerox Corp.

Although the bells and whistles approach may be questioned, the value of computers in the classroom is not.

At the University of California at Los Angeles, a computer program named Homer, after the Greek bard, is included in the school's $1.6 million war on bad writing. When a student logs on at UCLA, he or she is greeted with Dante's homage to Homer: *Honor the great poet—his shade, which has departed, now returns.*

The student enters a writing sample and Homer analyzes it for flaws—overly long sentences, too many uses of the verb "to be," too many prepositions, awkward construction, and other symptoms of overblown prose.

Homer tends to be a witty fellow. He tells students, "What a horrid mess" or "For all I can tell, this is good writing—but then, what does a dead poet know?"

He also warns students: "Remember, do not take my comments as gospel—I am really just a machine. I only look at your writing's surface. You must supply the logical thought, effective organization, and above all, something worth saying."

When Homer is finished with the student's work, he admonishes the writer, "Go thou and sin no more."

"Corporations aren't the only ones worrying about the existing educational establishment providing adequate training in computer science. . ."

Meanwhile, in Massachusetts, an experiment is being conducted on slow readers in six school districts. One thousand sixty-seven students were divided into two groups. Each group received half-hour reading lessons, but one group's lesson included 10 minutes of drill on a computer. The results have been termed "significant." In grades two through nine, the mean gain in reading level for computer students was 7.00 compared to .72 for non-computer students. The most significant gains were by children in grades one and two: 17.94 for computer users, 8.98 for non-users.

"The first year I taught Title I students, we didn't have computers," Linda Sparks, a teacher at the Dutile School in Billerica, MA, told the *Boston Globe*. "Students would have to be called out of regular classes for extra drill and were made to feel different from their classmates. Now, non-Title I students ask if they can use computers, too."

Dracut, MA, Title I teacher Janice Souza added: "I am not surprised by the good results of the study. Most of my sixth and seventh graders made more than a year's gain in eight months last year and three of my students made fantastic two-year gains during the same period. Only two out of 37 didn't make much progress."

"The main difference that computers make to me is in diagnosing student's needs," she said. "They give me a better idea of what their skills are, even in a one shot test. . . ."

"The records kept on individual students by the computer would not be possible for many teachers, particularly those whose classes have been enlarged because of [budget cutting]," George Hanify, director of computer application for the Merrimack Education Center, told the *Globe*.

Although the future of computer applications in the schools lies in programs like the one in Massachusetts, some educators have expressed concern over the impact of federal budget cuts on such experiments.

But Peter Burchyns, coordinator of planning and evaluation, for the San Mateo County, CA, office of education, told the computer newspaper *InfoWorld* that reductions in federal money might encourage the use of computers because of the high cost of personnel in the school.

However, how long will it be before school budget cutters attempt to use computers to lay off teachers? Never, William Engel, director of the $50,000 Florida Center for Instructional Computing, told a Florida newspaper. "Computers will actually be able to make the teaching process more humane," he said. "If a teacher is teaching a class and not managing well, that's not a humane way to learn."

"They teach in academies far too many things, and far too much that is useless."
—Goethe

At least one university has made a significant commitment to the "humane way to learn" and if it is fulfilled, college education will never be the same again. Echoing a bygone political promise of a "chicken in every pot," the president of Carnegie-Mellon University in Pittsburgh announced last fall the school planned to provide a personal computer terminal for each student, starting in 1986.

College education hasn't been the same since the arrival of computers, anyway. Computer firms are siphoning off faculty from computer science departments and even starting their own quasi-colleges that may be diverting money from the financially hard-pressed institutions of higher learning.

Wang Laboratories, Bell & Howell Education Group and International Business Machines have started up their own schools. The Wang Institute—whose entering class included employees from Digital Equipment Corp., Prime Computer and Honeywell—offers a degree in software engineering. Bell & Howell, a wholly-owned subsidiary of Bell & Howell Co., offers degree programs in computer science for business.

It has been estimated IBM annually spends an average of $2,000 per employee for education. Despite that commitment, however, the corporation felt it had to establish its own graduate level school in New York City, The Systems Research Institute.

Former vice president for technical personnel development, Jerrier Haddad, told the *New York Times* the institute was designed to enable engineers to keep up with rapidly changing technology in areas they do not directly work in and to broaden their perspective.

The institute, he said, enables IBM to educate far more engineers than it can send to a university; it stresses IBM products and can be more up-to-date than universities.

However, Murray Turoff, a professor at the New Jersey Institute of Technology and at the IBM institute, maintained in the same *Times* report that commitments by IBM and other companies to their own educational programs might limit their contributions to universities, which are suffering from inadequate equipment and faculty shortages.

Diminishing university computer science departments could backfire on the corporate giants, the Office of Technology Assessment suggested. It said: "Because of this close connection between technology and research, the vitality of the computer industry is in part dependent on the vitality of academic computer science. However, university departments of computer science are experiencing problems."

"The direction which education starts a man will determine his future life."
—Plato

Corporations aren't the only ones worrying about the existing educational establishment providing adequate training in computer science; so are many parents of school-age children. Those parents may seek alternatives like Computer Camp East, in East Haddom, CT, and the Randolph School in Huntsville, AL, to give their children an "edge" in the future.

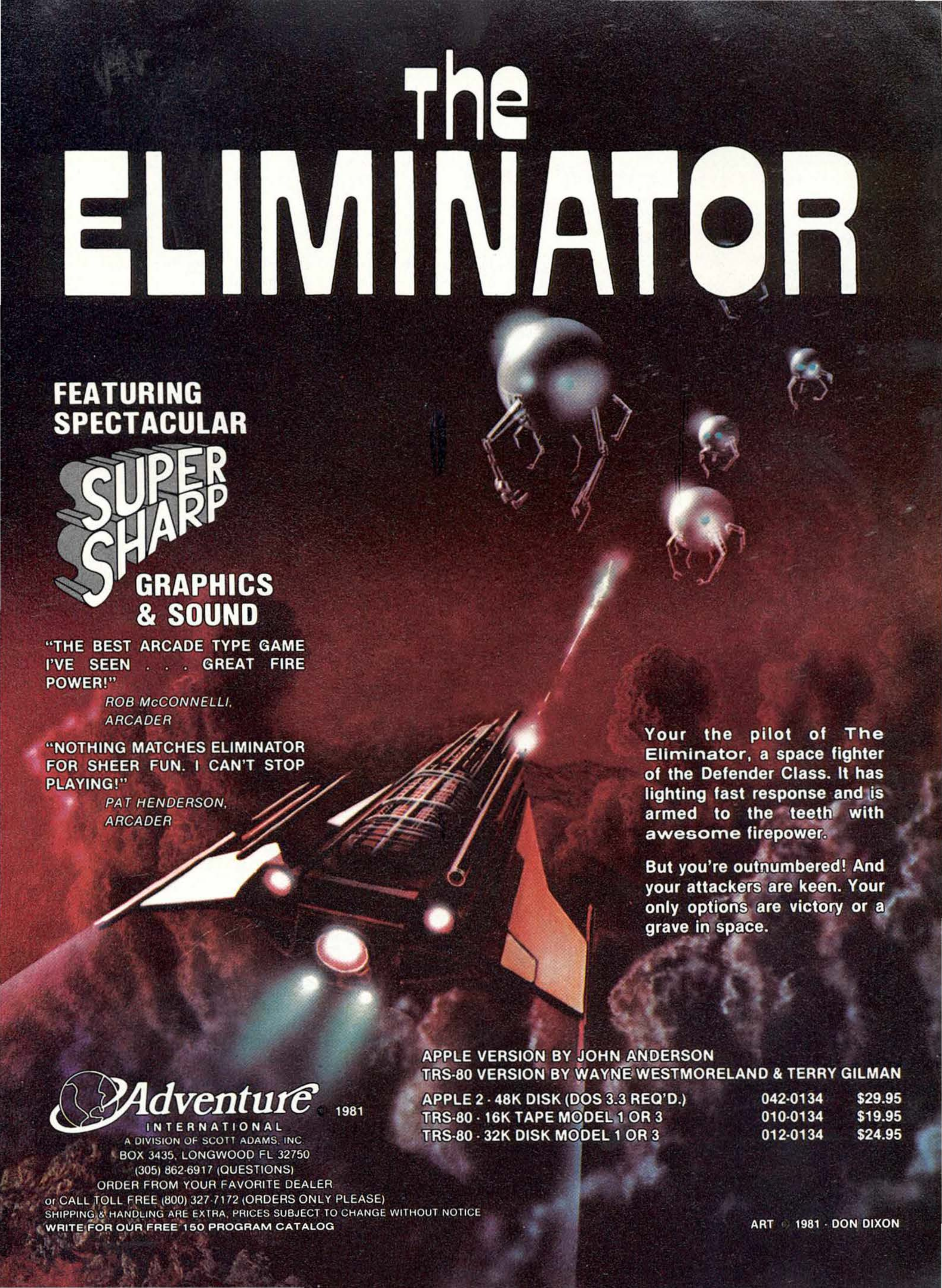
In a *New York Times* report, William Gibson, director of instruction at the computer camp, noted: "Parents are sending their children here because computer skills will be necessary no matter what field they get into. Parents are not confident schools are keeping up with this trend, and they want their kids to have a head start."

"Because of the role of the computer in society today," Randolph School Headmaster Dennis Brown told the *Huntsville Times*, "we feel that schools should assume the responsibility to expose students to them in a coordinated program."

The Randolph School's computer program runs from kindergarten to grade 12. "We have two levels in mind," said Jeff Farber, computer coordinator for the school. "For the elementary school, it is on a computer awareness level. For high school students, it goes into the function and application of computers in all disciplines."

As the approach adopted by Computer Camp East and the Randolph School becomes more prevalent, even traditional graduation ceremonies may be changed. William Geist of the *New York Times* described the send-off at the computer camp:

"After dinner came the final night's ceremonies, a time at most camps for bonfires, group song and ceremonies reminiscent of


the ELIMINATOR
FEATURING SPECTACULAR
SUPER SHARP
GRAPHICS & SOUND
"THE BEST ARCADE TYPE GAME I'VE SEEN . . . GREAT FIRE POWER!"
ROB McCONNELLI, ARCADER
"NOTHING MATCHES ELIMINATOR FOR SHEER FUN. I CAN'T STOP PLAYING!"
PAT HENDERSON, ARCADER
Your the pilot of The Eliminator, a space fighter of the Defender Class. It has lighting fast response and is armed to the teeth with awesome firepower.
But you're outnumbered! And your attackers are keen. Your only options are victory or a grave in space.
APPLE VERSION BY JOHN ANDERSON
TRS-80 VERSION BY WAYNE WESTMORELAND & TERRY GILMAN
APPLE 2 · 48K DISK (DOS 3.3 REQ'D.) 042-0134 $29.95
TRS-80 · 16K TAPE MODEL 1 OR 3 010-0134 $19.95
TRS-80 · 32K DISK MODEL 1 OR 3 012-0134 $24.95
Adventure INTERNATIONAL © 1981
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435, LONGWOOD FL 32750
(305) 862-6917 (QUESTIONS)
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172 (ORDERS ONLY PLEASE)
SHIPPING & HANDLING ARE EXTRA, PRICES SUBJECT TO CHANGE WITHOUT NOTICE
WRITE FOR OUR FREE 150 PROGRAM CATALOG
ART © 1981 · DON DIXON

". . . as the information society evolves, information literacy will become essential for employment."

Indian rituals."

"Here, however, the campers sat on picnic tables looking at a Quasar video monitor. As certificates of merit were handed out to each camper, a staff member attempted to play 'Pomp and Circumstance' on a computerized music synthesizer."

"The philosophic aim of education must be to get each one out of his isolated class and into the one humanity."
—Paul Goodman

Horace Mann, the father of public schools, said education was the balance wheel of the social machinery, but some observers feel the education system under the influence of computers will have just the opposite effect on society.

According to the Office of Technology Assessment report, "Some observers have suggested that the advent of information technology will widen the gulf between the haves and have-nots in society. This view is based on relative differences in what might be called information literacy, the ability to use information technology to cope with every-day life."

According to the Congressional report, as the information society evolves, information literacy will become essential for employment. However, obtaining information literacy today seems to be a function of wealth: Educational games range from $60 to $140; computer camp costs $795 for two weeks; and tuition at the Randolph School ranges from $1,000 to $2,250.

"The most important factors in the use of computers for instruction are wealth of the school and wealth of the population served by the school," according to the Market Data Retrieval survey.

"In school districts spending over $75 per student for instructional materials (the wealthiest districts in the United States), 46 percent have instructional computers. In school districts spending under $30 per student for instructional materials, the figure falls to 20 percent.

"Computer use per capita income of the school district's population follows the same pattern. Thirty percent of the schools in upper income areas use computers for instruction while only 12 percent of low income area school buildings have access to computer-aided instruction."

However, there may be a source of light for this bleak prospect. Computer technology has continued to lower the cost of computer-aided education, William Gattis, director of Tandy/Radio Shack's education division told *InfoWorld.* If that trend continues, universal access to micros could prove to be a social equalizer.■

TAKE A TEST DRIVE.

With a reliable TRAXX™100/200/300 series add-on drive system.

Our drives feature excellent engineering, and all of the advanced performance features you've come to expect from the nation's leading disk drive manufacturers. All systems are completely burned-in and tested. And, you'll see at least five quality assurance stamps on each and every drive, which is how we make sure our drives will run and will continue to run past our optional two year extended warranty.

Our drive packages start at $250.00 and include a comprehensive operations manual and an attractive static free, dust free cover.

Systems available for Altos, [1]Apple™, Atari, Heath™, North Star™, S-100, [2]TRS-80™ (Model I, II, III, Color), Zenith™.

If one of our drives fails to meet your highest expectations of how trouble free and reliable a disk drive can be, then return it to us for a complete refund.*

So, before you buy another drive, take a test drive with one of ours. We're sure that you'll find TRAXX to be the finest.

IT'S GUARANTEED!

TRAXX
COMPUTER CORPORATION

Call our toll-free TRAXX LINE: 1-800-621-3102. In Illinois, call: (312) 987-1024. 10AM-6PM CST. Monday thru Friday. *For full refund drives must be returned within 10 days of purchase.

[1] Apple is a registered trademark of Apple Computer Inc.
[2] TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

They don't call it Basic for nothing!

Anything Jodi Can Do . . .

Jodi Tallman
3468 SE Roanoke
Hillsboro, OR 97123

I was sitting in our family room one day reading a code book, and I heard my dad talking about writing programs. I wanted to write a program on our computer before my brother (he's older). I wanted to show my brother and dad you can be any age and write a program. I also wanted to learn more about computers.

My dad was always saying things about writing a program. I only knew how to *play* games, so I wanted to write one. I wanted to have fun with computers. I was getting bored because my mom was on vacation, so I didn't have anything to do but work on a program. My dad was also always talking about how much money you can make by writing an article. I thought about a program I could do and I came up with the idea of a program about codes.

My program is a game called "Code Guesser." I worked on it with our TRS-80; it didn't take very long because my dad helped me. It took only two days. I typed it into the TRS-80 and saved it on a floppy disk. A floppy disk is a square thing you save your program on. It stores a lot of information like a tape, but the computer makes the floppy go faster than tape. My dad gave my brother and I each a floppy to use.

Photo 1. Jodi at the Computer

Writing the Program

To get me started, my dad wrote a program that shows all the programs on the floppy when you first turn on the computer. It keeps the programs in a list. I added the name Code Guesser to the list.

As I was working I saw how much this program grew from my other programs about my family. The other programs about my family were simple programs—just Print commands.

The commands I learned to use are: CMD"S", Save, Print, Edit, Run, List, CLS, and Input. CMD"S" directs the computer so you can use DIR to see everything on the floppy disk. Save means the computer will put whatever you type in, like SAVE "CODES" (name of your program), on the floppy. Then it will save it. Print means put it on the screen so whoever is running the program can see it. I used Print statements to give instructions to whoever is running my code program. Edit is used for making changes. The command List shows you the program so you can see it in case something goes wrong or you want to change something in the program. It also means print the program on the screen. CLS means clear the screen. Input makes the computer wait so you can type what it tells you to. I used Print statements to give instructions to the user of my code program.

I thought about how I was going to write my program. Then I started to type it in so the computer could understand and do it. When I started a list for the computer to understand, I typed commands and how to run it. I thought of some codes and wrote them down and worked on them. I thought

Reader Service for facing page ✓357→


The Galaxy Invaders Have
Returned in This Newest Game
of Skill and Excitement.
DEFENSE
COMMAND
BIG FIVE
SOFTWARE

"My brother was making a program too, so we set a time limit. . ."

of some sentences I could have be my questions in the game. I typed the codes and questions into the program. I got quite a bit of help from my dad. He thought some things were too hard to explain to me so he just typed them in.

I think it is a pretty good program if you like to solve problems and figure puzzles out.

Program Listing 1. Jodi's Game, Code Guesser

```
100 CLEAR(512)
110 CLS
120 PRINT"THIS IS A GAME WHERE YOU TRY TO GUESS A CODE THEN YOU WILL TYPE IT IN
USING NORMAL LETTERS DO NOT!!!!!!!ANSWER THE QUESTION
130 PRINT"THERE WILL BE 3  KINDS OF CODES
140 PRINT"I WILL GIVE YOU 1 OF THOSE CODES
150 PRINT"THE CODES WILL BE DIFFERENT
160 PRINT"THE CODES WILL FLASH ON TOP OF THE SCREEN.YOU CAN WRITE THEM DOWN OR T
RY TO REMEMBER THEM
170 PRINT"I WILL ASK YOU YOUR NAME
180 PRINT"GOOD LUCK!!!!!!!!!!!!
190 PRINT"WHAT IS YOUR NAME?
200 INPUT N$
210 CLS
220 PRINT N$
230 PRINT"THE CODES WILL FLASH ON TOP OF THE SCREEN THEN I WILL ERASE THEM."
240 PRINT"THIS IS THE CODE
250 PRINT"A B C D E F G H I J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X"
260 PRINT"1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24"
270 PRINT
280 PRINT"Y  Z"
290 PRINT"25 26"
300 PRINT
310 PRINT"THIS IS THE SECOND CODE
320 PRINT"A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R S T U V"
330 PRINT"26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5"
333 PRINT
334 PRINT"W X Y Z"
335 PRINT"4 3 2 1"
350 PRINT"PUSH ENTER FOR MORE CODES";
360 INPUT A$
370 CLS
380 PRINT"THIS IS YOUR THIRD CODE
390 PRINT"A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U"
400 PRINT"1  26  2  25 3 24  4  23 5  22 6  21 7  20 8  19 9  18 10 17 11"
410 PRINT
420 PRINT"V  W  X  Y  Z"
430 PRINT"16 12 15 13 14"
440 PRINT
450 'PRINT
460 'PRINT"THIS IS YOUR FOURTH CODE
470 'PRINT"CROSS OUT THE LETTERSIN ORDER C-O-D-E!THEY HAVE TO BE IN ORDER OR ELS
E IT WONT WORK
480 'PRINT
490 PRINT"PUSH ENTER FOR YOUR QUESTION
500 INPUTQ$
510 CLS
520 PRINT"HERE IS THE QUESTION
530 M=1
540 READQ$:IFQ$<>"END"THENM=M+1:GOTO540
550 RESTORE:N=RND(M-1)
560 FORI=1TON:READQ$:NEXTI:Q1$=Q$
570 Z=INSTR(Q$," ")
580 IFZ>0THENQ$=LEFT$(Q$,Z-1)+RIGHT$(Q$,LEN(Q$)-Z):GOTO570
590 'PRINT Q$
600 A=RND(3)
610 ON A GOSUB 710  ,810  ,840
620 PRINTC$
630 PRINT"WHAT DO YOU THINK THE QUESTION IS?
640 INPUT A$
645 IF A$<>Q1$ THEN PRINT"WRONG. TRY AGAIN":GOTO630
650 PRINT"YOU GUESSED THE CODE. WOULD YOU LIKE TO TRY AGAIN";
660 DATA HOW OLD ARE YOU
662 INPUT A$:IFA$="YES"THEN490
663 RUN"MENUE"
670 DATA WHAT DO YOU THINK OF COMPUTERS
680 DATA DO YOU SKI
690 DATA WHERE DO YOU WANT TO GO FOR VACATION
700 DATA WHAT IS YOUR NAME
710 REM
720 B$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
730 C$=""
740 FORI=1TOLEN(Q$)
```

Program continues

How it Works

Here is how it works: The instructions aren't too hard to follow. The computer will flash all the codes on the screen. Then it will give you a coded sentence and you have to figure it out. There are three different codes and five different messages. The computer will say if your guesses are wrong or right.

I had the computer print what happened if the guess is wrong, and if it is wrong, to print "You are wrong, you can try again." If they got it right, it prints "Correct. Would you like to try again?" If the answer is yes, the game starts over. If no, the computer runs the Menu.

I have a listing of the program. Here is an explanation of what each part is: At the beginning my dad put in line 100. Lines 110 to 180 put the directions on the screen. Lines 190 to 200 ask for your name. Line 210 clears the screen. Lines 220 to 350 print examples. From line 360 to line 490 the computer prints your third code. Lines 500 to 510 stop the game and clear the screen. My Dad did lines 520 to 600, about how to pick the code. Lines 610 to 620 pick the code. In 630 to 663 the program asks you to type in your answer and then checks it. Lines 670 to 700 contain the questions the computer makes intc codes. From line 710 to 1170 is the part my dad wrote that helps the computer put the sentences into codes.

After Effect

I had a few problems. My brother was making a program too, so we set a time limit for each of us and then worked on our programs (one at a time). It was the only fair way. Another problem was that I had a frustrating time because I just learned the keyboard. It wasn't really a big problem because once I got started I remembered the keys.

I had fun showing the game to my friends. When my friends came over to see my programs on my family (the first programs I did, mentioned earlier), they thought they were dumb, so I wanted them to enjoy something I wrote on the computer. I noticed my friends had more fun playing my game than seeing the earlier programs. Playing the code game is fun for them although it isn't much fun for me because I know the answers.

Now it is done I know a lot about computers, like how to print things on the screen; how to edit, run, save, input and how to change the line with edit. ■

Jodi Tallman is 9½ years old, and is learning to program computers. This is her first published article.

Hayes Stack™

Microcomputer Component Systems

A beautiful match, the Smartmodem and the TRS-80. *Your TRS-80 can talk with other computers,* over the telephone lines. And with no acoustic losses or distortions.
Access time-sharing systems and information utilities such as the Source,* CompuServe† and MicroNet.
Direct hook-up with no interference noises. The Smartmodem hooks to the telephone line just like a modular telephone, simply insert in a wall jack.
"Love at first sight" – your TRS-80 and the Smartmodem!

Brawny – because it does so many things. *Auto-dial and auto-answer features built in.* With the Smartmodem, your TRS-80 can *automatically* dial the telephone, answer the telephone, receive and transmit, and hang up the telephone. Completely unattended.
*Pulse dialing or Touch-Tone.*** The Smartmodem can be connected to any telephone system in the U.S. because it allows pulse-dialing, Touch-Tone dialing or a combination of the two. FCC approved.
Program controllable in any language using ASCII character strings. This is a unique feature of the Hayes Smartmodem.

Brainy – because it does them all so simply. *Seven LED indicators on the front panel* give you visual signals of the status of the Smartmodem: MR – Modem Ready, SD – Send Data, CD – Carrier Detected, etc.
The audio monitor feature lets you "listen in" on the call being dialed and the connection made. You are immediately alerted to busy signals, wrong numbers, etc.
Over 30 different commands can be entered directly from your TRS-80 keyboard, including the unique "Set" commands which allow you to select and change various optional parameters such as dialing speed, escape code character, length of time for a dial tone, and number of rings to answer. There are 17 "Set" commands.
The Smartmodem is completely compatible with the Bell-103 type modems, the type of modem most time-sharing systems have. Operation can be in full or half-duplex, with a transmission speed of 0-300 baud.

The Smartmodem is ready to "get-together" with your TRS-80. TRS-80 Model II and TRS-80 Color Computers have RS-232 serial ports and can immediately interface with the Smartmodem. Expansions that permit use of the Smartmodem with TRS-80 Model I and Model III are available through your TRS-80 dealer.

Match your TRS-80 with a Hayes Smartmodem for a sophisticated, high performanced data communication system. Available at computer stores nationwide (except TRS-80 dealers) – call or write for the location nearest you. And don't settle for anything less than Hayes.

Hayes

Hayes Microcomputer Products, Inc.
5835 Peachtree Corners East,
Norcross, Georgia 30092 (404) 449-8791

Your TRS-80® computer and the Hayes Stack™ Smartmodem. Beauty, Brains, and Brawn!

™Hayes Stack is a trademark of Hayes Microcomputer Products, Inc. *Trademark of Source Telecommunication Corporation. **Trademark of American Telephone and Telegraph. †Trademark of CompuServe Information Services, Inc. TRS 80 is a registered trademark of Tandy Corp. © Hayes Microcomputer Products, Inc. Sold only in the U.S.A.

Program continued

```
750 D$=MID$(Q$,I,1)
760 E=INSTR(B$,D$)
770 D$=STR$(E)
780 C$=C$+D$+" "
790 NEXTI
800 RETURN
810 REM
820 B$="ZYXWVUTSRQPONMLKJIHGFEDCBA"
830 GOTO730
840 REM
850 C$=""
860 FORI=1TOLEN(Q$)
870 D$=MID$(Q$,I,1)
880 IFD$="A"THENE$="1"
890 IFD$="B"THENE$="26"
900 IFD$="C"THENE$="2"
910 IFD$="D"THENE$="25"
920 IFD$="E"THENE$="3"
930 IFD$="F"THENE$="24"
940 IFD$="G"THENE$="4"
950 IFD$="H"THENE$="23"
960 IFD$="I"THENE$="5"
970 IFD$="J"THENE$="22"
980 IFD$="K"THENE$="6"
990 IFD$="L"THENE$="21"
1000 IFD$="M"THENE$="7"
1010 IFD$="N"THENE$="20"
1020 IFD$="O"THENE$="8"
1030 IFD$="P"THENE$="19"
1040 IFD$="Q"THENE$="9"
1050 IFD$="R"THENE$="18"
1060 IFD$="S"THENE$="10"
1070 IFD$="T"THENE$="17"
1080 IFD$="U"THENE$="11"
1090 IFD$="V"THENE$="16"
1100 IFD$="W"THENE$="12"
1110 IFD$="X"THENE$="15"
1120 IFD$="Y"THENE$="13"
1130 IFD$="Z"THENE$="14"
1140 C$=C$+E$+" "
1150 NEXT I
1160 RETURN
1170 DATA END
```

How I Taught My Children

by Jim Tallman

I decided to acquaint my children to the TRS-80 in a way that wouldn't overwhelm and confuse them. I wanted to avoid the need to deal initially with the concepts of directories, DOS, and Basic. I chose to show them a few basic procedures, but to insulate them from the things they didn't need to know yet, letting their need and questions guide the rate of growth. Also, I decided to avoid sitting beside them all the time they were working, to allow them to proceed at their own speed, to experiment, and to use their imaginations. I was concerned that if I watched over them continually, they would feel I was imposing pressure on them. I let them make their own decisions as often as I could.

To get them started with the computer, I first showed them, with little explanation, how to turn it on. To start the program, I used the Auto command in NEWDOS to get the TRS-80 to go to Basic and load their program. I later added a menu driven program, so they might select between a variety of programs they had written. Jodi refers to that in her manuscript.

Launching the Exploration

We began, therefore, with one of the simplest and most useful commands, the simple version of Print. I chose this command because it's useful and easy to understand and use. This decision was reinforced by its selection by the author of the Level I manual as his starting point, also.

Using only Print statements allowed the children to learn and exercise some important concepts about the computer. For example, Basic's way of using the numbers to identify each line and using line numbers to control where additional lines are to be added within those already entered. We needed to remove some lines in the children's programs, so I explained how to do that; just type the line number with no Print command to erase it. Correcting an error in a line required retyping the whole line over. I postponed explaining how to use the Edit capabilities of the TRS-80 for the time being.

The children learned some control concepts. They grasped the concept of this hidden thing which controls the computer's operation called a program. They could create it and interact with it. They understood the idea of Listing a program to see it, and Running it to begin its operation.

My two children took alternating turns on the computer. To maintain the insulation from system concepts, I would save their programs and restart the other child's at each shift change. I quickly tired of this and they learned the Save command.

It was FUN

This level of interaction with the computer was exciting because the children were typing on the keyboard (which they always want to do on our typewriter), but also, they were having fun making the computer respond, understanding, and controlling this mysterious thing, the computer. Their interest was sustained by their early success at having the computer respond to their wishes. Using the Print command quickly led to questions about how to start the program, and allowed them to expand their understanding. They were quite receptive to new commands when they had a need, and not so receptive when they couldn't identify the applicability.

There were some concepts I judged too hard to teach them at this early stage. Rather than explain the decoding and encoding algorithms for Jodi, I decided to program them for her. She selected the coding patterns, using me to try out ideas, but I programmed the algorithm. I wasn't sure she could grasp the mathematics and even if she could, explaining them could distract her from the main activity, getting acquainted with the computer. She accepted that readily.

What Was Successful

I feel comfortable with the approach I took regarding the rate of supplying new information about new commands. The children had some acquaintance with the computer from playing games on it, so they knew it could ask questions and use their responses. When they encountered a need for that, they asked about it. I then explained the Input command. I feel this approach was successful.

The children learned some programming commands and concepts, like Listing, Running, and Saving. Later they learned about the floppy disk.

Both children had trouble with a few concepts, which they eventually mastered. They both confused the letter O with the number 0. This was a problem during line numbering and using the GOTO command. Naturally, most of us draw the two to look the same, but they are quite different to the computer. The children also had lots of trouble with the crazy keyboard we have all grown to accept. They found the needed keys eventually and became used to it.

Not All Worked Out as Planned

I tried numerous times to get Jodi and her brother to read the manual and teach themselves. I never was sure why, but they didn't respond to the manual. They much preferred to have me get them started and consistently persuaded me to provide the desired information. I even opened the book to the page with the relevant information for them, to no avail. Perhaps they knew I knew the answer and would give them individual attention. In any event, I regularly succumbed to their persistence in querying me rather than see them get discouraged.

Dad had fun doing this too! ■

Program Listing 2. Jodi's Dad's Menu Program

```
50 REM      MENU
100 'AUTO LOAD PROGRAM FOR EASIER USE
110 ONERRORGOTO1000
120 DIM NA$(40),PROG$(40)
140 J=1:CLS:PRINTCHR$(23)
160 PRINT@0,"NUMBER"TAB(14)"NAME":PRINT
200 RESTORE
220 READ NA$(J),PROG$(J):IF NA$(J)="END" THEN 320
260 IFJ<10THENPRINTTAB(1)J;"..... ";NA$(J)
280 IFJ=>10THENPRINTJ;"..... ";NA$(J)
300 J=J+1:GOTO220
320 PRINT@904,"TAP YOUR SELECTION  ";
340 I$=INKEY$:IFI$=""THEN340ELSEPRINTI$
360 I=VAL(I$)
380 IF I>=JTHENPRINT@960,"THE NUMBER MUST BE";J-1;"OR SMALLER";:GOTO320
400 IFI<1THENPRINT@960,"THE NUMBER MUST BE 1 OR LARGER ";:GOTO320
420 CLS:PRINTCHR$(23):PRINT@76,NA$(I):LOAD PROG$(I),R
500 DATA MY RELATIVES,PEOPLE
520 DATA ABOUT MY MOMMY,MOMMY
540 DATA ABOUT MY DADDY,DAD
550 DATA ABOUT MY BROTHER,STEVE
560 DATA ALL ABOUT ME,JODI
570 DATA MY FRIENDS ARE...,FRIENDS
580 DATA CODE GUESSER,CODES
590 DATA ELIZA,ELIZA
700 DATA END,END
1000 REM ERROR HANDLER
1020 IF(ERR/2)<>53THEN1200
1040 PRINT:PRINT" I CAN NOT FIND THAT PROGRAM":PRINT:PRINT" TAP ANY KEY TO MAKE
ANOTHER                SELECTION"
1080 A$=INKEY$:IFA$=""THEN1080
1200 RESUME 140
```

Program continues

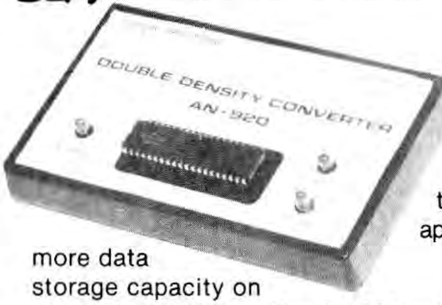
HARDWARE BREAKTHROUGH

DESIGN SOLUTION INC. presents the AN-SERIES DEVELOPMENT SYSTEM
for TRS-80™, Apple II™, Commodore PET™, Superbrain™*

$99.95 DISK CONTROLLER

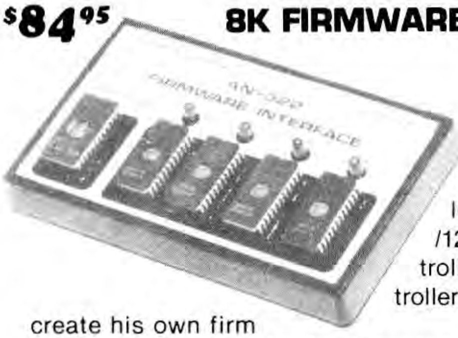
An affordable multidrive floppy disk interface. The AN-760 supports 35, 40, 80 and 160 track drives. Using proprietary Digital Data Separation Techniques, maximum permissible data transfer integrity is assured. Read, write, and step LED indicators prompt operator during all disk I/O. The AN-760 comes complete with power supply module and operation manual. (Operating system software optional.)

$129.95 DOUBLE DENSITY CONVERTER



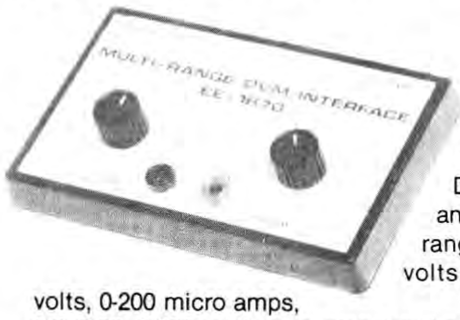
Using DSI Proprietary Data Separation and Write Precompensation Techniques, the AN-920 provides approximately 1.6 times more data storage capacity on your 5¼ disk drives. The AN-920 can be used in conjunction with the AN-760 FDC or the TRS-80 expansion interface. Double Density System comes complete with power supply module and operation manual. (Required Newdos-80 Ver 2.0 Disk Operation System Optional.)

$84.95 8K FIRMWARE INTERFACE



8K Bytes of User Programable Firmware (4-2716 Eproms) are automatically loaded by system /12345 Command controlled by a 2716 controller chip. The user can create his own firm operating system to load and execute programs from Eprom or to provide user defined arithmetic functions. Comes complete with power supply, operation manual and controller chip. (The AN-522 requires AN-551 Eprom programmer.)

$99.95 MULTI-RANGE DVM INTERFACE

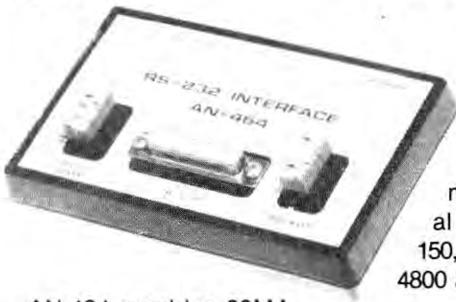


Analog Signal Interface is now as easy as turning a knob. EE-1670 provides DVM functions volts, and milli-amps and 8 ranges, 1-200 millivolts, 0-2, 0-20, 0-200 volts, 0-200 micro amps, 0-2, 0-20, 0-200 milliamps, standard banana jacks and test leads create perfect connection scheme for real time analog data acquisition. EE-1670 system is complete with power supply and manual of operation.

$99.95 EPROM PROGRAMMER

An enhanced version of our original Eprom Programmer, the AN-551 will now program the single supply 2516, 2716, 2532 and 2732 Eproms from Basic or machine language. Software provided will load Eprom from TRS-80 Ram or load TRS-80 Ram from Eprom with complete on-screen verification. The AN-551 comes complete with power supply and operation manual.

$89.95 RS-232 INTERFACE



Can be configured to communicate with data communications equipment or data terminal equipment at 110, 150, 300, 600, 1200, 2400, 4800 and 9600 Baud. The AN-464 provides 20MA loop output and RS-232C through a DB-25P connector. Software selectable odd/even parity, 5-8 bit word lengths and stop bit formats are standard along with power supply module, software driver, and operation manual.

All connections made to AN-SERIES products from your breadboard are simply pushed through the front panel. Custom connectors on the P.C.B. provide super reliable connection for thousands of operations. All AN-SERIES products are warranted for a full 90-DAYS under DSI's limited warranty policy. Complete documentation is provided for each model in an attractive folder, including theory of operation, and special interest projects and applications. All units are supplied with external power supply modules that connect through a mini-jack on the front panel.

DESIGN SOLUTION INC.
BOX 1225, FAYETTEVILLE, AR 72701

C.O.D. add $5.00 — Other add $3.00 S/H.
(Exports slightly higher)
Orders and Information: (501) 521-0281

✓123

A story on inner-city computing and a teacher who cared.

An '80 in the Apple

Stephen Radin
751 Bard Ave
Staten Island, NY 10310

I began teaching children to program in grades six through eight in 1971 when it became apparent that a supplement to our regular curriculum was needed for our most gifted youngsters. At the time, I was teaching and supervising in a New York City Title I (poverty level) school, it was very unlikely that computers, which then cost more than $100,000 apiece, were a viable possibility.

The youngsters were very bright. Their competitive spirit was often tested and usually ranked among the top in New York City despite the fact that the students' average reading and math abilities were far below those of many other local schools. I designed a course for them which included two years of advanced study (some of our eighth graders completed both the ninth and eleventh year New York State mathematics curriculum before they left us). We also provided them with enrichment studies in subjects such as: symbolic logic, special relativity, and statistics to name just a few.

I felt that we should try programming. Not only did it appear that it would provide a useful skill for their future lives but it would keep them stimulated and occupied.

I met with my most persistent students after school one day with a programmable calculator which I was able to borrow from a local store. I taught them the rudiments of programming on this machine. A few months later we were able to get a similar machine of our own.

My students soon mastered the basics of a calculator and wanted more. I was interested in introducing them to the power of a computer and began to look around. Surely we could never own one but maybe someone would let us use theirs.

Developing Slowly

The College of Staten Island had an IBM 1130 machine, and after some convincing they were kind enough to allow us to use their facility during periods when their machine was underutilized. I could now teach my students to program. I began with PL/1 (a language) and programs which were used by the college for their introductory course. The subjects included array handling, string manipulation and some fundamental arithmetic manipulation.

It took some adjustment at first, but within a few months my students were programming the Fibonocci sequence, grade averaging programs, file management work and the like. The course developed over the next few years. I trained two of my colleagues and they helped me to transport the students and to teach them. We were able to develop a beginning and an advanced level course.

This went on for five years until Tandy released their first microcomputer. Now a computer for one fiftieth of the former cost was available. We had to have one! The machine could help with both our enrichment problem as well as our serious skills retardation problem.

Getting the Funds

Unable to raise the $1,000 for a machine from either our school P.T.A. (Title I parents have very little money), or our local community, we searched for other sources of funds. Our college friends came to the rescue. They advised us to write proposals to various governmental agencies (city, state and federal), applying for funds. I wrote to governmental agencies both locally and in Washington. I also applied to many private foundations for the necessary funds and crossed my fingers. My fingers got very tired that year, but it paid off. A grant of $2,200 was forthcoming from our state government which allowed us to buy a TRS-80 microcomputer, a lot of paper, and supplies.

A little thought made it clear that if my students could program the machine for their remedial classmates who needed arithmetic and English vocabulary practice we would be able to kill two birds with one computer.

Software Needs

I wrote the first few programs which included an interactive Computer Aided Instruction program in mathematics, as well as some file management programs for maintenance of our class records. The programs covered the fundamental skills of whole number and decimal arithmetic, as well as fraction and percent work and many word problem types. It became clear that the remedial students were very attracted to the new format of instruction because they saw the screen as an extension of their home television set. We had tapped the TV generation syndrome and found a gold mine. Many teachers told me that the children who came out of the lab were much more relaxed and orderly the following period than at any other time of the week.

Curriculum supplementation was possible in fields other than math. We began taking advantage of the equipment to support other areas such as music, art and typing.

Student Power

Students soon began to write all the 'in house' programs. Their free use of graphic cartoon characters and sound were excellent, making my earlier efforts pale by comparison. Student power became a useful means to our end.

Over the next few years we were able to get four more microcomputers and two LA36 Decwriters. The Decwriters allow us to communicate with the three IBM 370 computers at the City University. This has allowed us to expand our program to include an advanced class as well and have some children work in other languages not available on our microcomputer.

Our biggest problems seemed to be getting the staff and children away from the machines at outrageous times of the day. This past year more than two classes full of children and eight of our teachers were involved with programming, and ten of our remedial classes were involved in remedial practice. Many teachers see computer programming as a way to enliven their classwork, while others have expressed an interest in computing as a second career.

We have reaped the benefits of this energy in the many programs which have been written in the areas of language arts, mathematics, modern languages and art, as well as the many spinoffs such as school scheduling, treasury, data file maintenance, music composition and test readiness.

Our primary source of funds has been through the receipt of grants from governmental agencies. Many of our colleagues in other schools have been successful in obtaining help from private industry and even local department stores. Still others wrote to private foundations for funds. There is money available for public, private and parochial schools who are interested in using computers as an educational aid. Just keep your fingers crossed and you will succeed. ■

LOAD 80

The Key Box

Basic Level II
Model I
16K RAM
line printer required

Learn to add special graphic routines to your educational programs.

Computer Etch-A-Sketch

Thomas W. Mustico
RD #2 Route 104A
Sterling, NY 13156

Educational programming on the TRS-80 presents some very special challenges when graphics are used for reward or for illustration. Though there have been several excellent articles demonstrating the use of the Set, POKE and CHR$ commands, the typical programming teacher usually finds incorporating graphics into a learning program a difficult task. This is because most graphics are too slow, either in execution or in initialization.

Most of us are envious of the speed produced by machine code or by packed strings that allow fast action games that capture the undivided attention of students. For teachers, however, extensive use of machine code is out of the question. Also, many teachers find translating a graphic idea into CHR$ codes and producing concatenated strings not only very hard work but very expensive in terms of memory use

After months of wondering how to accomplish effective graphics with reasonable expenditure of programming time, I found the solution by mixing a number of ideas from William Barden Jr. *(Programming Techniques For Level II Basic)*, Dan Keen ("Kid Stuff," September, 1980), David Grimes ("Stringy Machine Code," September, 1980), and Valerie Vann ("Season's Greetings," December, 1980).

My solution is called GSETTR6/BAS (pronounced G SETTER 6) which combines the notion of a machine etch-a-sketch and packed strings through the use of the POKE and VARPTR commands. Once the graphic has been accomplished on the screen, GSETTR6/BAS will produce a self-modifying program that produces graphics at high speeds and yet minimizes both memory requirements and initialization time. With a small amount of coaxing you may also inter-mix characters with the graphic codes.

Once GSETTR6/BAS has written the graphics program, producing the finished package can be approached either by merging or adding additional code which makes use of the graphics. The only problem is that GSETTR6/BAS can only be printed before it is run the first time; that is, before it modifies itself.

Program Requirements

GSETTR6/BAS was designed for a 32K TRSDOS 2.3 configuration but has been altered so that it will work on a 16K Level II Model I if the system has a line printer. Under the DOS version the generated program is stored in ASCII format on the disk and can be run directly. The DOS version is preferable because it eliminates the necessity of typing the generated program back into the machine.

Generating the Graphics Program

When GSETTR6/BAS is run it produces the option of reviewing the directions, assigning a program name, or printing the generated program. In general, the production of graphics is only a matter of moving a flashing cursor to a desired position on the screen and then setting or resetting the position. The cursor position is moved by the four arrow keys. After the flashing cursor has been positioned, the pixel can be set by hitting S or reset by hitting R. Holding any arrow key and S or R simultaneously allows you to set or reset continuously until one or the other of the two keys is released. (Sorry, but you cannot achieve diagonal motion by simultaneously holding two arrow keys.)

Once the graphics are produced on the screen, shift G causes GSETTR6/BAS to write the generated program to disk, printer, or both. The generated program will contain a translation of the screen stored in data statements and a section which will POKE the data into an established dummy string array of minimum length. Using space compression and line control codes helps to minimize space requirements. The third section of the program will delete all data statements, the POKEing section and the Delete instruction itself, when the program is run.

All that remains at this point is a minimum program containing the graphics in one or more elements of a string array named DG$(i), and a loop which will print it. If you are not working with a disk system, be sure to save the generated program to tape before entering Run the first time (just in case one of Murphy's Laws is operative).

GSETTR6/BAS Commands

Of course, you realize that nothing is as simple as it first appears, so there are a

"The typical teacher finds incorporating graphics into a learning program a difficult task."

few more commands at your disposal when using GSETTR6/BAS. Only two of them, however, are absolutely necessary for success.

When the translation of the graphics on the screen takes place, the program requires a marker for the upper left and lower right corners of a rectangle as the definition of the area to be translated. These markers are inserted by positioning the cursor in these two positions and entering a B(egin) and an E(nd) at the upper left and lower right positions, respectively. Thus, the minimum set of commands for GSETTR6/BAS are S, R, B, E, shift G and the arrows.

The H command is an easy and convenient way to get the cursor out of the way. Hitting H will return the cursor to the H(ome) position, the upper left corner of the video screen.

F is also a cursor-oriented command which is used to control the number of set positions that the cursor will move in response to hitting an arrow key. After F, hit any number from one to nine and the cursor will jump into the F(ast) mode. The F command can be used to create patterns for every other or every third, etc., position.

The V command is for planning ahead when you may want to pinpoint certain coordinates during the use of the graphics. Hitting V will cause the Set coordinates for (X,Y) to be printed in the lower left hand corner of the video screen.

The use of X causes the program to scan the video screen over the entire rectangle defined by the B and E positions and to store the necessary CHR$ codes to reproduce the graphics on the video screen.

The D and shift D commands cause the stored graphics to be displayed on the screen. The difference between the two is that the D will clear the screen before displaying the graphics, but the shift D will not. Both D commands form the display graphics one character at a time by POKEing to video memory. (The D, shift D, L, and P commands must be preceded by the use of an X.)

The List command will form the graphics as a set of concatenated packed strings and list these strings to a printer, if one is connected. If no printer is connected, the program will simply form the strings necessary to produce the graphics. Once the strings have been formed, the P command can be used to print the graphics to the screen at high speed. The final command of this group, M, will multiply the graphics by printing it such that the upper left corner of the block is located at the position of the cursor. M will not clear the screen before printing (or over-printing as the case may be). The M command will not be of much use for multiplying the graphics in the horizontal, however, if the Begin position is the first location of any video print line. Why? Because in this case, GSETTR6/BAS will use internal codes to move to the beginning of the line.

In the process of putting GSETTR6/BAS through its paces to eliminate any bugs, I discovered that it would be advantageous to allow some form of lettering to be part of the graphics. Thus came the Q command which will calculate the cursor's nearest printable location and print keyboard characters to the screen until Enter is pressed. The Q command will not allow a quotation mark (") to be entered because this would cause an unintended termination of the string built by the program. Lettering should probably be postponed until the graphics are finished because the cursor cannot pass through the lettering as it does the graphics. The Q command is terminated by pressing Enter.

If you are using the Q command with the Model I which contains the upper/lowercase modification, you should activate the driver before using GSETTR6/BAS. When the driver is inactive the internal codes in video memory turn out to be 64 less than you expect. If you PEEK at a location containing an A, the result is one instead of the expected 65. I felt it simpler not to load ULCDVR, so I added this line of code to the program: 535 IF PEEK(IP)>0 AND PEEK(IP)<27 THEN POKE IP,PEEK (IP) + 64.

My Model I is one of the early versions where the screen distorts badly when whited out; but for those of you with later and more stable video screens the shift W will white the screen and allow you to use the R command to produce a graphic which is black on white. Even earlier versions will look okay if there is enough black used in the areas where the screen tends to bend.

Now that you have read about all of your options the last may turn out to be the best. The shift H command is recognized by the program as a request for help. This command will automatically log the graphic and then produce a list of directions. When you hit Enter to continue, the program will redraw the graphic and allow you to continue. Even if you have specified a B(egin) and E(nd), the program will log the entire screen. The program will then reset your original B and E.

How GSETTR6/BAS Makes Decisions

GSETTR6/BAS will allow the choice of sending the generated program to disk, printer or both. It will PEEK location 14305 to find out if it is in a disk operating system and, if not, will automatically send the generated program to the printer. This action will be taken no matter what the response is to the print option (after all if you don't send something to either disk or printer, you must have loaded the wrong program).

Because the generated program must have a name to be stored on disk, GSETTR6/BAS will present an opportunity for you to enter the name of your choice. If you fail to do so, and a disk write operation is performed, you will find the generated program on the disk under the name DUMMY/TXT. This feature will keep a lot of ASCII coded files from resulting in a Disk Full message at the wrong time.

When the program is running, the printer status is continuously examined for a ready state by PEEKing 14312. If a L(ist) command is given and the printer is out for repairs, off or reset, the program will still perform the function of generating concatenated strings even though they are not directed to the printer. These strings are the bases for the use of the high speed prints to video by the P(rint) and M(ultiply) commands.

GSETTR6/BAS is willing to accept a new command at any time when the cursor is flashing. If the cursor is not flashing, then the program is either busy or waiting for input.

Some Notes for the Model III

Because GSETTR6/BAS uses no machine ROM routines and both the video and keyboard memories are in the same locations in the Model I and Model III, the program should work on either machine. (I'm still waiting for my Model III DOS manual, so I can't promise some adjustments won't be required.)

I have loaded GSETTR6/BAS and generated graphics on a Model III disk system and it worked without a hitch. Of special interest is that the Model III will actually List the graphics rather than a bunch of translated tokens that drive the video crazy. One thing to note on the Model III is lowercase—GSETTR6/BAS expects com-

"There are a few more commands at your disposal. . . only two of them, however, are necessary for success."

mands to be from an uppercase keyboard. Remember, that means a shift G is a lowercase letter.

Features of the Generated Program

Programs generated by GSETTR6/BAS will contain statements to reduce the clear space back to 50. They will also define I–N as integers, and X–Z as strings. The dummy string array will be set up starting at line number 50. Data statements which contain CHR$ codes will begin at line number 205 and the program segment to POKE the CHR$ codes into the dummy string array will begin at line number 500. A short program segment to print the graphics will begin at line number 905.

When the generated program is run for the first time, the information contained in the data statements is POKEd into the dummy string array by the program segment beginning at line number 500. After this operation is complete, the statement at line 900 will delete all code from lines 205 to 900. At this point you will have a Ready message on the screen. Entering Run will then produce your new graphics with amazing speed.

You may now store this short version on tape or disk; or begin immediately to add new statements to manipulate the graphics. If you are using TRSDOS 2.3, please note that you cannot use the Merge command to link two graphics files containing packed strings. While it is possible to edit other lines in the program, you can not edit the lines containing the packed strings (well you can—but they won't work if you do). It is possible to renumber the program without causing errors.

If you are determined to merge two GSETTR6/BAS programs here are some hints. Let me assume you have two programs that are named A and B. Load program B and change all references to DG$(i) to some other variable name such as BG$(i); Delete the Clear, DEFINT, and Dimension lines; and then renumber the program so it starts with line 1000. Now save the version (note that it has not been run before saving). Next load and run program A, delete 950, and merge program B.

If you are using TRSDOS 2.3 remember that program B must be saved with an A option in order to successfully use the Merge command.

Avoid Typing Long Programs

If you have examined the accompanying program listing you may have noticed the heading contains a reference to CLEAR-80 (College Library for the Exchange of Educational Applications of the Radio Shack TRS-80). For those of you out there who send either a tape or a disk (containing TRSDOS 2.3) with a proper self addressed-postage paid return, I will send back the recorded program.

What's the hitch? Well, I hope that when you produce an educational program which you allow to be freely exchanged that you would send it along to CLEAR-80. It would be appreciated by teachers who are trying to find good programs to use in educational environments at the right price (free). ■

Program Listing

```
10 CLS
20 PRINT"GSETTR6/BAS  DOS 2.3 32k OR
             LEVEL II 16K WITH PRINTER
"
30 PRINT"BY T. W. MUSTICO
STATE UNIVERSITY COLLEGE AT OSWEGO
"
40 PRINT"PROGRAM DISTRIBUTED THROUGH
OSWEGO CLEAR-80
STATE UNIVERSITY COLLEGE AT OSWEGO
110 POUCHER HALL
OSWEGO NY, 13126
"
50 INPUT"PRESS ENTER TO CONTINUE";A$:CLS
60 CLEAR 3000
70 DEFINT I-N
80 DIM IG(1025),GG$(12)
90 CLS:INPUT "DO YOU NEED DIRECTIONS";D$:IFLEFT$(D$,1)="Y"THEN G
OSUB1160 :CLS ELSE CLS
```

Program continues

✓See List of Advertisers on page 290

Program continued

```
100 INPUT "TO PRINT PROGRAM UNDER <SHIFT> <G> OPTION ENTER <1>";
MP
110 IF LLIST(14305)=255 THEN MP=1
120 GN$="":INPUT "INPUT NAME TO SAVE FILE OF GRAPHIC TO DISK
DEFAULT NAME WILL BE DUMMY/TXT";GN$
130 IF GN$=""GN$="DUMMY/TXT"
140 CLS
150 'TO TRANSLATE ANY GRAPHIC - INSERT AT 9000 AS A SOUBROUTINE
160 'GOSUB9000
170 A$="":IX=0:IY=0:J=1
228 LC=POINT(IX,IY):JX=IX:JY=IY
190 JA=PEEK(14400):KA=PEEK(14340):IFJA<>0  GOTO1300
200 IFA$="D" GOSUB 600  :GOTO170
210 IFA$="d"GOSUB610  :GOTO170
220 IFA$="Q" GOSUB1410 :GOTO170
230 IFA$="S" THEN SET(IX,IY):GOTO420
240 IFA$="R" THEN RESET(IX,IY):GOTO420
250 IF A$="H" THEN GOTO170
255 IFA$="h"THEN JB=IB:JE=IE:GOSUB1030:GOSUB510:GOSUB 1160:GOSUB
600:IB=JB:IE=JE:GOTO170
260 IF A$="F" GOSUB490
270 IF A$="B" GOSUB980   :IB=LO:NX=IX:NY=IY
280 IF A$="E" GOSUB980   :IE=LO:MX=IX:Y=IY
290 IF A$="L" G$="":LX=0:GOSUB660  :GOTO170
300 IF A$="X" THEN GOSUB1030 :GOSUB510  :GOTO170
310 IF A$="P" GOSUB1140 :GOTO170
320 IFA$="w" THEN GOSUB 1120
330 IFA$="g" THENGOSUB 1030 :GOSUB500  :GOSUB 1470 :GOTO120
340 IFA$="M" THEN GOSUB 970  :GOSUB1140 :GOTO 180
350 IFA$="V"PRINT @960, IX;,IY;
360 GOTO 450
370 IF L =-1 THEN SET(JX,JY) ELSE RESET(JX,JY)
380 IF IX>127 THEN IY=IY+1:IX=0
390 IF IX<0   THEN IY=IY-1:IX=127
400 IF IY<0  THEN IY=0
410 IF IY>47 THEN IY=47
500 NC=POINT(IX,IY)
430 SET(IX,IY):RESET(IX,IY)
440 IF IR=1 RETURN
450 A$=INKEY$: IF A$="" GOTO430
460 IF NC=-1 THEN SET(IX,IY) ELSE RESET(IX,IY)
470 GOTO180
480 ' ETCH VALUE FOR CURSOR INCREMENTS
490 J=VAL(INKEY$):IFJ=0 THEN GOTO 490  ELSE RETURN
500 ' TRANSLATING ROUTINES HERE
510 FOR I=IB TO IB+IH*64 STEP 64
520 FOR J=0TOLG:IP=I+J
530 IF PEEK(IP)=128 POKE IP,32
535 IFPEEK(IP)>0ANDPEEK(IP)<27 THEN POKE IP,PEEK(IP)+64
540 IG(IP-15360)=PEEK(IP)
550 NEXTJ:NEXTI
560 RETURN
570 FOR I=1TO1024:PRINTCHR$(IG(I));:NEXT
580 RETURN
590 'THIS ROUTINE REDRAWS GRAPHIC AND RETURNS FOR ALTERATION
600 CLS
610 FOR I=IB TO IB+IH*64STEP64
620 FOR J=0TOLG:IP=I+J
630 POKE IP,IG(IP-15360)
640 NEXT J:NEXTI
650 RETURN
660 'THIS ROUTINE TRANSLATES SCREEN & PRINTS STRING SERIES
670 G$="":FOR I=1TOLX*GG$(I)="":NEXTI
680 IF PEEK(14312)=63 THEN LP=1 ELSE LP=0
690 IF LG=64 THEN LF$="":LF=0:LE$="": ELSE LE$=CHR$(26):LF$="CHR
$(26)":LF=1
700 BS$=LF$+"+TRING$("+STR$(LG+LF)+",24)"
710 IF LG+LF>=64 BS$="+CHR$(9)"
720 IF LP=1LPRINT"NOTE BS$= ";BS$:LPRINT" "
730 IC=0:N=1:FOR I=IB TO IB+IH*64 STEP64
17124 IF IC=1 AND LP=1  LPRINT"+BS$";
750 IF IC=1 G$=G$+LE$+STRING$(LG+LF,24):GOSUB1080
760 IC=1:FOR J=0TOLG-1:IP=I+J
770 IF PEEK(IP)=128 POKEIP,32
780 IF PEEK(IP+1)INSTR128 POKEIP,32
790 IF PEEK(IP)=PEEK(IP+1)THEN N=N+1:GOTO 820
800 IF N=1 GOSUB 890  :GOTO820
810 GOSUB 920  :N=1
820 NEXTJ
830 IF N>1THENGOSUB920  :N=1
840 NEXTI
850 IF N>1GOSUB920
860 IF LP=1 LPRINT" "
870 RETURN
```

Program continues

Program continued

```
880 'THIS ROUTINE TRANSLATES SCREEN AND PRINTS CHR$ SERIES
890 IF LP=1 LPRINT "+CHR$(";PEEK(IP);")";
900 G$=G$+CHR$(PEEK(IP)):GOSUB1080
910 RETURN
920 IF LP=1 THEN GOTO 930  ELSE GOTO940
930 IF PEEK(IP)<>32 THEN  LPRINT"+STRING$(";N;",";PEEK(IP);")";
ELSE IF N<63 THEN LPRINT"+CHR$(";192+N;")"; ELSE LPRINT "+CHR$(1
0)";
940 IF PEEK(IP)<>32 THENG$=G$+STRING$(N,PEEK(IP)) ELSE IF N<63 T
HENG$=G$+CHR$(192+N) ELSE G$=G$+CHR$(10)
950 GOSUB1080 :N=1
960 RETURNP>READTHEN'ROUTINE TO DETERMINE POKE POSITION ON SCRE
EN FOR IX,IY
980 ID= IY/3
990 IO=IX/2
1000 LO =ID*64 +IO  +15360
1010 RETURN
1020 'ROUTINE TO CALCULATE BLOCK
1030 IF IB=0 AND IE=0 THEN LG=64:IH=15:IB=15360:IE=16383:RETURN
1040 LG=(MX-NX)/2+1.6:IFLG>64 THENLG=64
1050 IH=(IE/64+n99)-(IB/64+.99)
1060 IF IH>15:IH=15
1070 RETURN
1080 'MULTIPLE STRING ROUTINE
1090 IF LEN(G$)+LG >253 THEN LX=LX+1:GG$(LX)=G$:G$="" ELSE RETUR
N
1100 EDITLP=1 LPRINT" ":LPRINT" "
1110 RETURN
1120 W$=STRING$(255,191):PRINT@0,W$;:FOR I=1TO3:PRINTW$;:NEXTI:W
$="":POKE16383,191:POKE16382,191:POKE16381,191:RETURN
1130 ' PRINT GRAPHIC F
21280 TRING CONCATENATIONS IN GG$
1140 IFA$="M"THEN A$="":K=LO-15360 ELSE K=IB-15360:CLS
1150 PRINT@K,;:FORI=1TOLX:PRINTGG$(I);:NEXT:PRINTG$;:RETURN
1160 CLS:PRINT"ARROWA CAUSE BLINKING CURSOR TO MOVE IN DIRECTION
 OF ARROW
S - WILL SET THE POSITION"
1170 PRINT "R - WILL RESET THE POSITION
B - INTERNALLY LOGS UPPER LEFT  POSITION OF GRAPHIC
E -             LOGS LOWER RIGHT POSITION OF GRAPHIC
F - FOLLOWED BY 1 TO 9 CONTROLS CURS
R JUMPS"
1180 PRINT"H - HOMES THE CURSOR
X - LOGS THE GRAPHIC FROM B TO E"
1190 PRINT"D - DISPLAY THE GRAPHIC USING CHR$ POKES
L - CONVER TO CONCATENATED STRINGS - PRINT IF PRINTER ON"
1200 PRINT"P - DRAW HIGH SPEED GRAPHIC (L MUST BE USED FIRST) -
IF NOT CORRECT ALTER`B OR E"
1210 PRINT"M - DRAW HIGH SPEED AT CURSOR POSITION USE L FIRST"
1220 PRINT"V - OUTPUT X,Y SET COORDINATES OF CURSOR"
1230 PRINT"Q - INSURT A LABEL AT CURSOR POSITION"
1240 INPUT"PRESS ENTER TO CONTINUE";D$
1250 PRINT"<SHIFT> D - DISPLAY WITHOUT CLS"
1260 PRINT"<SHIFT> W - WHITE`OUT THE SCREEN
<SHIFT> G - CREATE PROGRAM ON DISK TO DRAW GRAPHIC

           AND/OR PRINT CREATED PROGRAM"
1265 PRINT"<SHIFT> H -ELP WITH COMMANDS"
1270 PRINT"
BLINKING CURSOR INDICATES COMMAND COMPLETED
"
1280 INPUT"PRESS ENTER TO CONTINUE";D$
1290 RETURN
1300 'FAST CURCOR ROUTINE
1310 IF JA =8 THEN IY=IY-J: GOTO 1350
1320 IF JA =32 THEN IX=IX-J: GOTO1350
1330 IF JA =64 THEN IX=IX+J: GOTO1350
1340 IF JA =16 THEN IY=IY+J:GOTO1350
1350 IR=1:GOSUB370
1360 IF KA=4 A$="R": RESET(IX,IY):NC=0
1370 IF KA=8 A$="S": SET(IX,IY):NC=-1
1380 IR=0
1390 JA=PEEK(14400):JA=0
1400 GOTO460p
1410 ME$="":GOSUB980  :RESET(IX,IY):PRINT@LO-15360,"";
1420 B$=INKEY$:IF B$="" OR B$=CHR$(34)GOTO1420
1430 IF B$=CHR$(13) RETURN
1435 IFB$=CHR$(24) THEN GOTO1440 ELSE GOTO 1450
1440 B$="":IF LEN(ME$)>0THENME$=LEFT$(ME$,LEN(ME$)-1)
1450 ME$=ME$+B$
1460 PRINT@LO-15360,ME$;:GOTO1420
1470 'GENERATE BASIC PROGRAM IN ASCII NON-COMPRESSED HERE
1480 IFPEEK(14305)<>255 THEN OPEN "O",1,GN$
```

Program continues

Program continued

```
1490 GOSUB2150
1500 K=IB-15360:CC=0
1510 LI=0:GOSUB2140 :DG$=DG$+"REM **GRAPHICS IN THIS PROGRAM
GENERATED WITH GSETTR6/BAS
BY T. W. MUSTICO** NAMED "+GN$:GOSUB2040
1520 GOSUB2140 :DG$=DG$+"CLEAR 50":GOSUB2040
1530 GOSUB2140 :DG$=DG$+rDEFINTI-N:DEFSTRX-Z":GOSUB2040
1540 GOSUB2140 :DG$=DG$+"DIM DG$(9)":GOSUB2040
1550 LI=200
1560 KA=(K/64-INT(K/64))*64:GOSUB2070
1570 IC=0:FOR I=IB TO`IB+IH*64 STEP 64
1580 IF IC=1 THEN LI=LI+5:GOSUB2000 :DG$=DG$+DB$:GOSUB2030 :CC=C
C+BC
1590 LI=LI+5:GOSUB2000
1600 N=1:IC=1:FORJ=0TOLG-1:IP=I+J
1610 IFJ<L-1ANDPEEK(IP)=32ANDPEEK(IP+1)=32THENN=N+1:GOTO1710
1620 IFN=1 THEN II=PEEK(IP):GOSUB2010 :GOTO1710
1630 IFJ<>LG-1 GOTO1670
1640 IFPEEK(IP)=32 THEN II=192+N
1650 IF II>255 THEN II=193:GOSUB2010 :II=255
1660 GOSUB2010 :GOTO1700
1670 II=192+N:IF II>255THEN II=193:GOSUB2010 :II=255
1680 GOSUB2010
1690 IFPEEK(IP)<>32 THEN II=PEEK(IP):GOSUB2010
1700 N=1
1710 NEXTJ:GOSUB2030
1720 NEXTI
1730 LI=495
1740 GOSUB2140 :DG$=DG$+" REM CLEAR SPACE FOR "+STR$(CC)+ "  CHA
RACTERS.":GOSUB 2040
1750 NA=CC/22%:NB=CC-NA*225
1760 GOSUB2140 :DG$=DG$+"REM EDIT DG$("+STR$(NA)+") TO "+STR$(NB
)+" CHARACTERS":GOSUB2040
1770 GOSUB2140 :DG$=DG$+"I=0:J=0:A=0:II=0:AK=0:IG=0":GOSUB2040
1780 GOSUB2140 :DG$=DG$+"FOR I=0TO4":GOSUB2040
1790 GOSUB2140 :DG$=DG$+"B=VARPTR(DG$(I))":GOSUB2040
1800 GOSUB2140 :DG$=DG$+"REM CA\CULATE POKE ADDRESS":GOSUB2040
1810 GOSUB2140 :DG$=DG$+"A =PEEK(B+2)*256+PEEK(B+1)":GOSUB2040
1820 GOSUB2140 :DG$=DG$+"IF A>32767 THEN A=-1*(65e36-A)":GOSUB20
40
1830 GOSUB2140 :DG$=DG$+"REMREAD AND THEN POKE EACH CHARACTER":G
OSUB2040
1840 GOSUB2140 :DG$=DG$+"FOR J=1TO225":GOSUB2040
1850 GOSUB21d0 :DG$=DG$+" IG=A+J-1":GOSUB2040
1860 GOSUB2140 :DG$=DG$+"IF I*225+J>"+STR$(CC)+"GOTO900":GOSUB20
40
1870 GOSUB2140 :DG$=DG$+"READ II:POKE IG,II:NEXTJ":GOSUB2040
1880 GOSUB2140 :DG$=DG$+"NEXTI":GOSUB2040
1890 LI=895:GOSUB2140 :DG$=DG$+"DELETE 205-900":GOSUB2040
1900 GOSUB2140 :DG$=DG$+"CLS:IK=INT("GOTOSTR$(CC)+"/225):PRINT@"
+STR$(K)+",Z$;":GOSUB2040
1910 GOSUB2140 :DG$=DG$+"FOR I=0TOIK:PRINT DG$(I);:NEXTI":GOSUB2
040
1920 LI=945:GOSUB2140 :DG$=DG$+"OTO950":GOSUB2040
1930 LI=45
1940 IFCC<225GOTO1980
1950 FORI=0TONA-1
1960 GOSUB2140 :DG$=DG$+"DG$("+STR$(I)+")="+F$:GOSUB2040
1970 NEXTI
1980 IFNB>0:GOSUB2140 :G$=DG$+"DG$("+STR$(NA)+")="+LEFT$(F$,NB+
1)+CHR$(34):GOSUB2040
1990 IFPEEK(14305)<>255 THENCLOSE:RETURN ELSE RETURN
2000 DG$=RIGHT$(STR$(LI),3)+" DATA":RETURN
2010 IF LEN(DG$)>235 THEN GOSUB2030 :LA=LI:LI=LI-4:GOSUB2140 :LI
=LA:DG$=DG$+"DATA"
2020 CC=CC+1:DG$=DG$+RIGHT$(STR$(II),3)+",":RETURN
2030 IF RIGHT$(DG$,1)="," THEN KG=LEN(DG$)-1:DG$=LEFT$(DG$,CG)
2040 IF PEEK(14305)<>255 THENPRINT#1,DG$
2050 IF MP=1 LPRINT DG$
2060 RETURN
2070 KC=3:DB$="26":IF KA=0 THEN KC=1
2080 IFKA+LG>=64 KC=2
2090 IFKA=0 AND KC=2 THEN D
14626 ":BC=1:RETURN
2100 ON KC GOTO 2110 ,2120 ,2130
2110 DB$="10":BC=1:RETURN
2120 DB$="9"
2130 FOR I=1TOLG:DB$=DB$+","+"24":NEXTI:BC=LG+1:RETURN
2140 LI=LI+5:DG$=RIGHT$(STR$(LI),3):RETURN
2150 C$="THIS IS A DUMMY STRING "
2160 F$=C$+C$+C$+C$+C$+C$+C$+C$+C$
2170 F$=LEFT$(F$,225)
2180 F$=CHR$(34)+F$+CHR$(34)
2190 RETURN
```

FREE SHIPPING

(ONLY IN CONTINENTAL U.S.)

MODEL III DISK KITS

TWO 40/track drive kit and hardware. Comes complete with all the necessary hardware and 2 drives of your choice, (MPI or TANDON 40 track). NO SOLDERING OR TRACE CUTTING REQUIRED! COMPLETE EASY TO FOLLOW INSTRUCTIONS INCLUDED **875.95**

TWO 80/track drive kit and hardware. Same as above **1,095.00**

ONE 40/track drive kit with all hardware and your choice of MPI or TANDON 40/ track drive. (INSTRUCTIONS INCLUDED) **599.99**

ONE 80/track drive kit with all hardware and one MPI 80/track drive. No soldering! (INSTRUCTIONS INCLUDED) **699.99**
TRS-80 MODEL III, 48K, 2 40 Track TANDON or MPI drives **2,095.99**
TRS-80 MODEL III, 48K, 2 80 Track MPI drives **2,325.99**

1 YEAR WARRANTY

16 K MEMORY 19.95

MODEL I, III APPLE AND EXIDY. Instructions included. (MOD 1 Keyboard requires jumpers. $2.00 extra)

ARCHBOLD SPEED-UP MODIFICATION FOR MOD I TRS-80. RUN UP TO 300% FASTER CPU SPEED! SPEED UP SLOW PROGRAMS WITHOUT SOFTWARE! **45.00**

MONITORS

Black and White Monitors 12" (TECO) **129.95**
Green Phospher Monitors 12" (TECO) **159.95**
AMDEK Black and White 12" Monitor **159.95**
AMDEK Green Phopher 12" Monitor 100G **189.95**

MODEL I DISK DRIVES

These drives come complete with power supply and chassis. 90 day warranty on disk drives. ALL DRIVES COMPLETELY TESTED PRIOR TO SHIPPING! 40 track drives are fully compatible with TRSDOS, NEWDOS-80, LDOS etc. Fully compatible with the PERCOM DOUBLER and Double density! 80 track drives require MAKE-80 program to run regular software. (SPECIAL! GET A FREE 2 DRIVE CABLE WHEN BUYING 2 or more DRIVES!)

40 TRACK MPI or TANDON DRIVES **319.95**
80 TRACK MPI **439.95**
DOUBLE SIDED 80 TRACK MPI **579.95**
2 DRIVE CABLES **25.95**
4 DRIVE CABLES **35.95**
SPECIAL DOUBLE SIDED CABLES .... (CALL $)
MAKE-80 CONVERT 40 TRACK TO 80 TRACK. (SPECIFY MOD I, III) **29.95**

DISKETTES

SPECIAL: Brown Bag Diskettes (Verbatim) Bag of 20 . **49.95**
SPECIAL: Brown Bag Diskettes 100 Quantity **214.00**
VERBATIM DATALIFE DISKETTES Box of 10 .... **32.50**
VERBATIM DATALIFE DISKETTES 100 Quantity **275.00**

SPECIAL DISK HOLDER

Holds 50 Diskettes
Reg. Price $35.95
SPECIAL **$21.95**

PERCOM DOUBLER II (Includes DBLDOS) Doubles storage on Model I disk drives. Installs in Exp. Int. in minutes! Will not effect normal operation! **169.95**
NEWDOS-80 (New Version 2) **139.95**
LDOS **139.95**
DOSPLUS 3.3 .... **89.95**
(Specify Mod I or III)

LYNX TELEPHONE MODEM

AUTO/DIAL AUTO/ANSWER.
COMPLETE WITH SOFTWARE AND HARDWARE FOR MODELS I AND III
The RS-232 is not needed! Completely assembled. LYNX MODEMS are compatible with ST-80 III also **299.95**
ST-80 III: Smart terminal program. Allows saving data to disk or sending data from disk and more! Mod I or III ... **149.95**

PRINTERS

SPECIAL: FREE CABLES AND FREE SHIPPING WITH ALL PRINTERS!

CABLES FOR MODEL I EXPANSION INTERFACE, MOD III, or PMC-80 EXP 100 or PIF-40 only.)
PLEASE SPECIFY WHICH COMPUTER!

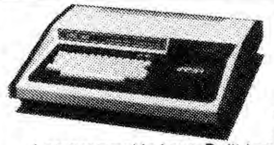
HIGH QUALITY DOT MATRIX

EPSON MX-80 (Tractor feed only) **539.95**
EPSON MX-80 FT (Tractor and Friction feed) **639.95**
EPSON MX-70 (Tractor feed only) **419.95**
EPSON MX-100 (Tractor and Friction feed 14.5 wide) . **895.95**
EPSON MX-80 Hi-Res Graphics Option **89.95**
Add-On Friction Feed Adaptor for MX-80 or MX-70 **59.95**
OKIDATA MICROLINE 80 (Friction and Pin feed) .. **414.95**
OKIDATA MICROLINE 82 (Friction and Pin feed) .. **597.95**
OKIDATA MICROLINE 83 .... **897.95**
(Friction, Pin and Tractor feed 14.5")
OKIDATA MICROLINE 82A. 120 cps **625.00**
(9x9 matrix w/descenders)
OKIDATA MICROLINE 83A. 120 cps **925.00**
(9x9 matrix w/descenders)
TRACTOR FEED ADAPTOR for 80 and 82 .... **75.00**

LETTER QUALITY PRINTERS

The C-itoh Daisy wheel printer is one of the finest Letter Quality printers we've seen. WHY PAY MORE FOR LESS??? Exchange Daisy wheels to get the kind of printing fonts you need. 12-10 pitch. Excellent with Scripsit or Electric pencil. Let's face it, sometimes dot matrix doesn't hack it! (You can use standard Diablo Carbon ribbons with the C-itoh to get crisp "COPY READY" printouts!)

C-ITOH DAISY WHEEL 25 CPS **1,795.25**
C-ITOH DAISY WHEEL 45 CPS **1,995.25**
C-ITOH TRACTOR FEED **219.95**



PMC-80 COMPUTER

16K LEVEL II Fully compatible with all TRS-80 Mod I Software **595.95**

PMC-81 COMPUTER 16K LEVEL II with upper/lower case. Upper/lower case driver in ROM. Auto key repeat. Numeric keypad. 4 programmable keys. Built in speaker. 100% Mod I software compatible **679.99**
EXPANDER 100 PMC's expansion interface 32K, disk controller, printer interface. (Expands PMC-80/81 to 48K) **499.99**
RF-MODULATOR for PMC or TRS-80 to TV Hook-up. Completely assembled and tested. (Plugs in Video port) UHF CH.33 .... **39.95**

TRS-80 Disk and Other Mysteries Book, 132 pp **22.50**
Microsoft Basic Decoded and Other Mysteries, 312 pp **29.95**
Custom TRS-80 and Other Mysteries, 300 pp **29.95**

CALL TOLL FREE: 800-528-1149

NO SURCHARGE FOR VISA, MASTER CARD OR AMERICAN EXPRESS!!! (10% DEPOSIT REQUIRED ON C.O.D.'S OVER $200.00) Send Certified Check, Money Order or Check.

✓19

SIMUTEK COMPUTER PRODUCTS INC.

4877 E. SPEEDWAY
TUCSON, AZ 85712
(602) 323-9391

CALL FOR FREE 100+ PAGE CATALOG!!!

TRS-80 IS A TM OF TANDY CORP. PMC-80 IS TM OF PERSONAL MICROCOMPUTERS. PRICES AND SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE.

✓See List of Advertisers on page 290

EDUCATION

LOAD 80

The Key Box

Basic Level II
Model I
16K RAM

Help school children learn the rules of punctuation.

To Comma or Not to Comma

John D. Perron
527-41st Avenue
San Francisco, CA 94121

One of the most arbitrary writing skills taught in school is punctuation. A teacher may introduce a rule for using a comma after an introductory clause, only to have a student come up with a published manuscript that breaks the rule. Even with general agreement on the basic rules, writers find distinctive situations where the rules have to be bent.

So why not use "real writing" to teach usage?

Punctuation Put-On helps teachers do this by presenting different styles of writing in a game format. It includes a diagnostic feature that keeps track of errors and helps students come to grips with their individual weaknesses. A Level II program, Punctuation Put-on uses 246 lines (see Table 1) or nearly all of 16K. Its fairly rapid motion (see Fig. 1 for flowchart) is based on positioning the most used subroutines up front and eliminating all superfluous semicolons.

Line Numbers	*Program Information*
1-8	Title
9-21	Wrong answer subroutine
22-24	Typing help subroutine
25-32	Correct answer subroutine
33-40	Run-time reminders
41-42	Entry points for selections
43-84	Cursor placement for different selections
85-87	Blinking cursor/answer check
88-90	End of selection/assign score
91-94	Start again/end
95-97	Writing selections
98-107	Choice processing
108-119	Full writing of Selection #1 "Ken Asks His Mother"
120-129	Full writing of Selection #2 "Reward for the Robot"
130-141	Full writing of Selection #3 "A Not-so-fast Fly"
142-155	Punctuation Mark Display—Typing Help
156-165	Final Score subroutine
166-177	Diagnosis of errors
178-181	No errors subroutine
182-216	Graphic eyes subroutine
217-246	Student instructions

Table 1. Program Lines for Punctuation Put-On

Why Punctuation?

Punctuation is a way of demonstrating the pauses, gestures, and emotional content of speech; the rules written communication requires for exactness. Yet, creativity depends on expanding the rules somewhat—and that can only be taught in actual writing situations.

Punctuation Put-On tries to provide a close sense of the real writing event. The program is written for upper elementary school students. Table 2 illustrates the punctuation marks in Punctuation Put-On and places them by grade where they are taught in the first eight years of school.

How It Works

To use Punctuation Put-On, a student simply chooses one of three writing selections:

Selection 1:	Ken Asks His Mother	(Hard)
Selection 2:	Reward for the Robot	(Harder)
Selection 3:	A Not-so-fast Fly	(Hardest)

Selection 1 is mainly dialogue, with punctuation arising through directly quoted speech (see Fig. 2). Selection 2 is a narrative in which the punctuation depends more on the flow of events than on quotations. And finally, selection 3 is a poem in which the punctuation appears totally arbitrary (although a poet always has subtle reasons for the choices!).

Students may wish to repeat the same selection, or they may prefer to play all

"One of the most arbitrary writing skills taught in school is punctuation."

three (in any order, although each is weighted according to difficulty). Prior to the first use of a selection, the program presents it slowly in its entirety to allow the student to read or study it. There are also run-time reminders (brief instructions) during the program which supplement the longer instructions presented at the start of the program. However, after a selection has been once used, the reminders and study time are eliminated, allowing students faster movement between selections.

Each writing selection covers the upper two-thirds of the screen, leaving the bottom third clear for processing answers. The computer randomly selects a punctuation mark and replaces it with a blinking cursor. Students must decide which punctuation mark goes there, based on the context of the writing; that is, they see what goes before and what comes after the punctuation mark, which helps them make an informed decision. When they type in their choice, it appears in the actual screen position—and the computer immediately tells them whether they're right or wrong.

If they're right, the computer prints an asterisk or a verbal response. The program moves on to another punctuation mark at the student's request, repeating this procedure 10 times for each writing selection. If the student's choice is wrong (for instance, the choice is an exclamation mark when the correct answer is a question mark), the computer responds with: "No, it is not (!) It is (?) Watch it blink . . ." And the question mark blinks in place on the screen.

When the program begins the student is shown where the punctuation marks are located on the keyboard. Whenever the student makes an error when responding to the blinking cursor, he or she is again offered the help routine for locating punctuation marks.

Scoring

After each selection, students see their score (number correct out of 10). When a student elects to quit playing, the final scores are presented, including the last student scores on the three selections (and their overall total). When there are errors, the student is shown a diagnostic chart containing the type and number of punctuation errors (see Table 3). The student can use this information to focus on his or her actual weaknesses in punctuation skills. If a student completes any number of selections without making an error, the program moves to a congratulatory display, allowing the student a chance to share the score with the teacher (via an option to loop back through the final score routine).

I prefer diversions to rewards—especially when rewards must be repeated again and again in rapid-fire motion. I have, therefore, included a graphic eyes subroutine. As a diversion, it occurs on the second and fifth writing selection. It appears as if a little man inside the computer is having trouble finding the selection.

True learning depends on how close the practice situation comes to the real thing. The major goal of any course design, textbook or software, should be a learning situation that complements or extends the natural writing or reading act. That's a main ingredient of Punctuation Put-On, placing the learner in the shoes of the writer and providing full contextual support.

Novice Courage

I'd like to summon all my novice courage and admit one thing: This is my first attempt at programming. I bought my TRS-80 last June and have been struggling to learn Basic ever since. I'm still learning, and I'm sure there may be better ways of doing some of the things in Punctuation Put-On (if you find them, let me know!). Still, the bottom line is that everything works, and there are enough error traps to allow even the youngest user to operate the program easily and smoothly.

One interesting problem I ran across

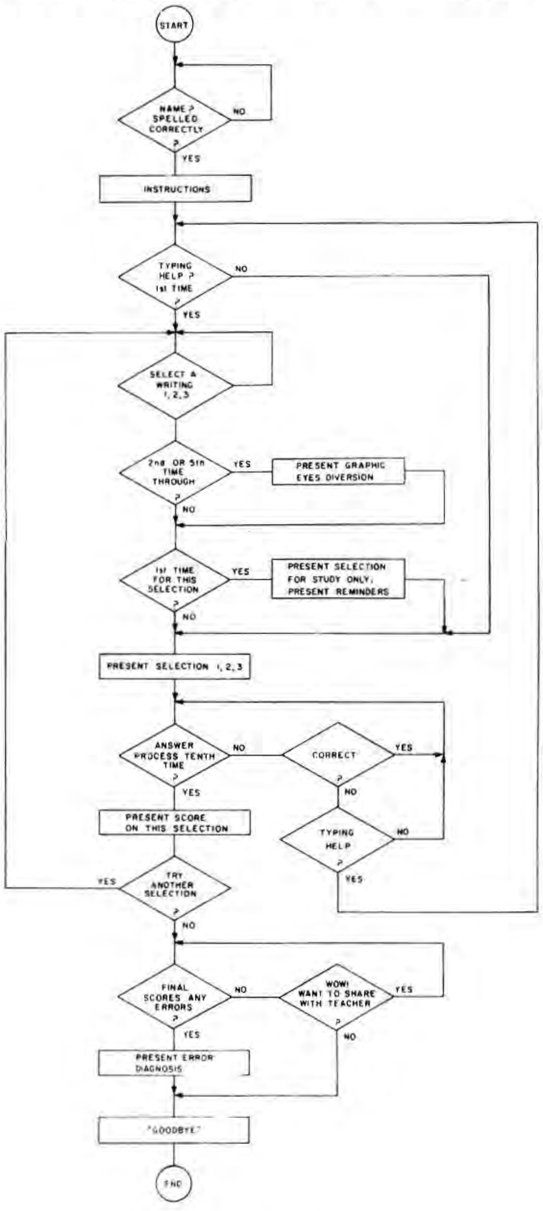


Fig. 1. Flowchart

See List of Advertisers on page 290

"I'd like to summon all my novice courage and admit one thing: This is my first attempt at programming."

may help others overcome a similar difficulty. I needed some kind of routine early in lines 43–84 to allow me to randomly select 20 numbers (NC) without repeating any one. A friend came up with the answer: Set up a 20-position array, P(NC), and treat each array position as a signal holder. Thus, when an NC number has been randomly selected, P(NC) is checked; if the array position of that number is a zero, the number has not been used before; the program then uses the number and places a one in that array position. The next time through, if that same NC number is randomly selected, there will already be a one in the array position, so the computer will return for another selection until it finds one with a zero array position.

Ken wanted to go to the circus. After school he came home and asked his mother: "Can I go■"
"Where?" she asked.
"You know," he said
"No," she answered, "I don't know."
"But all the other kids are going!" he shouted.
"But where are they going?" she shouted back.
"Oh," Ken said. "They're going to the circus."
"Oh," his mother laughed, "Well, why didn't you say so?"
Ken looked at her. "Well, can I go?"
"Go ask your father!" his mother said and turned away.

Fig. 2. Excerpt from Selection 1. The cursor blinks in place of a punctuation mark. The student decides what belongs there and types it in. This is repeated at ten randomly selected punctuation mark positions (out of 20 per selection). The student may then repeat the selection, or choose one of the other two.

More Memory?

If you have more memory at your disposal, let me offer one suggestion for upgrading the program. As it stands, a student learns the number and type of errors made, but since the initial selection process is random, one never knows how many times a punctuation mark may have been presented during a run. By adding a counter to the wrong answer routine (lines 9–21) and placing this information in a separate column in the diagnostic chart (lines 166–177), the teacher and student would benefit by knowing whether the number of errors was significantly related to the number of times the mark was actually presented in the program.

Another possible extension involves adding to or replacing the selections used in this collection. Those in the program are my own original works. Replacing these writings with your own is a fine idea—even better would be to replace them with

"Students have to bring a knowledge of punctuation to the program."

the writings of your students. The job of locating the cursor positions of the punctuation marks is a tough one, but there's always one student who would benefit from the challenge! The main thing to remember is that each selection should contain at least 20 punctuation marks (in lines 43-84).

One final note: Punctuation Put-On can't tell students anything about the rules of punctuation: It is not trying to replace the teacher. Students have to bring a knowledge of punctuation to the program. It then gives them a chance to test their skills against the variety that actually exists in the written word—and all in the spirit of fun! ■

John Perron is a curriculum writer of language arts and course software, and has published several books on these topics.

Punctuation Mark	*Grade Levels*
Ending:	
Period (.)	1-8
Question Mark (?)	1-8
Exclamation Mark (!)	1-8
Other:	
Comma (,)	1-8
Apostrophe (')	3-8
Quotation Mark (")	4-8
Semicolon (;)	5-8
Colon (:)	5-8
Hyphen (-)	5-8

Table 2. Punctuation at Elementary Grade Levels

Here are the kinds of errors you made, if any:

Type of Error	*Number of Errors*
Exclamation Mark (!)	1
Quotation Mark (")	0
Apostrophe (')	4
Question Mark (?)	0
Comma (,)	1
Period (.)	0
Semicolon (;)	0
Colon (:)	0
Hyphen (-)	0
Total	6

Write down your score before your press <Enter>?

Table 3. Error Diagnostics. Final score comes just before the diagnostics, so a student knows how many selections have been attempted and the individual and total scores overall.

Program Listing

```
1 REM MIXED GRADES (uPPER ELEMENTARY) SAMPLER VERSION 6.6
2 CLEAR 200:DEFINT A-Z
3 DIM P(20):CLS
4 PRINT@204,CHR$(23)"PUNCTUATION PUT-ON"
5 PRINT@348,"BY";:PRINT@464,"JOHN D. PERRON"
6 PRINT@590,"527 - 41ST AVENUE"
7 PRINT@708,"SAN fRANCISCO, caLIF 94121"
8 PRINT@848,"COPYRIGHT 1980":FOR T=1 TO 4500:NEXT T:GOSUB 218:GO
TO 96
9 REM             ***  WRONG ANSWER SUBROUTINE  ***
10 IF B$=CHR$(13) THEN B$="<ENTER>"
11 NC=0:C=C+1:PRINT@768,"NO, IT IS NOT ( "B$" )   IT IS ( "C$" )
   WATCH IT BLINK...";:FOR T=1 TO 500:NEXTT
12 FOR J=1 TO 30:PRINT@LC,C$;:FOR T=1 TO 25:NEXT T:PRINT@LC," ";
:FOR K=1 TO 25:NEXT K:NEXT J:PRINT@LC,C$;
13 IF C$="!" THEN W1 = W1 + 1:GOTO 23
14 IF C$=CHR$(34) THEN W2 = W2 +1:GOTO 23
```

Program continues

Program continued

```
15 IF C$="'" THEN W3 = W3 + 1:GOTO 23
16 IF C$="?" THEN W4 = W4 + 1:GOTO 23
17 IF C$="," THEN W5 = W5 + 1:GOTO 23
18 IF C$="." THEN W6 = W6 + 1:GOTO 23
19 IF C$=";" THEN W7 = W7 + 1:GOTO 23
20 IF C$=":" THEN W8 = W8 + 1:GOTO 23
21 IF C$="-" THEN W9 = W9 + 1
22 REM          ***  TYPING HELP REQUEST  ***
23 R$="":PRINT@896,"WOULD YOU LIKE TYPING HELP (Y=YES/N=NO)?";:R
$=INKEY$:IF R$="" THEN 23
24 IF R$ <> "Y" THEN PRINT@768,STRING$(3,255);:GOTO 42 ELSE CLS:
GOSUB 143:CLS:GOTO 41
25 REM       ***    CORRECT ANSWER SUBROUTINE    ***
26 NC=0:C=C+1:R=R+1:G=RND(14):ON G GOTO 27,27,28,27,27,29,27,27,
30,27,27,31,27,27
27 PRINT@768,TAB(20)"*":GOTO 32
28 PRINT@768,TAB(20)"VERY GOOD, "Z$"!":GOTO 32
29 PRINT@768,TAB(20)"GREAT, "Z$"!":GOTO 32
30 PRINT@768,TAB(20)"THAT'S CORRECT.":GOTO 32
31 PRINT@768,TAB(20)"THAT'S RIGHT."
32 FOR T=1 TO 300:NEXT T:PRINT@896,"PRESS <ENTER> WHEN READY FOR
 THE NEXT ONE";:INPUT I$:PRINT@768,STRING$(3,255);:RETURN
33 REM  ***  RUN-TIME REMINDERS  ***
34 CLS:PRINT@448,"HERE IS PUNCTUATION PUT-ON STORY #";S:PRINT:PR
INT"READ IT FIRST.  THEN LOOK AT THE PUNCTUATION.":PRINT:PRINT:F
OR T=1 TO 2000:NEXT T:ON S GOSUB 109,121,131
35 PRINT@916,"WHEN YOU FINISH READING, PRESS <ENTER>";:INPUT I$:
CLS
36 PRINT@128,STRING$(63,42);:PRINT@215,"*** REMINDER ***":PRINT@
389,"THE BLINKER FLASHES WHERE A PUNCUATION MARK SHOULD GO."
37 PRINTTAB(5)"TAKE YOUR TIME.  i'LL WAIT UNTIL YOU CAN DECIDE W
HICH"
38 PRINTTAB(5)"MARK GOES THERE.  WHEN YOU TYPE YOUR CHOICE, IT W
ILL"
39 PRINTTAB(5)"SHOW UP ON THE SCREEN IN PLACE OF THE BLINKER.":P
RINT@704,STRING$(63,42);
40 PRINT@920,"READY TO BEGIN?  PRESS <ENTER>";:INPUT I$:CLS
41 IF C>=10 THEN 89 ELSE ON S GOSUB 109,121,131
42 IF C>=10 THEN 89
43 REM          ***  ON-SCREEN PRINT@ SELECTOR  ***
44 NC=RND(20):IF P(NC)=1 THEN 44 ELSE P(NC)=1
45 ON NC GOTO 46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78
,80,82,84
46 IF S=1,LC=35:C$=CHR$(46) ELSE IF S=2,LC=57:C$=CHR$(46) ELSE L
C=8:C$=CHR$(45)
47 GOTO 86
48 IF S=1,LC=96:C$=CHR$(63) ELSE IF S=2,LC=113:C$=CHR$(46) ELSE
LC=31:C$=CHR$(59)
49 GOTO 86
50 IF S=1,LC=140:C$=CHR$(34) ELSE IF S=2,LC=143:C$=CHR$(46) ELSE
 LC=56:C$=CHR$(33)
51 GOTO 86
52 IF S=1,LC=206:C$=CHR$(44) ELSE IF S=2,LC=197:C$=CHR$(34) ELSE
 LC=137:C$=CHR$(44)
53 GOTO 86
54 IF S=1,LC=151:C$=CHR$(46) ELSE IF S=2,LC=221:C$=CHR$(34) ELSE
 LC=148:C$=CHR$(33)
55 GOTO 86
56 IF S=1,LC=87:C$=CHR$(34) ELSE IF S=2,LC=220:C$=CHR$(63) ELSE
LC=155:C$=CHR$(63)
57 GOTO 86
58 IF S=1,LC=261:C$=CHR$(34) ELSE IF S=2,LC=294:C$=CHR$(46) ELSE
 LC=265:C$=CHR$(46)
59 GOTO 86
60 IF S=1,LC=279:C$=CHR$(44) ELSE IF S=2,LC=358:C$=CHR$(44) ELSE
 LC=272:C$=CHR$(44)
61 GOTO 86
62 IF S=1,LC=358:C$=CHR$(33) ELSE IF S=2,LC=386:C$=CHR$(39) ELSE
 LC=277:C$=CHR$(45)
63 GOTO 86
64 IF S=1,LC=389:C$=CHR$(34) ELSE IF S=2,LC=453:C$=CHR$(34) ELSE
 LC=288:C$=CHR$(33)
65 GOTO 86
66 IF S=1,LC=414:C$=CHR$(63) ELSE IF S=2,LC=468:C$=CHR$(33) ELSE
 LC=312:C$=CHR$(33)
67 GOTO 86
68 IF S=1,LC=433:C$=CHR$(46) ELSE IF S=2,LC=475:C$=CHR$(39) ELSE
 LC=392:C$=CHR$(44)
69 GOTO 86
70 IF S=1,LC=456:C$=CHR$(44) ELSE IF S=2,LC=493:C$=CHR$(44) ELSE
 LC=399:C$=CHR$(44)
71 GOTO 86
72 IF S=1,LC=474:C$=CHR$(39) ELSE IF S=2,LC=503:C$=CHR$(46) ELSE
 LC=405:C$=CHR$(44)
73 GOTO 86
74 IF S=1,LC=543:C$=CHR$(34) ELSE IF S=2,LC=517:C$=CHR$(34) ELSE
 LC=411:C$=CHR$(59)
75 GOTO 86
```

Program continues

✓See List of Advertisers on page 290

Program continued

```
76 IF S=1,LC=558:C$=CHR$(39) ELSE IF S=2,LC=520:C$=CHR$(44) ELSE
 LC=441:C$=CHR$(58)
77 GOTO 86
78 IF S=1,LC=571:C$=CHR$(63) ELSE IF S=2,LC=528:C$=CHR$(44) ELSE
 LC=522:C$=CHR$(33)
79 GOTO 86
80 IF S=1,LC=601:C$=CHR$(34) ELSE IF S=2,LC=529:C$=CHR$(34) ELSE
 LC=529:C$=CHR$(63)
81 GOTO 86
82 IF S=1,LC=616:C$=CHR$(63) ELSE IF S=2,LC=570:C$=CHR$(63) ELSE
 LC=542:C$=CHR$(33)
83 GOTO 86
84 IF S=1,LC=698:C$=CHR$(46) ELSE IF S=2,LC=602:C$=CHR$(34) ELSE
 LC=570:C$=CHR$(33)
85 REM          ***  ANSWER PRINT@ POSITION  ***
86 B$="":PRINT@LC,CHR$(128)+CHR$(8);:B$=INKEY$:Z=NOT Z:FOR X=1 T
O 15:PRINT CHR$(15+Z);:NEXT X:IF B$="" THEN PRINT CHR$(15);:GOTO
 86 ELSE IF B$=CHR$(13) PRINT CHR$(15);:GOTO 87 ELSE PRINT B$;CH
R$(15);
87 IF B$ <> C$ THEN 10 ELSE GOSUB 26:GOTO 42    ' CHECK ANSWER
88 REM          ***  END A STORY/ASSIGN SCORES  ***
89 CLS:PRINT@256,"THAT'S ALL FOR THIS STORY.  YOU GOT "R" OUT OF
 10.":IF S=1 THEN G1=R ELSE IF S=2 THEN G2=R ELSE IF S=3 THEN G3
=R
90 PRINT:PRINT"THAT MEANS A SCORE OF"INT(R/10*100)"PER CENT FOR
STORY #"S".":R=0
91 REM                 ***  RECYCLE OR END  ***
92 R$="":PRINT@704,"WANT TO TRY ANOTHER STORY (Y=YES/N=NO)?";:R$
=INKEY$:IF R$="" THEN 92
93 IF R$ <> "N" AND R$ <> "Y" THEN CLS:PRINT@930,"PLEASE TYPE Y
OR N.":GOTO 92 ELSE IF R$="N" THEN CLS:GOSUB 157:PRINT@470,"GOOD
BYE FOR NOW.":GOTO 94 ELSE IF R$="Y" THEN CLS:C=0:GOTO 97
94 END
95 REM                 ***  SELECTIONS  ***
96 PRINT"O.K., "Z$", WHICH SELECTION WOULD YOU LIKE TO TRY FIRST
?":PRINT:PRINT:F=F+1:GOTO 100
97 FOR NC=1 TO 20:P(NC)=0:NEXT NC:CLS:PRINT"ALL RIGHT, "Z$", PIC
K ANOTHER SELECTION:":F=F+1:PRINT
98 REM          ***  SELECTION TITLES  ***
99 REM             ***  DIALOGUE  ***
100 PRINT:PRINTTAB(5)"SELECTION #1:  KEN ASKS HIS MOTHER     (HA
RD)"
101 REM            ***  NARRATIVE  ***
102 PRINT:PRINTTAB(5)"SELECTION #2:  REWARD FOR THE ROBOT    (HA
RDER)"
103 REM            ***   POETRY    ***
104 PRINT:PRINTTAB(5)"SELECTION #3:  A NOT-SO-FAST FLY        (HA
RDEST)":PRINT
105 S=0:PRINT@832,"PICK YOUR CHOICE AND TYPE ITS NUMBER (1, 2, O
R 3):  ";:S$=INKEY$:IF S$="" THEN 105 ELSE S=VAL(S$)
106 IF S=>4 OR S<=0 THEN PRINT:PRINTTAB(30)"THE NUMBER MUST BE 1
, 2, OR 3":FOR T=1 TO 500:NEXT T:CLS:GOTO 100 ELSE IF F=2 OR F=5
 OR F=9 THEN CLS:GOSUB 183
107 IF S=1 AND G1>0 OR S=2 AND G2>0 OR S=3 AND G3>0 THEN 41 ELSE
 GOTO 34
108 REM                ***  SELECTION # 1  ***
109 CLS:PRINTTAB(5)"KEN WANTED TO GO TO THE CIRCUS.  aFTER SCHOO
L HE CAME HOME"
110 PRINT"AND ASKED HIS MOTHER:  "CHR$(34)"CAN i GO?"CHR$(34)
111 PRINTTAB(5)CHR$(34)"WHERE?"CHR$(34)" SHE ASKED."
112 PRINTTAB(5)CHR$(34)"YOU KNOW,"CHR$(34)" HE SAID."
113 PRINTTAB(5)CHR$(34)"NO,"CHR$(34)" SHE ANSWERED, "CHR$(34)"i
DON'T KNOW."CHR$(34)
114 PRINTTAB(5)CHR$(34)"BUT ALL THE OTHER KIDS ARE GOING!"CHR$(3
4)" HE SHOUTED."
115 PRINTTAB(5)CHR$(34)"BUT WHERE ARE THEY GOING?"CHR$(34)" SHE
SHOUTED BACK."
116 PRINTTAB(5)CHR$(34)"OH,"CHR$(34)" KEN SAID. "CHR$(34)"THEY'R
E GOING TO THE CIRCUS."CHR$(34)
117 PRINTTAB(5)CHR$(34)"OH,"CHR$(34)" HIS MOTHER LAUGHED, "CHR$(
34)"WELL, WHY DIDN'T YOU SAY SO?"CHR$(34)
118 PRINTTAB(5)"KEN LOOKED AT HER:  "CHR$(34)"WELL, CAN I GO?"CH
R$(34)
119 PRINTTAB(5)CHR$(34)"GO ASK YOUR FATHER!"CHR$(34)" HIS MOTHER
 SAID AND TURNED AWAY.":RETURN
120 REM                ***  SELECTION # 2  ***
121 CLS:PRINTTAB(5)"ONE DAY A ROBOT NAMED WUNTOO WAS ROLLING DOW
N A ROAD.  ALL"
122 PRINT"OF A SUDDEN IT SPIED A BAG OF MONEY IN THE BUSHES.  TH
ERE WAS"
123 PRINT"NO ONE IN SIGHT.":PRINTTAB(5)CHR$(34)"I WONDER WHO LOS
T THIS?"CHR$(34)" BEEPED WUNTOO."
124 PRINTTAB(5)"ON THE BAG WAS THE NAME OF A BANK.  wUNTOO TOOK
IT TO"
125 PRINT"THE BANK MANAGER.  WHEN WUNTOO ARRIVED, THE BANK MANAG
ER SAID"
126 PRINT"HE'D JUST BEEN ROBBED."
```

Program continues

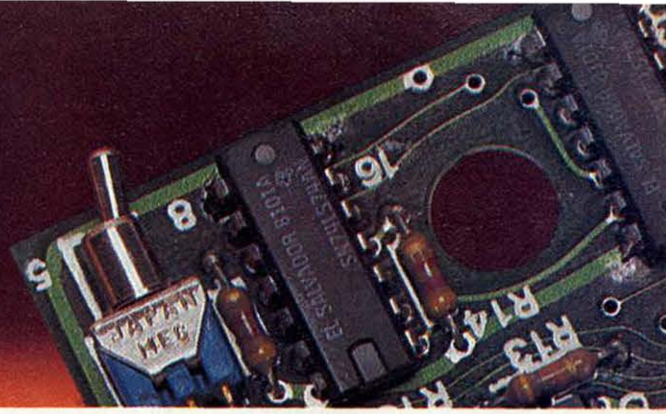
Program continued

```
127 PRINTTAB(5)CHR$(34)"YOU ARE A HERO!  HERE'S A $1,000 REWARD,
"CHR$(34)" HE SAID."
128 PRINTTAB(5)CHR$(34)"NO, THANKS,"CHR$(34)" BEEPED WUNTOO. "CH
R$(34)"bUT DO YOU HAVE ANY OIL?  i"
129 PRINT"HAVE A SQUEAK IN MY WHEEL."CHR$(34):RETURN
130 REM          ***  SELECTION # 3  ****
131 CLS:PRINTTAB(5)"ZIG-ZAG, BUZZ, FLASH, STOP;":PRINT
132 PRINTTAB(5)"BUZZ, ZIP, SLAP!  GONE?":PRINT
133 PRINTTAB(5)"BACK.  BUZZ, ZIG-ZAG, SLASH!":PRINT
134 PRINTTAB(5)"ZIP, FLASH, BUZZ, STOP;":PRINT
135 PRINTTAB(5)"SWISH!  MISS?  BUZZ, SLAP!":PRINT
136 FOR T=1 TO 1000:NEXT T:PRINT@53,"MAD!";:GOSUB 141
137 PRINT@181,"GRAB";:GOSUB 141
138 PRINT@309,"BAT!";:GOSUB 141
139 PRINT@437,"BUzZ:";:GOSUB 141
140 PRINT@565,"SPLAT!":RETURN
141 FOR T=1 TO 300:NEXT T:RETURN
142 REM       ***  PUNCTUATION DISPLAY  ***
143 PRINT"HERE'S WHERE TO FIND THE PUNCTUATION MARKS ON YOUR KEY
BOARD":PRINT
144 PRINTTAB(5)"PRESS THE SHIFT KEY WITH THE FOLLOWING:"
145 PRINT"(!) EXCLAMATION MARK -- TOP ROW ON LEFT (1ST KEY)"
146 PRINT"("CHR$(34)") QUOTATION MARK -- TOP ROW ON LEFT (2ND KE
Y)"
147 PRINT"(?) QUESTION MARK -- BOTTOM ROW ON RIGHT (NEXT TO SHIF
T KEY)
148 PRINT"(') APOSTROPHE -- TOP ROW IN MIDDLE (KEY #7)":PRINT
149 PRINTTAB(5)"THE FOLLOWING ARE TYPED ALONE (NO SHIFT KEY):"
150 PRINT"(,) COMMA -- BOTTOM ROW ON RIGHT (3RD FROM SHIFT KEY)"

151 PRINT"(.) PERIOD -- BOTTOM ROW ON RIGHT (2ND FROM SHIFT KEY)
"
152 PRINT"(;) SEMICOLON -- SECOND ROW FROM BOTTOM (NEXT TO ENTER
 KEY)"
153 PRINT"(:) COLON -- TOP ROW ON RIGHT (STAR KEY *)"
154 PRINT"(-) HYPHEN -- TOP ROW ON RIGHT (NEXT TO BREAK KEY)"
155 PRINT:PRINTTAB(15)"PRESS <ENTER> WHEN READY TO GO ON";:INPUT
 I$:CLS:RETURN
156 REM          **** FINAL SCORE SUBROUTINE ****
157 CLS:IF G1<=0 THEN 158 ELSE PRINT:PRINT"YOU GOT"G1"OUT OF 10
ON SELECTION #1":PRINTTAB(30)"SCORE...."INT(G1/10*100)"%":PRINT
158 IF G2<=0 THEN 159 ELSE PRINT"YOU GOT"G2"OUT OF 10 ON SELECTI
ON #2":PRINTTAB(30)"SCORE...."INT(G2/10*100)"%":PRINT
159 IF G3<=0 THEN 160 ELSE PRINT"YOU GOT"G3"OUT OF 10 ON SELECTI
ON #3":PRINTTAB(30)"SCORE...."INT(G3/10*100)"%":PRINT
160 IF G1<=0 AND G2<=0 AND G3<=0 THEN RETURN
161 IF G1<=0 AND G2<=0 THEN G4=10:GOTO 164 ELSE IF G1<=0 AND G3<
=0 THEN G4=10:GOTO 164
162 IF G2<=0 AND G3<=0 THEN G4=10:GOTO 164
163 IF G1<=0 OR G2<=0 OR G3<=0 THEN G4=20 ELSE G4=30
164 PRINT"THAT MEANS YOU GOT"G1+G2+G3"CORRECT OUT OF"G4"OVER-ALL
.":PRINTTAB(30)"TOTAL SCORE...."INT((G1+G2+G3)/G4*100)"%"
165 PRINT:PRINT"wHEN YOU'VE READ YOUR SCORE, PRESS <ENTER>";:INP
UT I$:CLS:IF (G1+G2+G3=30) OR (G1+G2=20 AND G3=0) OR (G1+G3=20 A
ND G2=0) OR (G2+G3=20 AND G1=0) THEN 178 ELSE IF (G1=10 AND G2+G
3=0) OR (G2=10 AND G1+G3=0) OR (G3=10 AND G1+G2=0) THEN 178
166 PRINT"HERE ARE THE KINDS OF ERRORS YOU MADE, IF ANY:":PRINT
167 PRINTTAB(5)"TYPE OF ERROR                    NUMBER OF ERRORS":P
RINT
168 PRINTTAB(5)"EXCLAMATION MARK (!) .............."W1
169 PRINTTAB(5)"qUOTATION MARK ("CHR$(34)") ................"W2
170 PRINTTAB(5)"APOSTROPHE (') ...................."W3
171 PRINTTAB(5)"QUESTION MARK (?) ................."W4
172 PRINTTAB(5)"COMMA (,) ........................."W5
173 PRINTTAB(5)"PERIOD (.) ........................"W6
174 PRINTTAB(5)"SEMICOLON (;) ....................."W7
175 PRINTTAB(5)"COLON (:) ........................."W8
176 PRINTTAB(5)"HYPHEN (-) ........................"W9
177 WW=W1+W2+W3+W4+W5+W6+W7+W8+W9:PRINTTAB(15)"TOTAL...........
..............."WW:PRINT@960,TAB(5)"WRITE DOWN YOUR SCORE BEFORE
YOU PRESS <ENTER>";:INPUTI$:IFWW>0THEN181
178 CLS:PRINT@448,CHR$(23)"W O W !   N O   E R R O R S ! !":FOR
T=1 TO 50:NEXT T:U=U+1:PRINT@448,CHR$(255):FOR T=1 TO 50:NEXT T:
IFU<=10 THEN 178 ELSE U=0:CLS:R$=""
179 PRINT@320,"WOULD YOU LIKE TO SHOW YOUR RESULTS TO YOUR TEACH
ER?":PRINT@448,"THEN CALL YOUR TEACHER OVER BEFORE YOU TYPE Y (F
OR YES).":PRINT@576,"IF NOT, JUST TYPE N (FOR NO).":R$=INKEY$:IF
 R$="" THEN 179
180 IF R$="Y" THEN CLS:GOTO157 ELSE IF R$<>"N" THEN CLS:PRINT"TY
PE Y OR N, PLEASE...":GOTO179
181 CLS:RETURN
182 REM        ***  GRAPHIC EYES SUBROUTINE  ***
183 E1$=CHR$(156)+CHR$(172)+CHR$(193)+CHR$(156)+CHR$(172)
184 E2$=CHR$(198)
185 E3$=CHR$(140)+CHR$(188)+CHR$(193)+CHR$(140)+CHR$(188)
186 E4$=CHR$(188)+CHR$(140)+CHR$(193)+CHR$(188)+CHR$(140)
187 E5$=CHR$(176)+CHR$(188)+CHR$(193)+CHR$(176)+CHR$(188)
188 E6$=CHR$(188)+CHR$(176)+CHR$(193)+CHR$(188)+CHR$(176)
```

Program continues

THE SWITCH



SWITCH TO 5"/8" DOUBLE DENSITY

LNDoubler 5/8

FEATURES

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

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

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

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

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

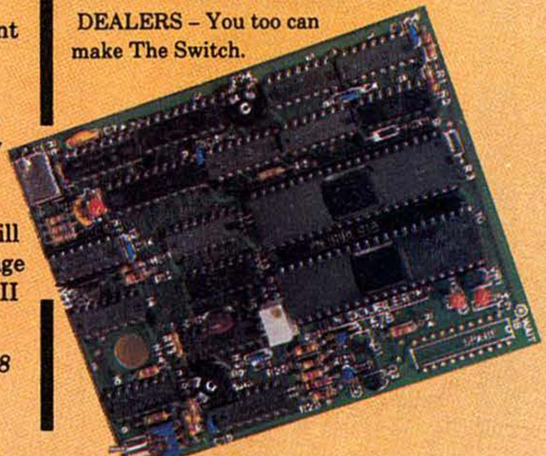
EASY TO INSTALL – The *LNDoubler 5/8* is easy to install. There are no traces to cut, no wiring to do, just a screwdriver and a few minutes of your time is all that is required. The instructions are fully illustrated for all interfaces. In minutes you will be 'up-and-running', and enjoying your computer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

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

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

$219.95

DEALERS – You too can make The Switch.



LNW RESEARCH CORPORATION

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

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

Program continued

```
189 E7$=CHR$(159)+CHR$(175)+CHR$(193)+CHR$(159)+CHR$(175)
19Ø H=RND(3):ON H GOTO 191,192,193
191 PRINT:PRINT"HMMMMMM...NOW LET'S SEE...WHERE DID I PUT THAT O
NE?":GOTO 194
192 PRINT:PRINT"LET ME THINK...WHAT DID I DO WITH THAT ONE?":GOT
O 194
193 PRINT:PRINT"HMMMM...WHERE DID I PUT IT?"
194 FOR T=1 TO 8ØØ:NEXT T
195 HA=RND(3):ON HA GOTO 196,197,198
196 PRINT@64Ø,"HERE?":GOTO 199
197 PRINT@37Ø,"UP HERE?":GOTO 199
198 PRINT@448,"MAYBE HERE?"
199 FOR T=1 TO 8ØØ:NEXT T
2ØØ HB=RND(3):ON HB GOTO 2Ø1,2Ø2,2Ø3
2Ø1 PRINT@734,"HOW ABOUT HERE?":GOTO 2Ø4
2Ø2 PRINT@832,"DOWN HERE?":GOTO 2Ø4
2Ø3 PRINT@685,"OVER HERE?"
2Ø4 FOR T=1 TO 3ØØ:NEXT T:GOSUB 211
2Ø5 CLS:HP=RND(3):ON HP GOTO 2Ø6,2Ø7,2Ø8
2Ø6 HS$=" HEY, i FOUND IT!":GOTO 2Ø9
2Ø7 HS$="GREAT! HERE IT IS!":GOTO 2Ø9
2Ø8 HS$="WOW! i FOUND IT!"
2Ø9 PRINT@46Ø,CHR$(23);HS$:FOR T=1 TO 5Ø:NEXT T:U=U+1:PRINT@46Ø,
STRING$(LEN(HS$),32):FOR T=1 TO 5Ø:NEXT T
21Ø IF U<=1Ø GOTO 2Ø9 ELSE U=Ø:HS$="":CLS:RETURN
211 PRINT@512,E1$:GOSUB 215:PRINT@512,E2$:GOSUB 216:PRINT@916,E1
$:GOSUB 215:PRINT@916,E3$:GOSUB 215:PRINT@916,E1$:GOSUB 215
212 PRINT@916,E4$:GOSUB 215:PRINT@916,E1$:GOSUB 215:PRINT@916,E2
$:GOSUB 216:PRINT@476,E1$:GOSUB 215:PRINT@476,E5$:GOSUB 215:PRIN
T@476,E1$:GOSUB 215:PRINT@475,E2$:GOSUB 216
213 PRINT@476,E1$:GOSUB 215:PRINT@476,E6$:GOSUB 215:PRINT@476,E1
$:GOSUB 215:PRINT@475,E2$:GOSUB 216:PRINT@476,E1$
214 GOSUB 215:PRINT@476,E7$:GOSUB 215:RETURN
215 FOR T=1 TO 5ØØ:NEXT T:RETURN
216 FOR T=1 TO 5Ø:NEXT T:RETURN
217 REM      ****  STUDENT SECTION BEGINS  ****
218 CLS:PRINT@384,"TYPE YOUR FIRST NAME...THEN PRESS <ENTER>":IN
PUTZ$:IF Z$="" THEN 218 ELSE PRINT
219 PRINT@512,"YOUR NAME IS SPELLED "Z$", RIGHT?":PRINT@64Ø,"TYP
E 'Y' (YES)  OR 'N' (NO)";:R$=INKEY$:IF R$="" THEN 219
22Ø IF R$<>"Y" THEN 218 ELSE CLS
221 PRINTTAB(5)"HI, "Z$", wELCOME TO PUNCTUATION PUT-ON!":PRINT
222 PRINTTAB(5)"I'LL PRESENT A STORY, POEM, OR OTHER WRITING FOR
 YOU TO"
223 PRINT"READ.  LOOK CAREFULLY AT THE PUNCTUATION BECAUSE i HAV
E A"
224 PRINT"CHALLENGE FOR YOU.  i'M GOING TO KNOCK OFF A PUNCTUATI
ON MARK."
225 PRINT"YOU'RE GOING TO PUT IT BACK ON!":PRINT
226 PRINTTAB(5)"HERE'S HOW IT WORKS:  FIRST, A SELECTION WILL AP
PEAR"
227
228 PRINT"ON THE SCREEN WITH A BLINKING LIGHT IN PLACE OF ONE OF
 THE"
229 PRINT"PUNCTUATION MARKS.  YOU'LL DECIDE WHICH MARK GOES THER
E AND"
23Ø PRINT"TYPE IT.  IF YOU'RE RIGHT, i'LL PRINT A STAR (*)--USUA
LLY--AND"
231 PRINT"MOVE ON WHEN YOU'RE READY.  IF YOU'RE WRONG, i'LL BLIN
K THE"
232 PRINT"RIGHT MARK ON THE SCREEN.  I'LL REPEAT THIS 1Ø TIMES F
OR EACH"
233 PRINT"SELECTION.":PRINT@96Ø,"PRESS <ENTER> WHEN YOU'RE READY
 FOR MORE INSTRUCTIONS.";:INPUTI$:CLS
234 PRINTTAB(5)"THERE ARE THREE DIFFERENT SELECTIONS.  YOU MAY P
ICK"
235 PRINT"ANY ONE AND REPEAT IT AS OFTEN AS YOU'D LIKE.  i'LL SH
OW YOU"
236 PRINT"YOUR SCORE AFTER EACH SELECTION.  WHEN YOU WANT TO QUI
T, i'LL"
237 PRINT"PRESENT YOUR LATEST SCORES AND OVER-ALL TOTAL.  i'LL A
LSO TELL"
238 PRINT"YOU WHAT KINDS OF ERRORS YOU MADE--IF ANY.":PRINT
239 PRINTTAB(5)"SOME PEOPLE MAY NEED HELP FINDING THE PUNCTUATIO
N MARKS."
24Ø PRINT"NEXT, i'LL SHOW YOU WHERE THEY ARE LOCATED ON YOUR KEY
BOARD."
241 PRINT"IF YOU MAKE AN ERROR AT ANY TIME, i'LL OFFER TO SHOW Y
OU AGAIN.":PRINT
242 PRINTTAB(5)"pUNTUATION PUT-ON CAN'T TELL YOU ANYTHING ABOUT

THE RULES"
243 PRINT"OF PUNCTUATION.  YOU HAVE TO KNOW THAT YOURSELF.  BUT
IT WILL"
244 PRINT"GIVE YOU A CHANCE TO TEST YOUR SKILLS WITH DIFFERENT K
INDS OF"
245 PRINT"WRITING.":PRINT
246 PRINTTAB(2Ø)"READY TO CONTINUE?  PRESS <ENTER>";:INPUT I$:CL
S:GOTO143
```

NEED A PRINTER?

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

SALE

CENTRONICS 739

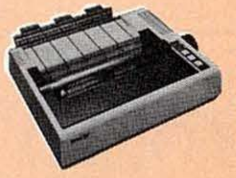
With Graphics and Word Processing Print Quality

• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 80/100 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns • Top of form

CENTRONICS 739-1 (Parallel) .... (List $955) **$Call**
CENTRONICS 739-3 (Serial) .... (List $1045) **$815**
GRAPPLER™ Apple graphics interface .... **$Call**

EPSON MX80/MX70/MX100

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

We also carry a full line of Epson Accessories.

EPSON MX80 (& MX80FT) .... (List $645) **$Call**
EPSON MX70 Dot graphics, 5 x 7 matrix .... (List $450) **$Call**
EPSON MX100 wide carriage, graphics .... (List $945) **$Call**

GRAPPLER™ Apple Graphics Interface .... **$ Call**

GRAFTRAX 80 - MX80 Dot Graphics .... **$ 95**

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 9501/9500 .... (List $1650) **$ Call**
GRAPPLER™ Apple graphics Interface .... **$ Call**

NEW

NEC 8023 DOT MATRIX

Dot graphics • 100 CPS • Bi-directional, logic-seeking • Tractors & friction feed • 5-Alphabet fonts • 8 character sizes • Proportional spacing

NEC 8023 DOT MATRIX .... (List $795) **$ Call**

C. ITOH STARWRITER

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 TEC.

C. ITOH STARWRITER .... (List $1895) **$1525**

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing.

7710 RO, Serial .... (List $3055) **$2575**
7730 RO, Parallel .... (List $3055) **$2575**

TELEVIDEO CRT'S

AT DISCOUNT PRICES!

TVI 910
TVI 912C
TVI 920C
TVI 950

Please Call Toll Free — Prices are too low to advertise

QUANTITY PRICING AVAILABLE

SUPER PRICES

IDS PAPER TIGERS

Dot Resolution Graphics, quality print, speed

9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking • 8 character sizes; 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification

IDS 460G 9 wire printhead, graphics .... (List $1094) **$Call**
IDS 560G wide carriage, graphics .... (List $1394) **$Call**
IDS PRISM Color Graphics Printer .... (List $1995) **$Call**
GRAPPLER™ Apple graphics interface .... **$Call**

CRT'S

ADDS VIEWPOINT .... (List $ 699) **$ 600**

PRINTERS

MALIBU 200 wide carriage, graphics, letter quality (List $2995) **$ Call**
QUME SPRINT 9 .... **$ Call**
DIABLO 630 .... **$ Call**
ANACOM 150 .... (List $1350) **$ Call**

INTERFACE EQUIPMENT

EPSON ACCESSORIES .... **$ Call**

TRS-80 CABLES to keyboard or Exp. interface .... **$ Call**
NOVATION D-CAT direct connect modem .... **$ 180**
HAYES MICROMODEM (Apple II) .... **$ Call**

CALL FOR INFORMATION & CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

RETAIL PRINTER STORES:

Store #1	3150 E. La Palma, #I, Anaheim, CA	(714) 630-3622
Store #2	13604 Ventura Bl., Sherman Oaks, CA	(213) 501-3486

Store Hours: M-F 10-6 Sat. 10-4

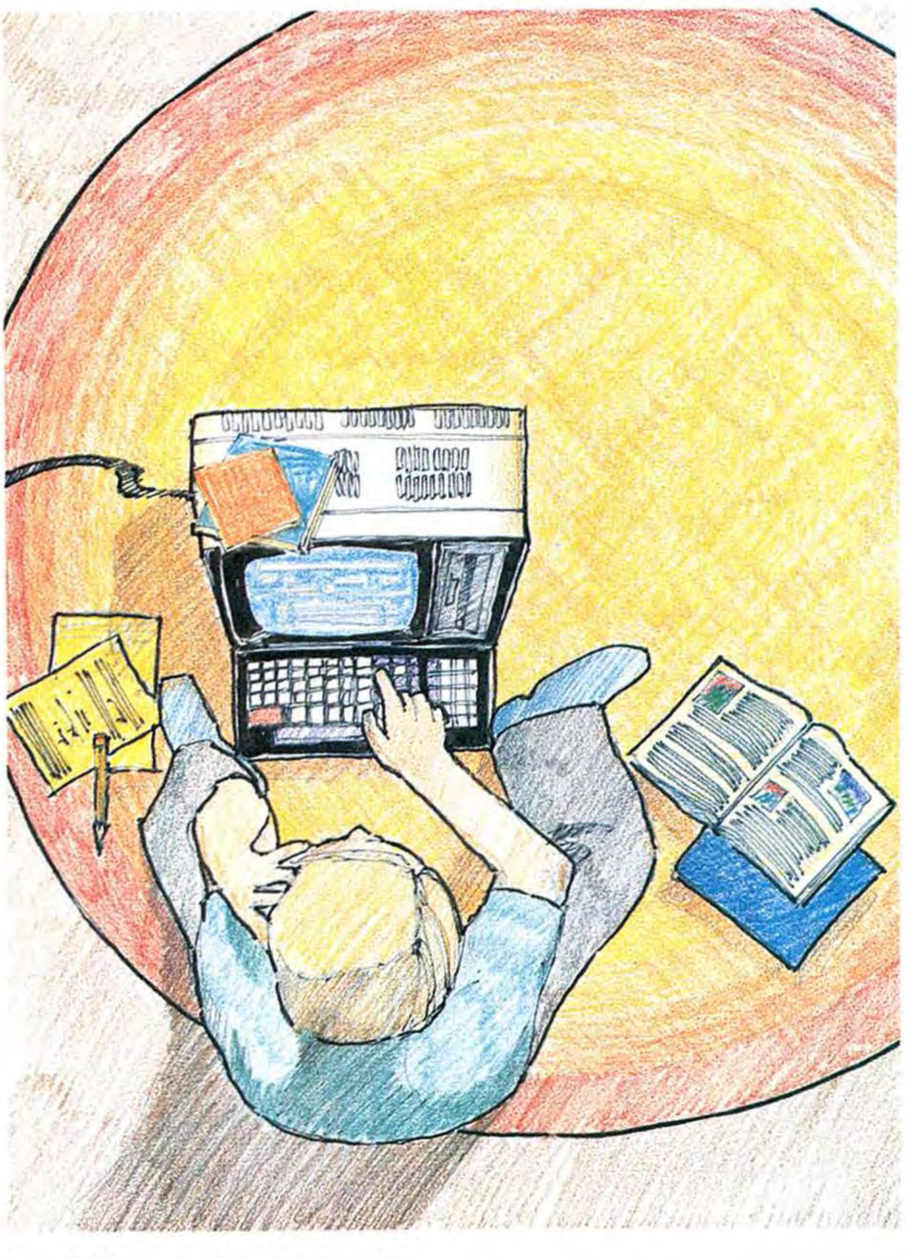
Orange Micro inc.
3150 E. La Palma, Suite G, Anaheim, CA 92806
"The Computer Printer Specialists"

Phone order WELCOME; same day shipment. Free use of VISA & MASTERCARD. COD's accepted. Manufacturer's warranty included on all equipment. Prices subject to revision.

A look at the future—education and computers.

The Classroom Crystal Ball

James Edward Keogh
121 Gordon Street
Ridgefield Park, NJ 07660



Most school systems throughout the country, although they are in serious need of funds, recognize the power microcomputers have as an educational tool. Lacking carefully designed instructions—software—computers can become a center of frustration.

The future for new educational programs is bright and will probably be an area of fastest growth for software in the next few years. Many teachers and programmers will turn their thoughts to designing new course software for their students. Before embarking on a program design that will revolutionize teaching methods it is important these creative minds first consider a few aspects of the typical classroom.

This article will discuss many aspects of the classroom environment, the needs of students and teachers, and ways the microcomputer will affect them. The educational software designer must take into consideration both the physical and emotional needs of the student. The programmer must also have a teacher's understanding of the material the program is to present, as well as skill to communicate to and motivate the student. In short, the programmer must wear several hats—programmer, educator and expert in the material being presented.

"How computers dehumanize society is an emotion-packed subject. . ."

The Dehumanized Classroom

When designing a computer program for the classroom, the programmer must seriously confront the issue of learning without teachers. How computers dehumanize society is an emotion-packed subject that causes lengthy discussion. We are only a few years away from 1984: The rapid development of the microprocessor and microcomputer may prove George Orwell's projections to be fairly realistic.

The issues are not merely theoretical problems posed by educators, but real obstacles that must be overcome if students and teachers are to truly benefit from the classroom computer. It is important the educational programmer realize the computer is not *replacing* the teacher, but is only *assisting* the teacher to educate the student.

Educational software is often designed so the microcomputer may take over many of the teacher's traditional roles. The computer displays the major points of the lesson and quizzes the student until correct answers are given.

When the lesson has ended—determined by the computer—the student continues to another programmed lesson.

Although it may appear that teachers will be required only to take attendance and monitor student transitions, this vision (or nightmare) is false: There will still be teachers teaching. Microcomputers are limited, and the need for adaptable, responsive teachers will always be present.

Targeting The Software Market

by James Edward Keogh

A programmer's first step towards selling software is to identify an unfulfilled need of microcomputer owners. This can be determined by researching micro owners to discover what they are using their microcomputer for.

To gather this information try contacting various microcomputer manufacturers or companies which sell micros to obtain their marketing and statistical surveys.

If you are more interested in educational software research the various ways school systems are using the computer. What grades is the computer most used in? And in what areas is software most needed? This information can alert the programmer to types of software the school systems are likely to purchase.

Once you have a good idea of the type of programs you would like to write, survey the marketplace, search computer magazines, stores, and major software houses for similarly-titled software packages available on the market. There is no point in putting a lot of time and effort into a program design only to find the market is glutted with similar packages.

Your next step is to determine the quality of any existing programs. Review your competition's products and estimate their strengths and weaknesses. Your product should eliminate many of the weaknesses and incorporate all the strengths of the competition's products.

By following these steps closely you will determine whether your software can compete in the market place, or out-perform those already there. Your next step is to sell it.

At this writing there are few uniform steps to follow to sell software. There are four methods used: computer stores, magazine advertisements/mail order, direct mail, and software houses. Computer stores do exist, but the concept is still in a growth cycle. They do, however, offer the software producer an excellent potential for a sale. One benefit is the sales staff can answer any questions the computer owner may have and direct his or her attention to your software. If the product is a good seller, the programmer will have to be concerned with bulk orders.

Although computer stores may be the ideal outlet for you, you are not the ideal source for the store owner. The store owner who is used to dealing with companies such as Apple and Commodore will want a professional staff to support the products he is selling, as well as a complete line of software.

Advertising in computer related magazines also offers the programmer an outlet to sell software. Although advertising rates may seem very expensive you must realize the number of readers reached by the magazine. You may actually be paying only a few cents per reader for the advertising space.

One of the drawbacks of advertising in a magazine is that your ad will be mingled with more than 100 pages of articles and other ads. The placement of the ad within the magazine is very important.

The most expensive and the highest-risk method of selling software is by direct mail, sending your sales literature directly to the consumer. Direct mail can cost more than 60 cents per address with a return of only one or two percent. Of a hundred letters mailed, two letters, or orders, will return. Remember, those two orders must pay for the cost of the mailing plus a profit.

An additional approach you might explore in an attempt to market your wares is to contact a major distributor of computer software or a software house. These firms purchase software from independent programmers and package them into a marketable product. They have a trained, full-time sales staff who assure the products get to market.

These distributors usually duplicate your cassettes or disks at their expense and pay a royalty for each one sold. This method may prove to be the least expensive and most profitable unless you have top business skills and contacts in the industry.

One last method you might consider is to develop a complete product line and contact a major firm who is currently selling products to your market, for example, an educational publisher. If your products are top quality, the firm may want to purchase your complete line or agree to distribute the line for you.

Creating and selling software is by no means easy but it is possible. If you have a noteworthy program you can compete in this growing market. ■

"Most microcomputer programs are written for users who have basic capabilities—the ability to read and spell. . ."

The Student

The prime objective of an educational program is to present material in a logical and interesting way, which can be easily absorbed. This may appear to be an easy task. However, there are a few steps which must be taken when designing educational software which complicate matters.

Most microcomputer programs are written for users who have basic capabilities—the ability to read and spell, for example. The educational programmer, however, does not have this freedom. Software designed for use in the classroom must be directed at a specific age group with specific capabilities.

We all would be able to guide a first or second grader through a lesson, but the programmer must translate an oral lesson into an understandable program—often for a student who may not yet be able to read or write. Although the programmer may be very facile with written communications directed at adults, he or she must be extremely careful when writing for young students.

Microcomputers pose an additional problem for the young student: The letter and number display on a microcomputer screen differs from the formation of letters and numbers with which they are familiar. For example, the numerals three and four can be presented in several ways. Most adults can identify these numbers in almost any form, but most young students can recognize a number only in the single form they are familiar with.

Programmers must assure the teacher that the student can read and understand the characters presented on the display screen. Research and creative thought must be given before deciding how the characters should appear. The first step is for the programmer to review printed material designed for the same age group as the program which the programmer is working on.

Educational publishers spend time and money researching this and have arrived at a few solid guidelines. The printed material you review will give you a good indication of character design. The programmer's objective is to project characters on the screen that match those in the printed material.

Programs designed for students in the upper grades will probably use characters which are directly generated by the microcomputer. So the programmer will have little control over the appearance of these characters. Usually students in the upper grades have assimilated the various character constructions, so microcomputer-generated characters will give him little difficulty.

Young children, however, will rely on the creative skills of the programmer. Normally, computer-generated characters are too small for these students to read. The programmer must enlarge the characters by "painting" the character on the screen. For example, a math addition problem will require the programmer to "fill" the screen with the entire problem.

In these circumstances the programmer will design the character to conform with those in educational printed material.

Very few computers can be programmed to alter character formation. Teachers who have tested educational programs in the classroom have pointed out that the size of the displayed characters is important. Many students have found it difficult to recognize the small letters and numbers displayed by the computer. They have a greater chance of recognizing an enlarged character.

Aside from the mechanical difficulties involved in programming for the classroom, the programmer must also have a feeling for how the student views the new experience of using computers. Many students will associate the new learning tool with computer games rather than a text book.

Because the computer is new to the student and is associated with fun and entertainment, the programmer has a captive audience—at least for a moment. Techniques developed for computer games can be combined with the material to be learned and designed into an entertaining educational program.

Students are willing to learn as long as the material presented to them is interesting and presented in an intelligent manner. The material should encourage the student to think and reason within his or her own capabilities. For example, a student in the younger grades may find computer animation an acceptable learning form. Older students may be insulted by such childishness and demand more involved graphics and higher difficulty levels in educational games. As many teachers learn quickly—never doubt the student's capabilities. Give the student a little more than he or she is ready to handle.

Encouragement is a Factor

Encouragement is a strong motivating factor in education. Students need someone to say "You've done a good job" from time to time. For a student, praise from a teacher seems more valuable than that from a computer.

The educational programmer must realize the computer is not replacing the teacher, but assisting the teacher in educating the student.

Teachers and students have apprehensions about computerized teaching. At a recent state teachers' convention a newspaper reported a teachers' union leader warning his members: "Soon you will be replaced by a computer."

Such a remark was not made in jest and articulates one of the first visible fears which a few teachers have developed.

Is this fear justified? No. The educational microcomputer programmers and educators will determine how the microcomputer will be used in the classroom. These fears may reflect an uneasiness with the unknown since the microcomputers are just beginning to enter the classroom. The programmer who is designing educational software must consider his leadership role in changing the attitudes of educators.

Students too will have their apprehensions about microcomputers although they should more readily adjust to this development. The student's biggest fear is the ease of interfacing with the microcomputer and the software.

Students are familiar with books. Once the book is open the student immediately enters the lesson. With a microcomputer, the student must master the mechanics—load the tape or disk, find his or her way through a typewriter keyboard, read letters and numbers from a television screen, and then learn the lesson. In the back of the minds of many students is the fear of breaking the microcomputer.

There is no easy solution for the programmer. He can only do his job with 200 percent accuracy to make the educational community comfortable with the reliability of educational software.

The Teacher

Before any school administration agrees to introduce a microcomputer into the classroom, they must feel certain the teaching staff is comfortable with the software systems. Like students, teachers will approach microcomputer technology with various preconceptions and expectations.

The teacher is the educational expert to whom the educational programmer must prove the usefulness of his or her software package.

". . . it is difficult to describe the difference between entertainment and learning material."

Educational programs will be judged by school administrators and teachers by educational standards—not programming standards. The program must be challenging for students and meet their educational needs. The program must be able to coexist with current learning methods and text books.

Educators will be looking for programs which are simple to use and understand, requiring little or no active participation from the teacher. Ideally, students could use the software package on an individual basis, permitting the teacher to spend more time with those students who require closer attention.

The primary objective of educational software is to help students learn. Teachers will not respond well to programs that are too entertaining and have little educational merit. Educational experts agree that it is difficult to describe the difference between entertainment and learning material. Many of us have grown up learning from the entertainment media. Somehow, the educational programmer must sift through these vague terms and come up with a product that will meet everyone's expectations.

The Program

Before designing a program, the programmer must anticipate every possible problem that might arise during a learning session, and develop contingencies for them.

All ambiguous words and phrases must be removed; educators have found throughout the years that students will ask teachers to clarify questions during a test. The computer can respond only to input considered by the programmer.

The computer's inability to communicate intelligently with the student is not the only problem which must be addressed. Unless the software is very carefully assembled and debugged, the student and teacher can be left very frustrated.

Documentation is important for any software package. Each package should contain two professionally written and printed instruction sets. One booklet should be directed to the student and a separate booklet directed to the teacher. Learning objectives must be clearly stated in the teacher's booklet, with examples of how the student is to use the package. The student's booklet need only inform the student—using vocabulary suited to the student—how to use the program.

Researching Your Material

A good place to start researching your material is by going to your local library and reviewing a few books designed for the same group as your proposed program.

Another tip is to visit a specialized library that deals primarily with teachers located at universities or teachers' colleges, to give you a better understanding of the subjects taught at each grade level and help you develop an understanding of the methods used to teach students. Also, review several of the educational series published by major publishing houses; these series contain books designed to completely educate a child from grades one through eight.

If you are not close to such a library, you may be able to obtain the same information by contacting your local school district.

The programmer can control the size and the formation of letters and numbers. For example, on the TRS-80 characters on the screen can be enlarged by pressing two keys. Another way to enlarge the characters is by "drawing" the letters and numbers on the screen. This is a lengthy process and may require more than 16K of memory, but the results are worth the effort.

Drawing characters is only suited for programs which require a limited display on the video screen, such as drills in addition and subtraction.

The program length and the amount of loading time is another factor to be considered. Remember—the load time of a program shortens the student's lesson time.

Testing the Software

When it comes to marketing software the only judgement that matters is the buyer's. The software producer must convince the potential customer of the quality of the package. Consider for a moment some of those mail order programs you have purchased that just did not live up to their advertising. Make sure your programs perform and are thoroughly debugged and tested.

Before marketing your software thoroughly test it in a simulated classroom environment. Such tests provide insight and useful student feedback.

A Look into the Future

In the 21st century microcomputers may be preparing individually designed tests based on each students educational profile. The computer would correct the test, grade the student, and present the teacher with an outline of the student's strengths and weaknesses. The teacher would then interpret this information and prepare an educational plan for that student. The only fragment that makes this scenario incomplete for today's classroom is the lack of creative educational software. ■

Reader Service for facing page ✓383→

CLEAR. QUICK. QUIET. ALL THREE, ONLY $1,095.*

You get sharp, easy-to-read printouts. You get them fast, over 150 characters per second, from a printer that's loaded with convenience features.

The Heath/Zenith 25 Printer is a heavy-duty, high-speed, dot matrix printer. It produces up to 300 lines per minute with whisper-quiet smoothness. The entire 95-character ASCII set prints in upper case and lower case with descenders, in a 9 x 9 matrix. All functions and timing are microprocessor-controlled.

The features described below tell only part of the story. You have to see it in action to know how good it really is.

Pick the store nearest you from the list at left. And stop in today for a demonstration of the Heath/Zenith 25 Printer. If you can't get to a store, send $1.00 for the new Zenith Data Systems Catalog of assembled commercial computers and also receive free the latest Heathkit Catalog. Write Heath Co., Dept. 035-864, Benton Harbor, MI 49022.

HEATH/ZENITH

Your strong partner

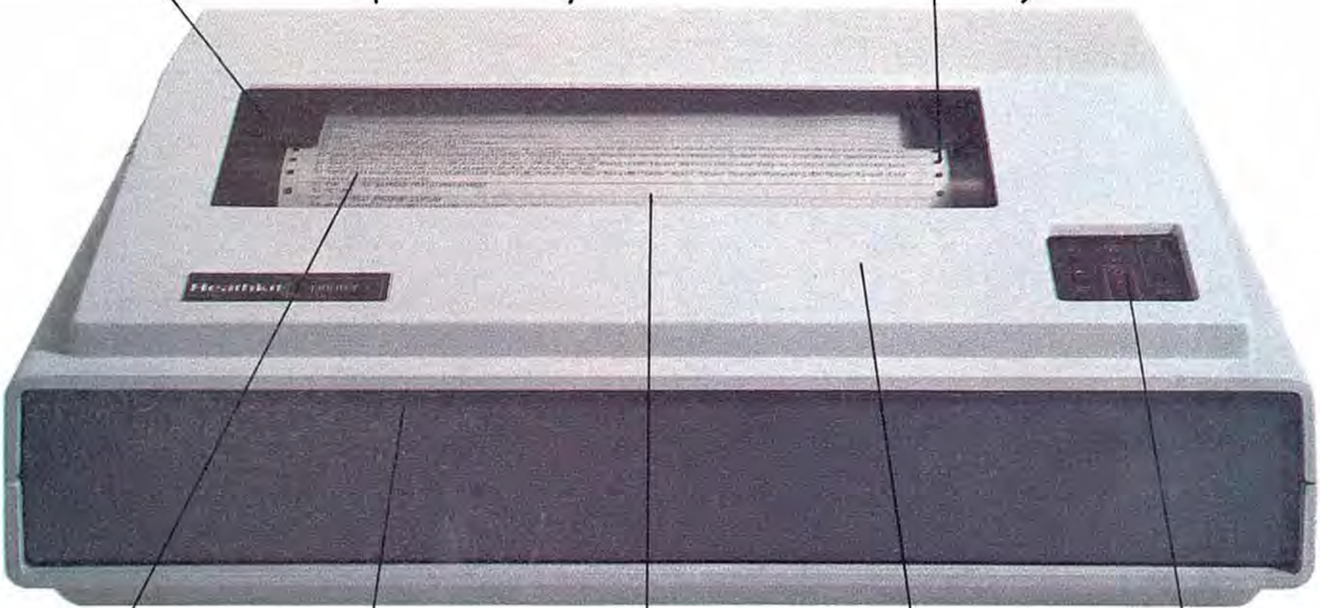
Adjustable tractor-feed width with dual sets of tractors for smooth, bi-directional paper movement. Adjustable vertical and horizontal tabs.

Character pitch is hardware or software-selectable at 10, 12, 13.2 and 16.5 characters per inch, for a maximum of 222 characters per line. That gives you great flexibility in setting up forms.

Standard RS-232C interfacing for compatibility with most systems. Also 20mA current loop serial interface.

Uses standard edge-punched papers in single or multiple forms or fanfold.

Software- or hardware-selectable baud rates at 110, 150, 300, 600, 1200, 4800 and 9600.



Character set includes 33 block graphic characters for charts and graphs.

Heavy-duty construction for reliable operation and long life under daily use.

Convenient cartridge ribbon for quick, no-mess replacement.

Completely enclosed cabinet muffles sound for quiet operation.

Special detectors tell you when you're out of paper or when paper jams.

*In kit form, F.O.B. Benton Harbor, MI. Also available completely assembled and tested at $1,595. Prices and specifications are subject to change without notice.

CP-204A

This program makes keeping track of students easy.

Roll Call!

Michael M. T. Henderson
Department of Linguistics
University of Kansas
Lawrence, KS 66045

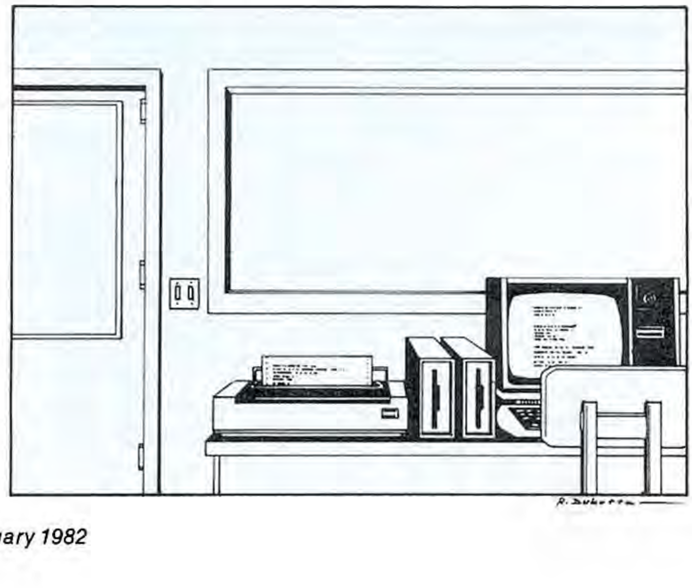
Microcomputers are amazing machines. If used properly, they save you time, money and trouble. But the road to efficient use of a micro is not yet smooth. This is the story of one attempt to use a TRS-80 to manage student records; a story instructive to those contemplating a desktop computer for similar uses.

For several years, I was the director of an English program for foreign students. Our student body grew so large that we had to change the way we kept records. We were in danger of losing track of students, and our secretaries were drowning in paperwork. During the first three weeks of every semester students move from class to class, necessitating retyping two class rosters for every move made—and with the names our students have, this was painful and error-prone work. I dreamed of typing each student's name only once a semester, and changing records by entering only the new data. We also wanted more statistical analysis than we had time to do by hand.

We began using our campus' academic computer, a Honeywell mainframe. I wrote the programs. A semester later, we were very impressed with the productivity increase, but we were dissatisfied with the slow phone line communication, and with the rather primitive Basic supported by the Honeywell (far less sophisticated than the Basic supplied with the TRS-80 in ROM). We needed our own in-house computer system; a micro would cost less than one year of connect fees to the Honeywell.

We decided to buy a TRS-80 for two reasons. I already had one, so I knew its capabilities and could develop the software at home. The software support for the TRS-80 is greater than that of its competitors, especially in data base management. I would write the software myself, since our needs were quite specialized; we needed, for instance, 25 fields in each student record. At the time no commercial DBMS system offered that many to micro users.

Eventually the great day came when our two disk, 48K RAM system with RS232 and a Line Printer I arrived from the independent Shack dealer who submitted the lowest bid. I eagerly unpacked everything, pleased as a kid at Christmas. My first impressions were not good. The printer ribbon had spilled all over the inside of the crate, and had to be replaced.



The RS232 interface did not work at all: It was attached by only one screw. A trip to the drawer of tiny screws in my basement took care of that. The printer would not line up columns. Warranty service fixed that problem. The printer was returned with a testy note to the effect that we should not expect nice column alignment with a worn-out drive belt. I thought of telling them the thing was a brand-new printer, but decided it was not worth the postage.

In my innocence, I thought all disk drives were the same. After all, Tandy got theirs from Shugart, and my two Shugart SA-400s from Apparat performed beautifully. The two drives with the Radio Shack labels on the front were very, very different. One, a Shugart 35-track drive, would boot a disk most but not all of the time; the other, a Tandon 40-track drive (as I found out a year later), sounded like a garbage disposal trying to digest a beer can pop-top, could not turn a constant 300 rpm, and produced parity errors much of the time. The drives also went in for warranty work,

The Key Box

Basic Level II
Model I
48K RAM
NEWDOS80.2
2 Disk Drives
Line Printer IV

along with the expansion interface (in case it was causing the problems). The drives came back in a week, aligned and trimmed, and the interface came a week later, with a new disk controller chip and a couple of new RAM chips. My own interface, with cheap RAM from somewhere in California and the old buffered cable, has consistently worked better and faster than the one with name-brand RAM that came factory-sealed from the Shack. I had based my decision to purchase a TRS-80 for the office largely on the fact that my own system worked so well. I did not know then that the TRS-80 is a great machine inside the keyboard unit, but for reliability, buy your peripherals elsewhere. The office system had the new keyboard with numeric pad, no keybounce and a much nicer monitor than my old 1978 model. That at least was an improvement.

The warranty service did not take care of our hardware problems. The printer has worked well since we installed the Service Technologies motor control so the motor does not run all the time. The disk drives have never worked satisfactorily. They continue to give too many parity errors for anyone but a computer hacker. I exchanged them for my ancient, reliable Shugarts, which now do four hours of work a day at the office with no problems. Unexplained glitches still show up from time to time. Power is rather erratic in the Great Plains, probably because of the high winds, and sometimes it stops altogether. We tried a Mayday, but it did not work. A Radio Shack Line Filter has cleaned up the power considerably, but just the other day the system hung up on me, and not even the reset button would release if from Silent Death. Liberal applications of a pink eraser to the contacts between the keyboard cable and the expansion interface helped contain this problem.

I have gone into some detail about hardware problems; people contemplating the purchase of a desktop computer should know that you cannot plug them in and forget them. Perhaps the new generation of one-piece computers will perform better in this regard.

We could not use the operating system that came with the machine, TRSDOS. I needed a system that could perform several functions automatically at power-up, such as enter Basic with a defined memory size and execute a menu program. NEWDOS Plus got the system going automatically, and apparently fixed all the errors in TRSDOS.

Even paring NEWDOS Plus down to the barest essentials, we soon ran out of disk space because of the complexity and variety of the programs we needed. Just in time Percom came out with the Doubler, Apparat with NEWDOS80 and Circle-J with Doublezap. The new file formats available with NEWDOS80 have not worked, probably because we are trying to use them with Doublezap. I have high hopes for them with NEWDOS80 version 2.0, but it is still incompatible with Bionic Basic, a utility from Apparat which has cut our processing time for class rosters by two thirds. I expect that Apparat will resolve the incompatibility quickly; they say they are working on it.

I recommend the following software products to anyone using a TRS-80 for desktop computing, whether for data base management, word processing, or other business uses: NEWDOS80 (the best DOS so far); Bionic Basic (includes many time-saving utilities); Keyedit or Keyplus (saves the programmers hours of frustration); DDT/BAS (vitally necessary—use every three months); BOSS/CMD (tremendously useful for debugging and variable tracing); and ST80D (has always worked beautifully in transferring files to or from our mainframe).

I expect the Model III will overcome many of the hardware problems, but I would still order disk drives from some vendor other than Radio Shack. You do not have to restrict yourself to an all-uppercase printer any more; I have excellent results from the Line Printer IV at home, and I have heard good things about the IDS Paper Tiger. A green screen on the office computer has cut down significantly on eyestrain.

What lessons can be learned from my experience? Computers can significantly increase office productivity. We now enter each student's name and test data only once; we use an edit utility to update only those fields that have changed. The machine makes the alphabetized class rosters every week. At the end of the semester, we enter the final scores and grades, and the computer prints English proficiency reports for each student.

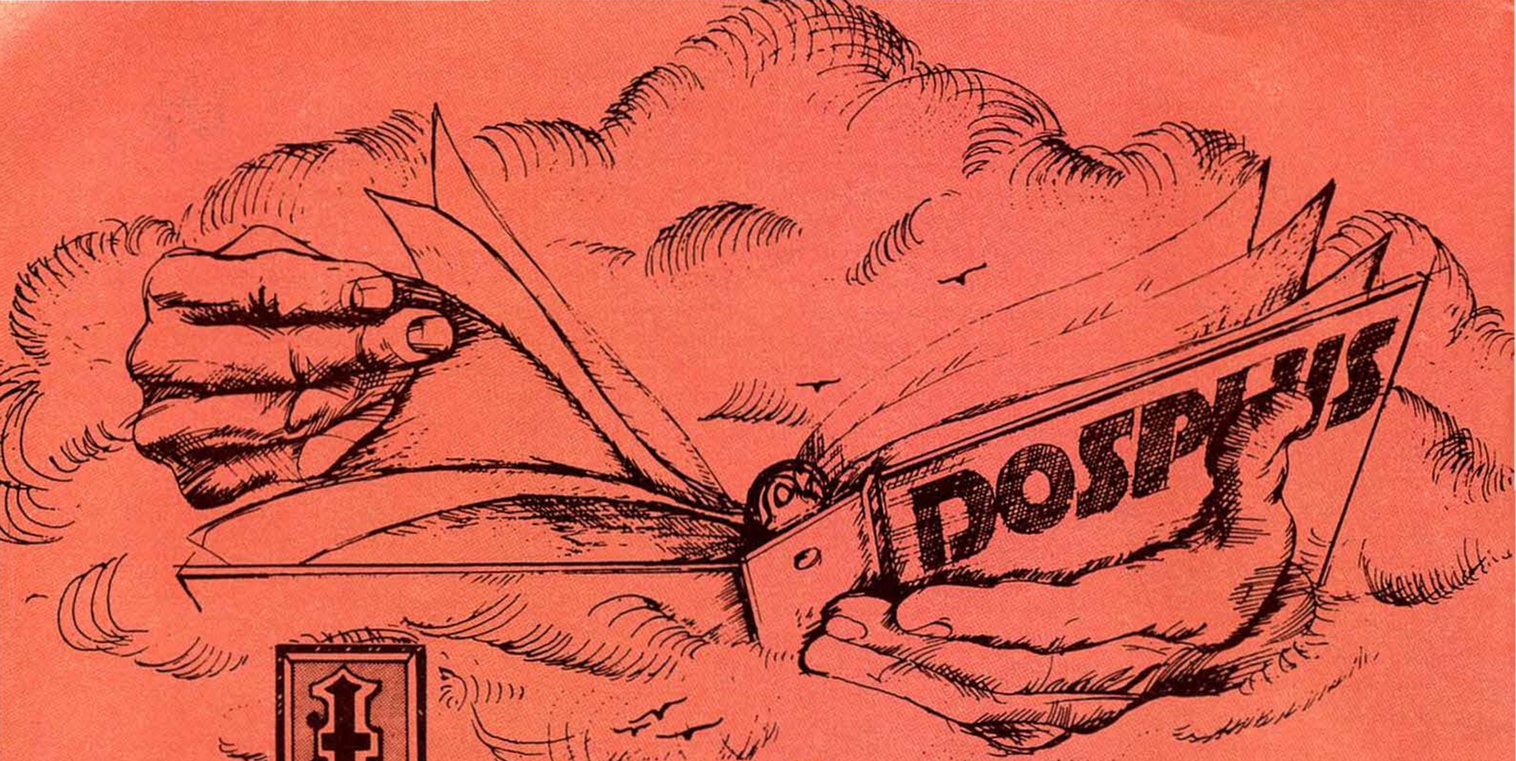
Developing a good system takes time. Do not expect to be up and running for several months after you first acquire the machines. Use your index cards and shoeboxes for at least one term or business cycle even after you are positive that everything is working. But don't give up: Persistence pays off.

Somebody in the office must understand what computers can and cannot do. If it is the boss, as in my case, so much the better for overriding the doubting Thomases when they see their first inaccurate printouts. Even more importantly, the person who programs the system must know why he or she is doing it. Many computerized operations fail because the programmers never understood the reasons for the programs they were writing, and none of the users could communicate with them.

Finally, despite all the problems, we have an excellent system which has made it possible for our staff to concentrate on improving instruction instead of just keeping track of student records. It has been worth it, and it has been fun for me. Perhaps that is the key: Somebody has to enjoy playing with the computer, or the whole project will end with the hardware gathering dust in a storeroom. ■

Michael Henderson is Associate Professor of Linguistics at the University of Kansas. He has been playing with TRS-80s since 1978.

✓See List of Advertisers on page 290


DOSPLUS

See List of Advertisers on page 290

You'll love this program at exam time.

Put Them to the Test

Barry Davis
24 Parson Drive
Stony Brook, NY 11790

Test writing is among the most tedious jobs teachers have. Over the years I have had to write thousands of test questions, so I finally wrote a program to ease the burden. My students are not as pleased about it as I am—it has led to my giving more tests because they are so easy to create.

A single 40-track disk holds up to 350 questions in single density or 650-700 in double density. With modifications the program will run on any micro configuration. The program is menu-driven and self-prompting.

Five modules carry out all the functions for storing questions or writing a test.

Module I—Question Entry

This brief section (lines 100-150) is deceptively simple. Most work is done in a subroutine at line 730. All you do is type in a question and a set of multiple-choice answers: How many angels can dance on the head of a pin? a) one; b) two; c) a thousand; d) a million; e) none of these. The right parenthesis,), must be used to set off each choice; it is used as a marker later in the program. The number of choices can vary from question to question.

The program jumps to the edit subroutine shared with the editing module. In this subroutine a question can be edited or replaced before being stored. The subroutine first checks the question length and reports if the 250 byte limit is exceeded, offers a menu to cancel the edit mode and return to the main menu without storing the question, change the question just typed, cancel the question and enter another in its place, and cancel the changes just made and return to the original for reediting and for storing the question.

Line 820 asks for the word or phrase to be modified. Line 830 finds the start of the phrase or word to be replaced. It also notifies you if the target phrase does not exist in the original question. Line 840 asks for the new (replacement) word or phrase. Line 850 divides the question into three parts. The first part includes everything in the question up to the beginning of the word or phrase to be modified. The second portion is the target, and the third part includes everything from the end of the target word or phrase to the end of the question. Line 860 concatenates the pieces into a single, corrected version of the question and sends the program back to the editing section for further changes or for storage.

As an example we will use the question: How may angels can dance on the head of a pin? a) one; b) two; c) a hundred; d) a thousand; e) none of these. We will change the word *dance* to *sit*. When we answer the edit menu with choice two, it prompts "Enter word or phrase to be replaced." Answer with *dance*. The next prompt is "Type the corrected version." Answer with *sit*. The computer then creates the strings "How many angels can", "sit", and "on the head of a pin? a) one; b) two; c) a hundred; d) a thousand; e) none of these."

Finally, the program concatenates the pieces into a single question and displays it for further editing. The length of the target string and its replacement need not be the same.

To insert new words so the question reads "How many angels or other nonphysical beings can dance on the head of a pin?" answer the edit prompt with "angels can" as the target phrase. The replacement phrase is "angels or other nonphysical beings can." The reconstruction is the same as in changing a word.

Module II—Edit Question From File

This section (lines 160-220) asks for the file number of the question to be edited. The computer gets the question and passes control to the editing

The Key Box

Disk Basic
Model I or III
32K RAM
TRSDOS, NEWDOS or DOSPLUS
1 Disk Drive
Paper Tiger Printer

Program Listing 1

```
10 '
                         TESTMAKER PROGRAM
                         COPYRIGHT, 1981, BARRY DAVIS
                         24 PARSON DR, STONY BROOK, NY 11790

20 CLEAR5000:
  OPEN"R",1,"TESTFILE":FIELD1,255 AS Q$

30 CLS:PRINTTAB(23)"TEST MAKER PROGRAM":PRINT:PRINT

40 PRINTTAB(20)"1. Enter new questions":
   PRINT:PRINTTAB(20)"2. Edit a question in the file":
   PRINT:PRINTTAB(20)"3. Print questions sequentially":
```

Program continues

Program continued

```
     PRINT:PRINTTAB(20)"4. Print a test":PRINT
50 PRINTTAB(20)"5. Remove 'USED' marker":
     PRINT:PRINTTAB(20)"6. End this run"
60 GOSUB930;:INPUTMD
70 IF MD<1ORMD>6THENGOSUB690:GOTO30
80 IF MD=6THENCLS:PRINT@468,CHR$(23)"RUN ENDED":CLOSE:PRINT:PRINT:PRINT:END
90 ON MD GOTO 110,210,240,340,610
100 '
                        QUESTION ENTRY
110 CLS:PRINTTAB(18)"Adding Questions to File":PRINT
120 QN=LOF(1)+1
130 PRINT"Enter Question ('END' to stop)"
140 PRINT:PRINT:LINEINPUTQQ$:GOSUB900:IFQQ$="END"THEN30
150 Q2$=QQ$:GOSUB740:GOTO110
160 '
                        EDIT QUESTION FROM FILE
170 CLS:PRINTTAB(25)"File Editor":PRINT:PRINT
180 PRINTTAB(20)"1. Edit question":
     PRINT:PRINTTAB(20)"2. Return to main menu":
     PRINT:PRINT:PRINT:PRINT
190 GOSUB940;:INPUTM1:IFM1<1ORM1>2THENGOSUB690:GOTO170
200 IF M1=2THEN30
210 CLS:PRINTTAB(23)"Editing Question":PRINT:
     INPUT"What is the # of the question to be edited ('0' to return to men
u)";QN:
     IF QN=0 THEN 30 ELSE IF QN>LOF(1)THENGOSUB690:GOTO210
220 GET1,QN:QQ$=Q$:P=INSTR(QQ$,"   "):
     QQ$=LEFT$(QQ$,P-1):Q2$=QQ$:GOSUB740:GOTO170
230 '
                   PRINTS OUT ENTIRE FILE
240 CLS:PRINTTAB(20)"Prints Entire File":PRINT:
     INPUT"At what question shall the printout start";SN:
```

Program continues

routine. The rest of this section operates as Module I.

If you do not know the question's file number, you can get it by calling for Module III.

Module III—
Prints File Sequentially

Module III (lines 230–320) calls the questions in order, starting at any number you choose, and goes to the end of the file or to any other number you select.

The subroutine, starting at line 950, breaks the question into lines of print. It will not break up a word or separate a letter from the choice it designates.

In line 960 the initial length of the first print line is set at 63 bytes. A five byte answer line, STRING$(5,95), and a question number is added later to the first line. These additions bring the first print line to about 70 bytes. Line 970 checks the last byte of the first print line. If the last byte is not blank, the line length is increased by one byte and is checked again. This process continues until a blank is found

Program continued

```
     PRINT:IFSN<1ORSN>LOF(1)THENGOSUB690:GOTO240

250 INPUT"At what question should the printout end ('E' for end of file)";C
N$:
     GOSUB910

260 IF CN$="E"THENCN=LOF(1)ELSECN=VAL(CN$)

270 IF CN>LOF(1)ORCN<SNTHENGOSUB690:GOTO250

280 CLS:PRINT@456,CHR$(23)"FILE BEING PRINTED"

290 LPRINTCHR$(01)"Sequential Printout - Questions ";SN;" to ";CN;CHR$(02):

     LPRINT"":LC=LC+2

300 FOR I=SN TO CN:GET1,I:QQ$=Q$:P=INSTR(QQ$,"   "):
     QQ$=LEFT$(QQ$,P-1):GOSUB960:
     NU$=STR$(I)+". ":LPRINTTAB(5)NU$;L1$:
     LPRINTTAB(5)L2$:LC=LC+2:IFL3$<>""THENLPRINTTAB(5)L3$:
     LC=LC+1

310 IFL4$<>""THENLPRINTTAB(5)L4$:LC=LC+1

320 GOSUB710:L1$="":L2$="":L3$="":L4$="":
     LPRINT"":LC=LC+1:NEXTI:GOTO30

330 '
                        TEST PRINTER

340 CLS:PRINTTAB(24)"TEST PRINTER":PRINT

350 PRINT@448,"COURSE  NAME  ";:LINEINPUTC$:PRINT@448,STRING$(64," ")

360 PRINT@448,"POINT VALUE OF EACH QUESTION ";:INPUTV:PRINT@448,STRING$(64,
" ")

370 PRINT@448,"AT WHAT QUESTION SHALL I BEGIN THE SEARCH ";:
     INPUTSN:PRINT@448,STRING$(64," ")

380 PRINT@448,"HOW MANY QUESTIONS ARE TO BE PRINTED ";:
     INPUTN:PRINT@448,STRING$(64," ")

390 PRINT@448, "TEST TITLE OR # ";:
     LINEINPUTTT$:CLS:
     PRINT@448,CHR$(23)"Set Paper to Top of Form and hit <ENTER>";:INPUTZZ$


400 CLS:PRINT@448,CHR$(23)"Test Heading Being Printed"

410 LPRINT C$;TAB(54)"NAME";STRING$(19,95):LPRINT

420 LPRINT "":LPRINTTT$;TAB(54)"DATE";STRING$(8,95);"PERIOD";STRING$(5,95)

430 LPRINTSTRING$(2,138):LPRINT"Multiple Choice -";V;" Points Each":
     LPRINT"":LPRINT"":LC=8

440 FORI=SNTOLOF(1):QN=I

450 GET1,I:QQ$=Q$:P=INSTR(QQ$,"   "):QQ$=LEFT$(QQ$,P-1):Q2$=QQ$

460 IF LEFT$(QQ$,2)="* "THEN U=1:QQ$=RIGHT$(QQ$,LEN(QQ$)-2):GOTO460

470 CLS:PRINTTAB(25)"QUESTION # ";I:PRINT:PRINTQQ$:PRINT:PRINT:PRINT

480 IF U=1 THEN PRINT@725,"QUESTION USED":U=0:PRINT:PRINT

490 P$="":INPUT"HIT <P> TO PRINT QUESTION - <ENTER> FOR NEXT QUESTION";P$:
     IFP$="p"THENP$="P"

500 IFP$=""THENNEXTI:GOTO570

510 IF P$<>"P" THEN GOSUB690:GOTO470

520 GOSUB960:NN=NN+1:NU$=STRING$(5,95)+STR$(NN)+". ":
     LPRINTNU$;L1$:LPRINTTAB(5)L2$:
     LC=LC+2:IFL3$<>""THENLPRINTTAB(5)L3$:LC=LC+1

530 IFL4$<>""THENLPRINTTAB(5)L4$:LC=LC+1

540 LPRINT"":LC=LC+1:QQ$=Q2$:Q2$=""

550 GOSUB710:L1$="":L2$="":L3$="":L4$="":
     QQ$="* "+QQ$:GOSUB810:IFNN=NTHENNN=0:CLS:
     PRINT@448,CHR$(23)"TEST IS COMPLETED":FORT=1TO500:
     NEXTT:GOTO30

560 NEXTI

570 CLS:PRINT@192,"END OF FILE REACHED":PRINT:
     PRINT"At what question # should file seach recommence":
     PRINTTAB(15);:INPUT "('0' to end search)";M4

580 IF M4=0THEN30ELSEIFM4>LOF(1)THENGOSUB690:GOTO570

590 SN=M4:M4=0:GOTO440

600 '
                REMOVES 'QUESTION USED' MARKERS

610 CLS:PRINT@448,CHR$(23)"USE MARKERS BEING REMOVED"

620 FOR I=1TOLOF(1)

630 GET1,I:QQ$=Q$

640 IF LEFT$(QQ$,2)<>"* "THEN 680

650 IF LEFT$(QQ$,2)<>"* "THEN670

660 QQ$=RIGHT$(QQ$,LEN(QQ$)-2):GOTO650

670 LSETQ$=QQ$:PUT1,I

680 NEXTI:GOTO30
```

Program continues

in the last position of the string. The print line cannot end in the middle of a word. Line 980 checks the length of the print line to see it does not exceed the capabilities of the printer. Line 990 checks the first print line for a ")" in the last two bytes of the line. If so, the line is shortened and line 970 again searches for a blank. Line 1000 increments the pointers used to break up the question. The rest of the subroutine repeats the process for the second and, if necessary, the third and fourth print lines.

After breaking up the question the program returns to line 300-310 where the question number and the question are printed. Then the process begins for the next question.

The Paper Tiger has a 2K buffer to eliminate time lag between breaking the question into print lines and actual printing. Printers without a buffer do create a time lag. Eliminate the lag by using a spooler such as the one included with Microsystems DOSPLUS 3.3. The spooler creates a buffer to feed the printer while the CPU works on other things.

Module IV—Test Printer

This section (lines 330-590) prints the multiple choice section. It provides a heading and an automatic form feeder based on a 66-line page. You are asked to enter the name of the course, how many points each answer is worth, where the computer should start searching for questions, how many questions are needed for the test and the test's title (Midterm, Spring, 1982).

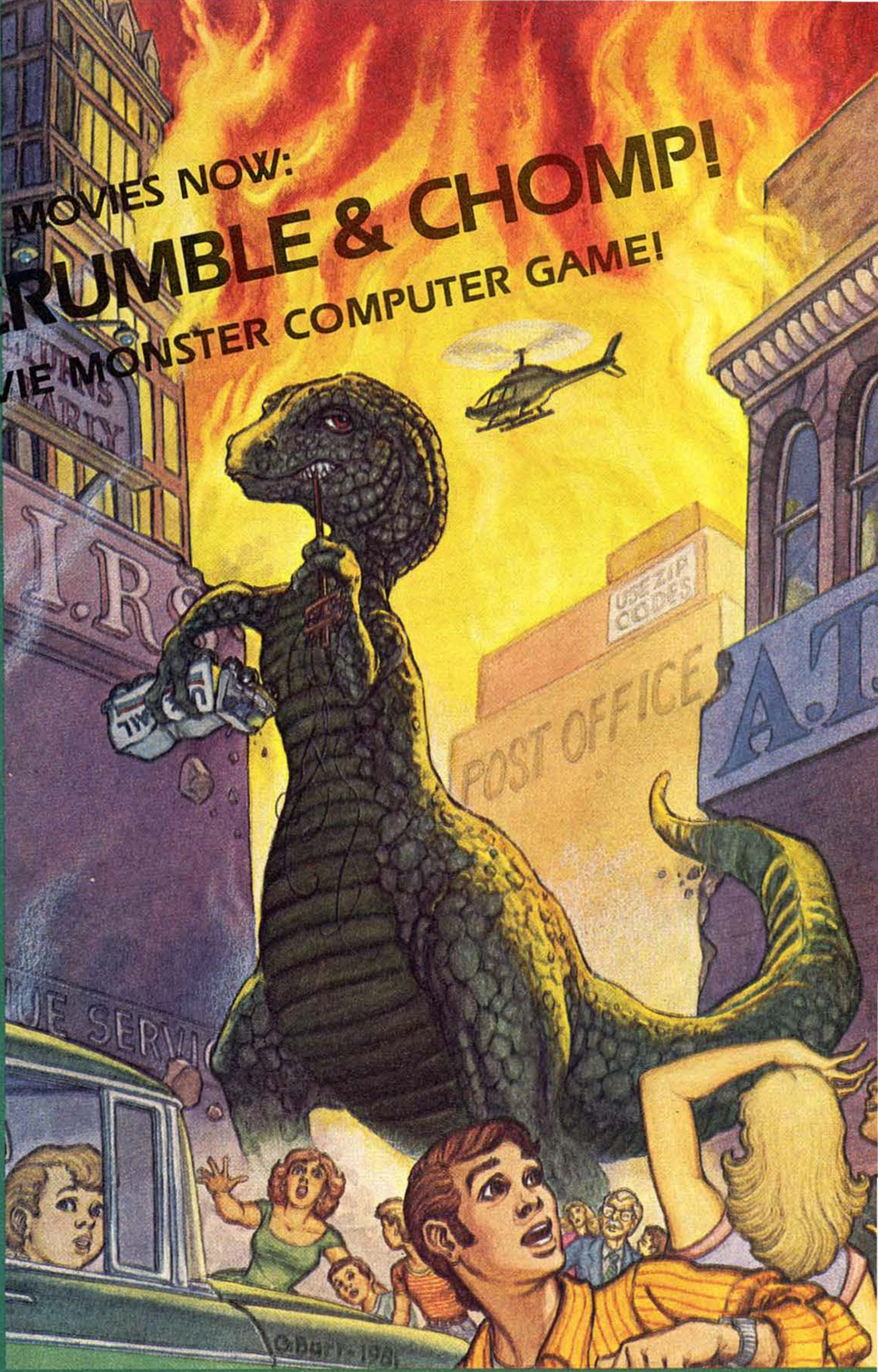
The program prompts you to set the printer to the top of the page and then prints the heading.

The first step in handling the question is a check performed by line 460 for "question used" markers. If any are attached to the question a flag is set and the markers are removed. Line 470 prints the requested question on the screen. If the question has been used on a previous test, line 480 prints this information on the screen. Line 490 allows you to designate a null, sending the computer to the

FIRST: MONSTER MOVIES NOW:

CRUSH, CRUMBLE & CHOMP!

THE GREAT MOVIE MONSTER COMPUTER GAME!



And guess who stars as the movie monster. You! As any of six different monsters. More if you have the disk version.

You can terrorize and destroy four of the world's largest and most densely populated cities in over 100 possible scenarios. From Tokyo to the Golden Gate, you are the deadliest creature in the air, on the land, or in the sea.

You can be the deadly amphibian who simultaneously smashes street cars, lunches on helpless humans and radiates a ray of death.

If you were a giant winged creature, think of the aerial attacks you could make on the terrified but tasty tidbits beneath you.

But as in all the best monster movies, you're up against everything the human race can throw at you—even nuclear warheads and a strange concoction developed by a team of mad scientists.

For only $29.95 you get 6 stupendous monsters, each with its own monstrous summary card, 4 teeming metropoli displayed in graphic detail on your computer display and mapped in the accompanying 48-page illustrated book, the awesome sounds of monsterly mayhem, and spine-tingling, real-time, edge-of-your-seat excitement.

GET CRUSH, CRUMBLE & CHOMP
now at your local dealer for your APPLE, ATARI, or TRS-80 . . **before it's too late.**

EPYX
COMPUTER GAMES THINKERS PLAY

Program continued

```
685 '
                          ERROR MESSAGE
690 PRINT:PRINT"Improper input - Please try again":
      PRINT:FORT=1TO350:NEXTT:RETURN

700 '
                          PAGER

710 IF LC>=60 THEN LPRINT STRING$(66-LC+4,138):LC=4

720 RETURN

730 '
                    SUBROUTINE EDITS QUESTIONS

740 CLS:PRINTTAB(23)"Question Editor":PRINT

750 PRINTQQ$:PRINT:IFLEN(QQ$)>250THENPRINT:
      PRINT"THIS QUESTION IS TOO LONG FOR THE FILE BY ";LEN(QQ$)-250

760 PRINTTAB(13)"Edit menu":PRINTTAB(17)"1. Exit edit mode":PRINTTAB(17)"2.
 Make change in present form":PRINTTAB(17)"3. Redo entire question":PRINTTA
B(17)"4. Cancel changes and return to original form":PRINTTAB(17)"5. Store
present form of question"

770 GOSUB940:INPUTM3:IFM3=1THENRETURNELSEIFM3<1ORM3>5THENGOSUB690:GOTO770

780 IFM3=2THEN820

790 IF M3=3 THENCLS:
      PRINT"What is the new question":PRINT:
      LINEINPUTQQ$:GOTO740

800 IFM3=4THENQQ$=Q2$:GOTO740

810 LSETQ$=QQ$:PUT1,QN:RETURN

820 CLS:PRINTQQ$:PRINT:
      PRINT"Enter word or phrase to be corrected.  If punctuation is to be
 altered, enter nearest word BEFORE the punctuation and include the punctua
tion mark. ('0' to cancel edit)":
      PRINT:LINEINPUTT$:IFT$="0"THENCLS:GOTO750

830 P1=INSTR(QQ$,T$)-1:
      IFP1=-1THENCLS:PRINT"The word(s) you want to correct don't exist - Tr
y again":
      PRINT:GOTO750

840 PRINT:PRINT"Type the corrected version":LINEINPUTN$:L=LEN(N$)

850 P2=P1+LEN(T$):L$=LEFT$(QQ$,P1):R$=RIGHT$(QQ$,(LEN(QQ$)-P2))

860 QQ$=L$+N$+R$:GOTO740

870 '
                    CONVERTS LOWER TO UPPER CASE

880 IFZ$="y"THENZ$="Y":GOTO920

890 IFZ$="n"THENZ$="N":GOTO920

900 IF  QQ$="end"THENQQ$="END":GOTO920

910 IFCN$="e"THENCN$="E"
920 RETURN

930 '
                          MENU QUESTION

940 PRINT:PRINT"INPUT THE # OF THE PROCESS TO BE CARRIED OUT";:RETURN

950 '
                 BREAKS UP QUESTION INTO PRINT LINES

960 EL=63

970 L1$=LEFT$(QQ$,EL):IFRIGHT$(L1$,1)<>" "THENEL=EL+1:GOTO970

980 IF LEN(L1$)>69 THEN EL=EL-15:GOTO970

990 IF INSTR(RIGHT$(L1$,2),")")>0 THEN EL=EL-6:GOTO970

1000 EL=EL+1:E2=EL+68

1010 IF LEN(QQ$)<= LEN(L1$)+74 THEN L2$=RIGHT$(QQ$,LEN(QQ$)-(EL-1)):RETURN

1020 L2$=MID$(QQ$,EL,E2-EL):
      IFRIGHT$(L2$,1)<>" " AND LEN(QQ$)> LEN(L1$)+LEN(L2$) THENE2=E2+1:GOTO
1020

1030 IF L1$+L2$=QQ$ THEN RETURN

1040 IF LEN(L2$)>78 THEN E2=E2-15:GOTO 1020

1050 IF INSTR(RIGHT$(L2$,2),")")>0THEN E2=E2-6:GOTO1020

1060 EL=E2:E2=EL+68

1070 IF LEN(QQ$)<=LEN(L1$)+LEN(L2$)+74 THEN L3$=RIGHT$(QQ$,LEN(QQ$)-(EL-1))
:
      RETURN

1080 L3$=MID$(QQ$,EL,E2-EL):
      IF RIGHT$(L3$,1)<>" " AND LEN(QQ$)>LEN(L1$)+LEN(L2$)+LEN(L3$) THEN E2
=E2+1:
      GOTO1080

1090 IFL1$+L2$+L3$=QQ$ THEN RETURN

1100 IF LEN(L3$)>78 THEN E2=E2-15:GOTO1080

1110 IFINSTR(RIGHT$(L3$,2),")")>0THENE2=E2-6:GOTO1080

1120 EL=E2:L4$=RIGHT$(QQ$,LEN(QQ$)-(EL-1)):RETURN
```

next question, or a P, sending the computer to the 950 subroutine, incrementing a question counter and sending the question to the printer. Finally, in line 550 a marker is added to the question so you will know the question was used. Every time you use the question another marker is added. This continues until the number of questions you designated has been printed. If you reach the end of the file before completing the test, you will be asked to designate a starting question number to begin the search cycle again.

Module V

This section checks each question for a Used marker and, if present, removes it. It is used at the end of the course to refresh the file.

Subroutines

While the functions of most subroutines are obvious, an index would be useful. Subroutine 690 is the error message. It flashes if you enter an incorrect Input anywhere in the program. Subroutine 700 checks after each question to see if 60 or more lines have been printed. If so, several line feeds are entered to move to the top of the next page. My Paper Tiger printer needs a CHR$(138) for each line feed. Most printers use CHR$(10). Subroutine 730 is the editor. Subroutine 870 converts lowercase input to uppercase. Subroutine 930 prints the menu questions used in both menus. Subroutine 950 breaks the question into print lines.

Possibilities

This program can be helpful for any application requiring sequential recall of data. A key word search routine would expand the program's usefulness while effectively eliminating the sequential aspect of the search. A line would be necessary to designate the key word. A second line can check if the key word is a null. If not, only questions containing the key word will be displayed. These two lines can be included in a subroutine or as part of Modules III and IV.■

MTI® can upgrade your TRS-80 Model III to have giant data storage like the Big Boys.

with prices like the little guys.

WINCHESTER/e

MTI DOES IT AGAIN. Our WINCHESTER/e can give you up to 30 megabytes of storage and turn your TRS-80 into a real business system.

LIKE THE BIG BOYS. That means you can access 5 or 7.5 megabytes of on-line storage used by the big companies, such as Xerox, IBM, Alpha Micro, and DEC. All of this and the expansion capabilities to 30 megabytes. Typically access is 12 times faster than floppy while maintaining file compatibility with TRS DOS and DOS plus. (6 or 9 megabytes unformatted.)

SIMPLE INSTALLATION. The disk system is easy to install. Simply plug the cable connector into the rear expansion port of the Model III.

INCREASED APPLICATIONS. With the MTI WINCHESTER/e you can run hundreds of business applications previously not possible with existing floppy based systems.

OURS IS REALLY USEABLE: Giving you Winchester capability with a really useble DOS system. Unlike other systems, the MTI operating system does not divide the file size to look like 54 floppy disks of storage. It gives the ability to define your file size, any size, up to 5 or 7.5 megabytes. For example: This would allow you to have ONE FILE with 58,000 customer names.

BACK-UP CAPABILITY. The MTI WINCHESTER/e allows fully back-up on any TRS-80 Model III floppy or Winchester. File may be backed up by file, platter, or entire drive. Other models on the market have severe limitations in this area. This will give you professional data storage ability for virtually hundreds of real business needs.

JUST LIKE THE BIG BOYS.

SIX MODELS AVAILABLE

Winchester 5/e	5 Megabytes
Winchester 7.5/e	7.5 Megabytes
Winchester 2 x 5/e	Double 5 (10 Megabytes)
Winchester 2 x 7.5/e	Double 7.5 (15 Megabytes)
Winchester 5/e80	with 80 trk. dual head floppy
Winchester 7.5/e80	with 80 trk. dual hd. floppy bk-up

Prices from $2,695.

CP/M-80b MTI can turn your TRS-80 Model III into two computer systems, TRS-DOS, and CP/M. By adding a CP/M card and an 80 x 24 column video card, you will convert your computer to a CP/M machine.

Speed-Up Enhancement	**$149.**
Internal Disk Drive	**$695.**
External Disk Drive	**$379.**

WANT MORE? For more information call any of our 64 authorized sales and service centers listed below. If you prefer, you can call or write us and we'll send you the information and a catalog.

TRS-80 is a Trademark of Radio Shack.
MTI is a Registered Trademark.

MTI AUTHORIZED SALES AND SERVICE CENTERS

Anchorage, AK	(907) 333-8322	Northridge, CA	(213) 886-9200	Mary Esther, FL	(904) 243-5793	Raleigh, NC	(919) 755-1175
Rogers, AR	(501) 636-9168	Pacheco, CA	(415) 689-2260	Panama City, FL	(904) 769-5887	Grand Forks, ND	(701) 772-7848
Mesa, AZ	(602) 833-8949	Port Hueneme, CA	(805) 985-2329	Tampa, FL	(813) 247-6023	West Milford, NJ	(201) 728-8080
Phoenix, AZ	(602) 244-9739	Redondo Beach, CA	(213) 370-5556	Norcross, GA	(404) 449-8982	Las Vegas, NV	(702) 870-4138
Sierra Vista, AZ	(602) 458-2479	Riverside, CA	(714) 781-6361	Blackfoot, ID	(208) 785-1497	Jericho, NY	(516) 333-2266
Tempe, AZ	(602) 831-5376	San Jose, CA	(408) 946-1265	Belleville, IL	(618) 277-2354	New Rochelle, NY	(914) 235-4444
Tucson, AZ	(602) 323-9391	San Diego, CA	(714) 275-4243	Collinsville, IL	(618) 345-5068	Syracuse, NY	(315) 487-5000
Anaheim, CA	(714) 773-0240	Vallejo, CA	(707) 554-4933	Oak Park, IL	(312) 386-3323	Troy, NY	(518) 273-8411
Costa Mesa, CA	(714) 968-4076	Walnut, CA	(714) 594-9760	Shreveport, LA	(318) 865-7189	White Plains, NY	(914) 761-9283
Covina, CA	(213) 332-4088	Westlake Village, CA	(213) 706-0333	Shawnee, KS	(913) 631-2029	Portland, OR	(503) 668-8666
Goleta, CA	(805) 967-7628	Grand Junction, CO	(303) 434-3616	Minneapolis, MN	(612) 427-5783	Tulsa, OK	(918) 836-9503
Huntington Beach, CA	(714) 842-1348	Groton, CT	(203) 445-5166	Camerson, MO	(816) 632-6528	Dallas, TX	(214) 698-9995
Lancaster, CA	(805) 942-5747	Wilmington, DE	(302) 762-0227	Joplin, MO	(417) 782-0880	San Angelo, TX	(915) 658-3781
Montclair, CA	(714) 626-4813	Hollywood, FL	(305) 981-1011	Missoula, MT	(406) 549-9715	Cheyenne, WY	(307) 632-9132

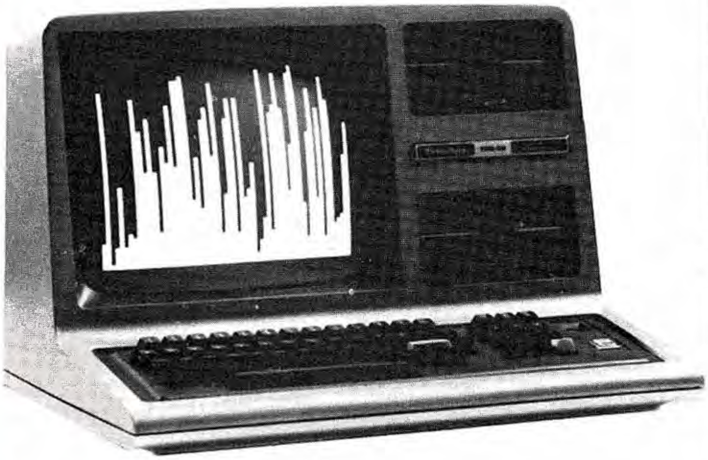
MICROCOMPUTER TECHNOLOGY INC.
3304 W. MACARTHUR, SANTA ANA, CA 92704
(714) 979-9923 • TWX 910-595-1902 MTISNA

28

Call or write for free brochure:
U.S. PRICES, F.O.B. SANTA ANA CALIFORNIA AND MAY VARY BY AREA.
CP/M is a registered trademark of Digital Research, Inc.

5/10 Mbyte TRS-80® Model III

REDUCED $1,000.00



Model 326 - 10+ Mbytes

Over 10 Mbytes of storage is attained using a double-headed 80-track drive and a 9.57 Mbyte Winchester, all housed in the Model III system. Drive storage is approximately equivalent to more than 50 single-sided 40 track floppy drives. Storage is approximately equivalent to 2000 pages of text

$5,495.00

Model 325 - 5+ Mbytes

Over 6.0 Mbytes of storage is attained using a double-headed 40-track drive and a 6.38 Mbyte Winchester, all housed in the Model III system. Drive storage is equivalent to more than 27 single-sided 40-track floppy drives. Storage is approximately equivalent to 1000 pages of text.

$4,995.00

COMPUTEX Model III Systems

Model 324

1.4 Mbyte floppy disk storage with 2 double-sided 80-track drives. Drives can be configured and accessed as 2 single-volume 160-track drives, or as 4 individual 80-track drives. Drive storage is equivalent to 8-drive system, using only 2 physical disk drives. Storage is approximately equivalent to 320 pages of text.

$2,595.00

Model 322

740 Kbyte floppy disk storage with 2 double-sided 40-track drives. Drives can be accessed as either 2 single-volume 80-track drives, or as 4 individual 40-track drives. Drive storage is equivalent to a 4-drive system using only 2 physical disk drives. Storage is approximately equivalent to 160 pages of text.

$2,295.00

Model 321

370 Kbyte floppy disk storage with 2 single-sided 40-track drives. Storage approximately equivalent to 80 full pages of text. System configuration is identical to Radio Shack 26-1066. less RS232.

$1,895.00

SPECIFICATIONS

Microprocessor • Z-80 runs 2.03 MHz, 4.0 MHz optional.

Winchester Drive Specifications • Tandon Winchester storage up to 10 Mbytes on 5¼" fixed disk • backup to floppy by file/name or sector count • hardware is Tandon TM602/TM603 Winchester, Western Digital WD1000 drive controller • Computex host adaptor/drive controller and switching power supply • Winchester drive is accessed as drive #4 • standard DOS commands are used in all Winchester accesses.

Floppy Disk Drive • Disk drives: Tandon TM100 series supporting track-to-track access time of 5ms. • storage capacities range from 340K to 9.57 Mbytes • drive controller board by Computex.

Transient Protection • all systems include a metal oxide varister which prevents transients from damaging your system.

SERVICE, ETC.

Every system is thoroughly tested before it leaves the plant and is backed by a 90-day parts and labor warranty. Annual maintenance contracts are available. Spare parts are available at very reasonable prices. Software is backed by the original producer and/or one of our highly qualified consulting software engineers.

To supplement our considerable in-house capabilities, we work with some of the best electronics engineers and computer programmers in the area. You can always count on Computex for continued support in service and update for your system.

COMPUTEX

(713) 488-8022

MODEL III DISK DRIVE EXPANSION KITS

• Switching power supply • storage to 2 Mbytes • supports TRSDOS, LDOS, DOS+, NEWDOS-80 • 32K RAM expansion • double density disk controller • goldplated edge card connections • includes TRSDOS and manual, all cables and necessary hardware, installation manual • fully tested and burned in • 90-day warranty

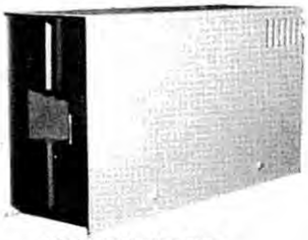
M3DK0 - Drive kit with one Tandon single-sided 40-track drive **$550.00**

M3DK1 - Drive kit with two Tandon single-sided 40-track drives **$765.00**

M3DK2 - Drive kit with two Tandon double-sided 40-track drives **$995.00**

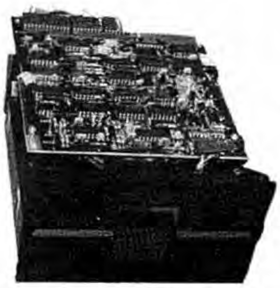
M3DK4 - Drive kit with two Tandon double-sided 80-track drives **$1,195.00**

TANDON DISK DRIVES



DRIVES WITH CASE & POWER SUPPLY

- Fast 3-5-ms track-to-track access time
- Single or double density operation
- Total compatibility with Model I or III
- Pre/post testing during 48-hour burn-in
- 90-day limited warranty



DRIVES W/O CASE & POWER SUPPLY

DRIVES WITH CASE & POWER SUPPLY		DRIVES W/O CASE & POWER SUPPLY
$299.95	TM100-1 - single-sided 40-track drive - 102 Kbytes single density 180 Kbytes double density	$235.00
$399.95	TM100-2 - double-sided 40-track drive - 204 Kbytes single density - 360 Kbytes double density	$335.00
$399.95	TM100-3 - single-sided 80-track drive - 204 Kbytes single density - 360 Kbytes double density	$335.00
$499.95	TM100-4 - double-sided 80-track drive - 408 Kbytes single density - 735 Kbytes double density	$435.00

2-drive cable - $24.95 • 4-drive cable - $34.95 • drive extender $14.95

8" THINLINE DISK DRIVES

TM848-1 **$495.00**
Single-sided 77-track - Storage: 600K single density, 1.2 Mbyte double density

TM848-2 **$595.00**
Double-sided 77-track - Storage: 1.2 Mbyte single density, 2.4 Mbyte double density

5¼" MINI-WINCHESTER DISK DRIVES

TM602S **$1,195.00**
3-ms access time - 612 tracks - 5 Mbytes/s data transfer rate - 6.38 Mbyte RPM capacity (unformatted)

TM603S **$1,295.00**
3-ms access time - 918 tracks - Mbytes/s data transfer rate - 9.57 Mbyte RPM capacity (unformatted)

MDX-2 SYSTEM EXPANSION

The new MDX-2 from Computex provides more capabilities than any other system expansion currently available. • build only what you need • silk-screened and solder-masked PCB • dip switch instead of hardware jumpers • no messy vertically-mounted termination resistors.

Complete Kit **$269.00**

- MDX02 printed circuit board....................$74.95
- Onboard direct connect modem kit....................39.95
- 2K/4K Eprom kit....................17.95
- RS232 mA serial interface....................17.95
- Real-time clock kit....................17.95
- Floppy disk controller kit....................31.95
- 32K RAM expansion kit (less RAM)....................11.95
- Cassette port....................3.95
- Analog power supply....................29.95
- Hardware and socket kit....................19.95
- Centronic line printer port....................8.95

COMPUTEX

Comprehensive Test Program $21.95

• disk drive speed test • disk drive alignment program (requires alignment disk - Dysan Alignment Disk $39.95) **• expansion interface RAM test • keyboard RAM TEST • video RAM test • keyboard bounce test • ROM checksum program • full screen RAM display program**

VISA, MASTERCARD, AMERICAN EXPRESS ACCEPTED • Net-30 terms for D&B-rated companies. All products will carry an additional 10 percent handling fee. Prices subject to change without notice. Prices do not include shipping.

COMPUTEX

✓ 392

321 El Dorado • Webster Texas 77598 **(713) 488-8022**

LOAD 80

Read all about the planets in this program.

EXTRA-Terrestrial

Thomas A. Wells
2401 Division St. Apt. E8
New Orleans, LA 70001

How nice it would be to have our sun's solar system available for inspection! Other programs that do this on the TRS-80 are not complete enough for me.

Planets provides a more detailed look at the subject. A map (within the limits of the computer's graphics) shows the locations and relative sizes of the planets in our solar system.

Planets provides facts about each of the planets including their known satellites, their actual sizes, distances from the sun and more.

One of the program's unique features is the option of comparing any planet to any other. If you are curious about how the earth's size compares with Saturn's, for example, just follow the prompts.

Running the Program

Type in the program as listed. Enter run. Answer the "Explorer's Name?" question and press Enter. Answer "Y" to the "Ready" question. The program then presents the menu. From this point the Enter key is not needed. Press the key corresponding to your menu choice. Wherever you go, there will be a prompt near the lower section of the screen. Press any of the letters indicated between quotation marks (for example, "L") to go to another program section. When you have seen enough, press Q to quit. Happy learning!

Program Description

The program is commented throughout to identify each section. You can omit the comments when typing it in.

Lines 20–50 are housekeeping. Here any variable beginning with the letter P is a string variable, saving a little typing. CHR$(34) is a double quote mark.

Lines 60-95 are all subroutines. The GOTO100 in line 50 jumps over them to continue execution.

The subroutine in line 60 clears the bottom and right side of the screen. Variable KP tells this line where to start clearing a line on the right of the screen; the whole screen does not have to be rewritten. This is used when comparing planets and listing satellites.

Line 70 is the INKEY$ keyboard input.

Lines 80–95 read the information from the data statements in lines 4990–5170. The If statement in line 90 determines whether there are any satellites belonging to the planet whose data is being read. If there are, line 80 reads the satellite names.

Lines 500–590 build strings for the solar system display. This takes less time than Set. The remainder of the program uses the same format: The screen is cleared and the requested data or list is displayed. Operator options are also displayed and a GOSUB70 waits for operator input. When a key is pressed, it is checked for validity and the program branches to the requested action.

Lowercase Note

I developed the program for the TRS-80 Model I using the lowercase option and the Radio Shack ULCDVR lowercase driver. It should work on any machine without lowercase with no changes to the program. If you do not have the lowercase option, leave out the part of each If statement containing lowercase. ■

Program Listing

```
10 REM
        Planets - (C) 1981 - T.A. Wells - N.O., La. 70118
19 REM
        Declare, dimension & assign variables.
```

Program continues

The Key Box

Basic Level II
Model I
16K RAM

Program continued

```
20 CLEAR400
30 DEFSTRP:DEFDBLD,M:DEFINTI-L
40 DIMP(9),DI(9),PV(9),PR(9),OD(9),SG(9),PW(9),PT(9),PS(9,16),PM
(9)
50 PQ=CHR$(34):KP=25:GOTO100
59 REM
        Subroutine to clear bottom & right side of screen.

60 PRINT@640,CHR$(31);:FORI=1TO16:PRINT@KP-1,CHR$(30);:KP=KP+64:
NEXT:KP=45:RETURN
69 REM
        Subroutine for keyboard input of choices.

70 P=INKEY$:IFP=""THEN70ELSERETURN
79 REM
        Read data.
80 FORJ=1TONS(I):READPS(I,J):NEXT:RETURN
90 FORI=0TO9:READP(I),DI(I),PV(I),PR(I),OD(I),SG(I),PW(I),PT(I),
NS(I):IFNS(I)>0THENGOSUB80:NEXT:RETURNELSENEXT:RETURN
95 FORI=1TO9:READPM(I):NEXT:RETURN
99 REM
        Title.

100 CLS:PRINT@274,CHR$(23)"P l a n e t s"
110 PRINT@398,"of the Solar System":GOSUB90:GOSUB95
120 PRINT@640,:INPUT"Explorer's Name - type it
then press <ENTER>";PE
130 IFLEN(PE)>8THENPE=LEFT$(PE,8)ELSEIFLEN(PE)=0THENPE="Explorer
"
139 REM
        Assign variables for planet display.

140 PP(0)="Planet":PP(1)="Distance from Sun (Km)":PP(2)="Revolve
s around Sun in":PP(3)="Rotates on its axis in":PP(4)="Diameter
(Km)":PP(5)="Density (Water = 1)":PP(6)="Mass":PP(7)="Temperatur
e (deg C)":PP(8)="Number of Satellites"
199 REM
        Introduction.

200 CLS:PRINT@30,"P l a n e t s"
210 PRINT@192,"Hi, "PE"."
220 PRINT@320,"   I'm your friendly space exploration helper. If
 you give me   directions I can tell you about any planet in the
 Solar System."
230 PRINT"   When we look, you'll be able to find out some thing
s about   the planet we're exploring. You'll also be able to com
pare the  planet to another planet or list its satellites."
240 GOSUB500:PRINT:PRINT"   Are you ready (Y/N)?"
245 GOSUB70
250 IFP<>"Y"ANDP<>"y"ANDP<>"N"ANDP<>"n"THEN245ELSEIFP="Y"ORP="y"
THENCLSELSE4000
299 REM
        Menu.
300 CLS:PRINT@18,"P l a n e t s"
310 PRINT@128,"O.K., "PE".  Here is a list of the places we can
go:
320 PRINT:PRINT"0 - Sun"TAB(30)"5 - Jupiter
1 - Mercury"TAB(30)"6 - Saturn
2 - Venus"TAB(30)"7 - Uranus
3 - Earth"TAB(30)"8 - Neptune
4 - Mars"TAB(30)"9 - Pluto
A - Asteroid Belt"TAB(30)"D - Display Solar System"
330 PRINT:PRINT"Press "PQ"L"PQ" to get back this list when you w
ant to.
Press "PQ"Q"PQ" to quit (anytime)."
340 PRINT:PRINT"Where shall we go?  <Press 0-9, "PQ"A"PQ" or "PQ
"D"PQ">":GOSUB70
350 IP=VAL(P):IT=IP:IF(IP>0)OR(IP=0AND(P="D"ORP="d"ORP="Q"ORP="q
"ORP="a"ORP="A"ORP="0"))THEN360ELSEGOSUB70:GOTO350
360 IFP="0"THEN3000ELSEIFP="D"ORP="d"THEN600
370 IFP="Q"ORP="q"THEN4000
400 IFP<>"a"ANDP<>"A"THEN1000ELSE2000
499 REM
        Assemble strings for Solar System graphic.

500 PB=CHR$(24)+CHR$(26)
510 PS=CHR$(140)+CHR$(176)+CHR$(144)+PB+CHR$(131)+CHR$(141)+CHR$
(180)+CHR$(144)+PB+CHR$(130)+CHR$(141)+CHR$(176)+CHR$(26)+CHR$(1
37)+CHR$(144)+PB+CHR$(130)+CHR$(164)+PB+CHR$(130)+CHR$(148)+PB+C
HR$(189)+PB+CHR$(159)+PB+CHR$(24)
520 PS=PS+CHR$(160)+CHR$(133)+PB+CHR$(24)+CHR$(24)+CHR$(160)+CHR
$(134)+PB+STRING$(2,24)+CHR$(152)+CHR$(129)+PB+STRING$(4,24)+CHR
$(160)+CHR$(156)+CHR$(131)+PB+STRING$(5,24)+CHR$(160)+CHR$(152)+
CHR$(134)+CHR$(129)
530 PS=PS+PB+STRING$(5,24)+CHR$(140)+CHR$(134)+CHR$(129)
540 MM$=CHR$(132):EV$=CHR$(140):JU$=CHR$(174)+STRING$(3,191)+CHR
$(157)+"Jupiter":SA$=CHR$(174)+CHR$(191)+CHR$(191)+CHR$(157)+"Sa
turn":UR$=CHR$(174)+CHR$(157)+"Uranus":NE$=CHR$(174)+CHR$(132)+"
Neptune"
590 RETURN
599 REM
        Display Solar System.

600 CLS:PRINT@64,PS;
610 PRINT@332,MM$"Mercury";:PRINT@397,EV$"Venus";:PRINT@462,EV$"
Earth";:PRINT@527,MM$"Mars";:PRINT@405,JU$;
620 PRINT@475,SA$;:PRINT@486,UR$;:PRINT@499,NE$;:PRINT@440,"Plut
o "CHR$(129);
630 PRINT@450,"SUN";:PRINT@74,"Note: Distances and sizes are onl
y approximations.";:PRINT@144,"If you are going, use a better ma
p!";
640 PRINT@848,PQ"L"PQ" - to return to List of Planets";
650 GOSUB70
660 IFP="L"ORP="l"THEN300ELSEIFP="Q"ORP="q"THENPRINT:GOTO4000ELS
E650
999 REM
        Print planet data.

1000 CLS
1010 FORI=0TO8:PRINTPP(I):NEXT
1020 PRINT@KP,P(IP)" (#"IPCHR$(24)")";:KP=KP+64:PRINT@KP-11+LEN(
STR$(DI(IP)))+1*(DI(IP)<1000000000),;:PRINTUSING"###########,#";
DI(IP);:KP=KP+64:PRINT@KP,PV(IP);:KP=KP+64
1030 PRINT@KP,PR(IP);:KP=KP+64:PRINT@KP+LEN(STR$(OD(IP)))-7,;:PR
INTUSING"###,###";OD(IP);:KP=KP+64:PRINT@KP,;:PRINTUSING"#.#";SG
```

Program continues

✓See List of Advertisers on page 290

Program continued

```
(IP);:KP=KP+64
1040 PRINT@KP,PW(IP);:KP=KP+64:PRINT@KP,PT(IP);:KP=KP+64:PRINT@K
P-3+LEN(STR$(NS(IP))),CHR$(30);:PRINTUSING"##";NS(IP):KP=25
1050 IP=IT:PRINT@640,CHR$(30);"Here are some things you can do:"

1060 PRINTPQ"C"PQ"-Compare to another Planet   "PQ"M"PQ"-More Da
ta on "P(IP)
1070 PRINTPQ"L"PQ"-Return to List of Planets   "PQ"S"PQ"-See "P(
IP);:IFRIGHT$(P(IP),1)="s"ORRIGHT$(P(IP),1)="S"THENPRINT"' Satel
lites"ELSEPRINT"'s Satellites"
1080 PRINT"Your orders, Captain "PE"?"
1090 GOSUB70
1100 IFP="S"ORP="s"THEN1200
1110 IFP="l"ORP="L"THEN300
1115 IFP="M"ORP="m"THEN1500ELSEIFP="Q"ORP="q"THEN4000
1120 IFP<>"C"ANDP<>"c"THEN1090
1139 REM
        Compare planets.

1140 KP=45:GOSUB60:KP=25:PRINT@640,CHR$(31);"Compare "P(IP)" (#"
IP;CHR$(24)") to which of the others?"
1160 FORIG=1TO9:IFIG<>IPTHENPRINTIG"- "P(IG),:NEXTELSENEXT
1180 GOSUB70:IX=VAL(P):IFIX<>IPANDIX>0THENIT=IP:IP=IX:KP=46:GOSU
B60:GOTO1020ELSE1180
1199 REM
        List satellites.

1200 KP=45:PRINT@640,CHR$(31);:GOSUB60:IFNS(IP)=0THENPRINT@KP,"
No";:PRINT@KP+64," Satellites";:PRINT@KP+128," Known";:IP=IT:GOT
O1220
1210 FORIS=1TONS(IP):PRINT@KP,CHR$(30)" #"IS;PS(IP,IS);:KP=KP+64
:NEXT
1220 PRINT@640,PQ"L"PQ" - List of Planets";:PRINT@704,PQ"C"PQ" -
 Compare Planets";:PRINT@768,PQ"M"PQ" - More Data on "P(IP);:PRI
NT@896,"Your Choice?";
1230 KP=24:GOSUB70
1240 IFP="L"ORP="l"THEN300ELSEIFP="Q"ORP="q"THEN4000
1250 IFP="M"ORP="m"THEN1500
1260 IFP="C"ORP="c"THEN1140ELSE1230
1499 REM
        Compare planets.

1500 KP=45:GOSUB60:KP=25:PRINT@640,CHR$(31)PM(IP)
1510 PRINT:PRINTPQ"L"PQ" - List of Planets"TAB(32)PQ"C"PQ" - Com
pare another Planet"
1520 PRINTPQ"S"PQ" - See "P(IP);:IFRIGHT$(P(IP),1)="s"ORRIGHT$(P
(IP),1)="S"THENPRINT"' Satellites";ELSEPRINT"'s Satellites";
1530 GOSUB70
1540 IFP="L"ORP="l"THEN300ELSEIFP="q"ORP="Q"THENPRINT:GOTO4000
1550 IFP="C"ORP="c"THEN1140
1560 IFP="S"ORP="s"THEN1200ELSE1530
1999 REM
        Asteroid belt.

2000 CLS
2010 PRINTTAB(20)"Asteroid Belt":PRINT
2020 PRINT"Location:  Between Mars (# 4) and Jupiter (# 5).
2030 PRINT"Composition:  Mostly rock fragments.
2040 PRINT:PRINT"   Asteroids range in sizes from 1,000 km. to l
ess than 1 km.  They are thought to be leftover debris from the
 formation of ourSolar System."
2090 GOTO3070
2999 REM
        Sun.

3000 CLS
3010 PRINT"The "P(IP):PRINTPP(3)TAB(30)PR(IP)
3020 PRINTPP(6)TAB(30)PW(IP)
3030 PRINTPP(4)TAB(29);:PRINTUSING"#########,";OD(IP)
3040 PRINTPP(7)TAB(30)PT(IP):PRINT
3050 PRINT"   The Sun is a rotating nuclear furnace composed mos
tly of
hydrogen and helium."
3060 PRINT"   Sunspots are magnetic storms, cooler places on the
 Sun's
surface.  Sunspot activity occurs in eleven year cycles.  The
latest maximum sunspot activity occurred in 1980."
3070 PRINT:PRINT
3080 PRINTPQ"L"PQ" - List of Planets"TAB(32);:IFIP<>0THENPRINTPQ
"M"PQ" - More Data on "P(IP)ELSEPRINT
3090 IFIP<>0THENPRINTPQ"C"PQ" - Compare another Planet    ";:PRI
NTPQ"S"PQ" - See "P(IP);:IFRIGHT$(P(IP),1)="s"ORRIGHT$(P(IP),1)=
"S"THENPRINT"' Satellites";ELSEPRINT"'s Satellites";
3100 GOSUB70
3110 IFP="L"ORP="l"THEN300ELSEIFP="q"ORP="Q"THENPRINT:GOTO4000
3115 IFIP=0THEN3100
3120 IFP="M"ORP="m"THEN1500
3130 IFP="C"ORP="c"THEN1140
3140 IFP="S"ORP="s"THEN1200ELSE3100
3999 REM
        End...

4000 PRINT"O.K. "PE", I am...";:END
4989 REM
        Data for Sun, Planets & Satellites.

4990 DATASun,,,25 days at its equator,1400000,,"333,000 x Earth"
,+15x10[6 Core / +5500 Surface,0
5000 DATAMercury,57900000,88 Days,59 Days,4878,5.4,0.055 x Earth
,+430 Day/-170 Nite,0
5010 DATAVenus,108200000,225 Days,243 Days,12100,5.3,0.8 x Earth
,+480,0
5020 DATAEarth,149600000,365.3 Days,23.9 Hours,12756,5.5,6x10[21
 Met. Tons,Avg. +15,1,Moon
5030 DATAMars,227900000,687 Days,24.6 Hours,6787,3.9,0.1 x Earth
,Avg. -50,2,Phobos,Deimos
5040 DATAJupiter,778300000,11.86 Years,9.9 Hours,142800,1.3,318
x Earth,-130 @ Cloud Tops,16,1979 J 3,Adrastea,Amalthea,1979 J 2
,Io,Europa,Ganymede,Callisto,Leda,Himalia,Lysithea,Elara,Ananke,
Carme,Pasiphae,Sinope
5050 DATASaturn,1427000000,29.46 Years,10.7 Hours,120600,.7,95 x
 Earth,-185 @ Cloud Tops,16,1980 S 28,1980 S 27,1980 S 26,1980 S
 1,1980 S 3,Mimas;Enceladus,Tethys,1980 S 13 (?),Dione,1980 S 6,
Rhea,Titan,Hyperion,Iapetus,Phoebe
5060 DATAUranus,2870000000,84 Years,17?-24? Hours,51800,1.2,15 x
 Earth,-215 @ Cloud Tops,5,Miranda,Ariel,Umbriel,Titania,Oberon
5070 DATANeptune,4497000000,165 Years,18(?) Hours,49500,1.7,17 x

 Earth,-200 @ Cloud Tops,2,Triton,Nereid
5080 DATAPluto,5900000000,248 Years,6.4 Days,3000,1,0.002 x Eart
h,-230,1,Charon
5090 DATAHeavily cratered and airless.  This dead planet is much
 like    Earth's moon.  It has changed little since volcanic act
ivity    stopped about 3 billion years ago.
5100 DATAVenus' atmosphere is about 100 times denser than Earth'
s.       Radar reveals both impact craters and an immense volcan
o.
5110 DATAThe only planet known to support life.  It is tectonica
lly      active - the surface broken into slowly drifting plates
.        Shaken by both earthquakes and volcanos.
5120 DATADry red planet with polar caps of water and carbon diox
ide.     Also a volcano 24 km. high - a 5000 km. long canyon sys
tem      - dunes & dry channels (possibly from running water).
5130 DATARapidly spinning ball of gas compressed to liquid in th
e        center.  A high pressure storm first seen 300 years ago
 makes   the Great Red Spot.  Volcanos may feed the planet's fai
nt ring.
5140 DATAIcy particles Make up the seven rings.  The moon Titan
is the   only one in the Solar System known to have an atmospher
e        (mostly Nitrogen).
5150 DATAAxis tilted 98 degrees.  Atmosphere mostly Methane - sh
ines     blue/green.  Nine narrow rings apparently held in place
 by      small satellites.
5160 DATAProbably has Methane clouds at the top of its atmospher
e.       Also may have rings not yet observed.
5170 DATAAppears to be a snowball of Methane and water mixed wit
h rocks. For a 20 year period each revolution it passes inside N
eptune's orbit.
```

MODEL I & III **PRICE $39⁹⁵** **48K DISK**

(See Your Local Dealer)

— Dealers Inquiries Invited —

The adventure continues, you are at the gates of the Forbidden City. . . once you enter, there is no turning back

It's you and your talking computer's skill against the city's master computer and it's robots. This is a continuation of Forbidden Planet, but you do not have to have Forbidden Planet to play Forbidden City.

- ★ *Talks Through Cassette Port*
- ★ *Three (3) Different Voices*
- ★ *Machine Language*

Fantastic Software
3110 Polaris, #3
Las Vegas, Nevada 89106

master charge

(702) 362-1457

VISA

225

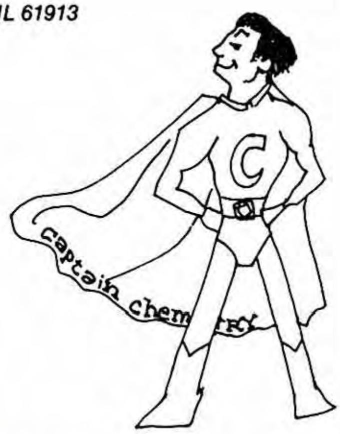
The Key Box

Basic Level II
Model I or III
16K RAM

Captain Chemistry to the Rescue.

Learning the Elements

James W. Wood
424 N. Missouri
Atwood, IL 61913



Deep within the cellars of Atwood Hammond High School lurks a figure. He is seen only by those brave souls enrolled in chemistry. He is their very own bungling hero—Captain Chemistry. The Captain has been known to emerge from behind the shelves of compounds with goggles askew and lab apron backwards for a cape. His adventures at first caused concern among administrators, but now they realize that he is harmless. The Captain may even help students learn chemistry.

Recently becoming the star of a local science movie raised his ego and the Captain now wishes to have his very own computer program. So help Captain Chemistry battle the evil elements.

This program is designed to teach the symbols for chemical elements. The Captain and evil elements are trying to destroy each other's castles. With each correct symbol the Captain's laser becomes stronger; each incorrect answer helps the evil elements. Once the correct symbol is given for an element, it will not be asked for again. Toward the end of the game the computer may take about five seconds longer to find an element. Blow the evil elements away and earn your cape, the highest award given to students of chemistry. ■

Lines

10—Sets dimensions for elements names and symbols; initializes positions for laser's beam.
60-100—Draws castle.
201-252—Places symbols and elements names into array.
290—Picks elements at random; chooses another if that one is already answered correctly.
300—Asks for symbol.
320—Checks symbol.
500—Shoots Captain's laser; determines if Evil's castle was hit; fixes elements so that it will not be chosen again.
510—Chooses positive response.
1000—Shoots Evil's laser; determines if Captain's castle was hit.
1010—Chooses negative response.
1310—Gives correct answer.
1500-1508—Graphic reward for winning.
2000—Losing message.
1000—Sets difficulty with V = V – 3; changes 3 to 1 or 2 for easier game; changes 3 to 4 or 5 for harder game.

Line Description

Program Listing

```
5 REM JAMES W. WOOD, JANUARY 1981
8 CLEAR250
10 CLS:DIM A$(104):DIM B$(104):DIM B(104):U=11   :V=114:RANDOM
20 PRINT"HELP CAPTAIN CHEMISTRY BATTLE THE ELEMENTS."
30 PRINT"EACH TIME YOU  GIVE THE CORRECT SYMBOL"
40 PRINT"THE CAPTAIN'S LASER PISTOL IS STRENGTHENED."
50 PRINT"BUT  EACH TIME YOU ARE WRONG, THE ELEMENT'S  LASER "
55 PRINT"BECOMES  STRONGER."
56 FOR Q=1 TO 5000 :NEXT :CLS
60 PRINT @ 0,"CAPTAIN'S PLACE":PRINT@  47,"EVIL  ELEMENT'S"
100 SET(6,6):SET(8,6):SET (10,6) :  FOR X =115 TO 121  STEP2
101 :SET(X,4):NEXT X
110 FOR X=115 TO 121 :FOR Y =5 TO 8:SET  (X,Y):NEXT Y:NEXT X
120 FOR X=6 TO 10:FOR Y=7TO  8:SET(X,Y):NEXT Y :NEXTX
201 A$(1)="HYDROGEN":B$(1)="H":A$(2)="HELIUM":B$(2)="HE"
202 A$(3)="LITHIUM":B$(3)="LI":A$(4)="BERYLLIUM":B$(4)="BE"
203 A$(5)="BORON":B$(5)="B":A$(6)="CARBON":B$(6)="C"
204  A$(7)="NITROGEN":B$(7)="N":A$(8)="OXYGEN":B$(8)="O"
205 A$(9)="FLUORINE":B$(9)="F":A$(10)="NEON":B$(10)="NE"
206 A$(11)="SODIUM":B$(11)="NA":A$(12)="MAGNESIUM":B$(12)="MG"
207 A$(13)="ALUMINUM":B$(13)="AL":A$(14)="SILICON":B$(14)="SI"
208 A$(15)="PHOSPHORUS":B$(15)="P":A$(16)="SULFUR":B$(16)="S"
209 A$(17)="CHLORINE":B$(17)="CL":A$(18)="ARGON":B$(18)="AR"
210 A$(19)="POTASSIUM":B$(19)="K":A$(20)="CALCIUM":B$(20)="CA"
211 A$(21)="SCANDIUM":B$(21)="SC":A$(22)="TITANIUM":B$(22)="TI"
212 A$(23)="VANADIUM":B$(23)="V":A$(24)="CHROMIUM":B$(24)="CR"
213 A$(25)="MANGANESE":B$(25)="MN":A$(26)="IRON":B$(26)="FE"
214 A$(27)="COBALT":B$(27)="CO":A$(28)="NICKEL":B$(28)="NI"
215 A$(29)="COPPER":B$(29)="CU":A$(30)="ZINC":B$(30)="ZN"
216 A$(31)="GALLIUM":B$(31)="GA":A$(32)="GERMANIUM":B$(32)="GE"
217 A$(33)="ARSENIC":B$(33)="AS":A$(34)="SELENIUM":B$(34)="SE"
218 A$(35)="BROMINE":B$(35)="BR":A$(36)="KRYPTON":B$(36)="KR"
219 A$(37)="RUBIDIUM":B$(37)="RB":A$(38)="STRONTIUM":B$(38)="SR"
220 A$(39)="YTTRIUM":B$(39)="Y":A$(40)="ZIRCONIUM":B$(40)="ZR"
```

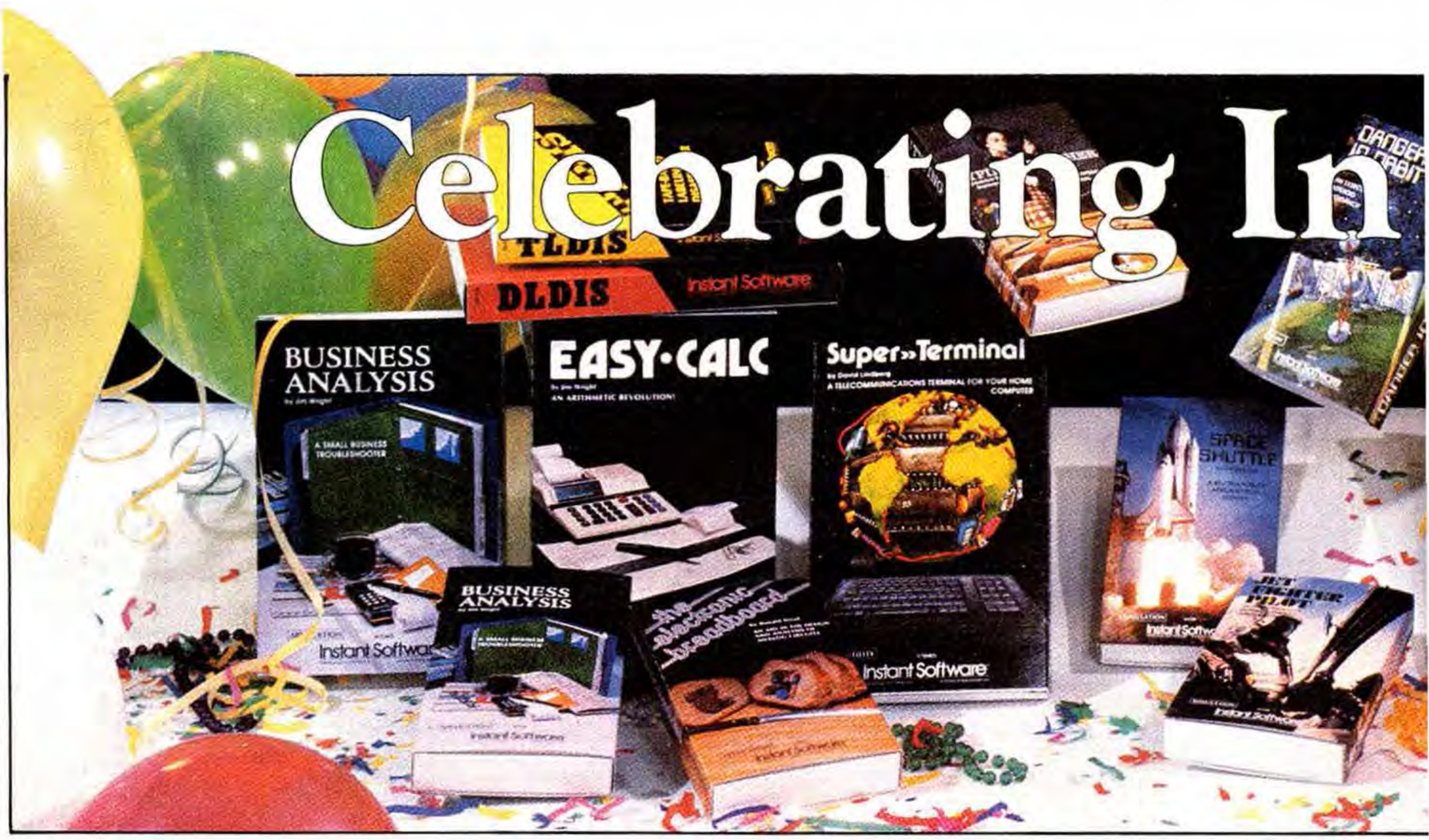
Program continues

Program continued

```
221 A$(41)="NIOBIUM":B$(41)="NB":A$(42)="MOLYBDENUM":B$(42)="MO"
222 A$(43)="TECHNETIUM":B$(43)="TC":A$(44)="RUTHENIUM":B$(44)="R
U"
223 A$(45)="RHODIUM":B$(45)="RH":A$(46)="PALLADIUM":B$(46)="PD"
224 A$(47)="SILVER":B$(47)="AG":A$(48)="CADMIUM":B$(48)="CD"
225  A$(49)="INDIUM":B$(49)="IN":A$(50)="TIN":B$(50)="SN"
226 A$(51)="ANTIMONY":B$(51)="SB":A$(52)="TELLURIUM":B$(52)="TE"
227 A$(53)="IODINE":B$(53)="I":A$(54)="XENON":B$(54)="XE"
228 A$(55)="CESIUM":B$(55)="CS":A$(56)="BARIUM":B$(56)="BA"
229 A$(57)="LANTHANUM":B$(57)="LA":A$(58)="CERIUM":B$(58)="CE"
230 A$(59)="PRASEODYMIUM":B$(59)="PR":A$(60)="NEODYMIUM":B$(60)=
"ND"
231 A$(61)="PROMETHIUM":B$(61)="PM":A$(62)="SAMARIUM":B$(62)="SM
"
232 A$(63)="EUROPIUM":B$(63)="EU":A$(64)="GADOLINIUM":B$(64)="GD
"
233 A$(65)="TERBIUM":B$(65)="TB":A$(66)="DYSPROSIUM":B$(66)="DY"
234 A$(67)="HOLMIUM":B$(67)="HO":A$(68)="ERBIUM":B$(68)="ER"
235 A$(69)="THULIUM":B$(69)="TM":A$(70)="YTTERBIUM":B$(70)="YB"
236 A$(71)="LUTETIUM":B$(71)="LU":A$(72)="HAFNIUM":B$(72)="HF"
237 A$(73)="TANTALUM":B$(73)="TA":A$(74)="WOLFRAM":B$(74)="W"
238 A$(75)="RHENIUM":B$(75)="RE":A$(76)="OSMIUM":B$(76)="OS"
239  A$(77)="IRIDIUM":B$(77)="IR":A$(78)="PLATINUM":B$(78)="PT"
240 A$(79)="GOLD":B$(79)="AU":A$(80)="MERCURY":B$(80)="HG"
241 A$(81)="THALLIUM":B$(81)="TL":A$(82)="LEAD":B$(82)="PB"
242 A$(83)="BISMUTH":B$(83)="BI":A$(84)="POLONIUM":B$(84)="PO"
243 A$(85)="ASTATINE":B$(85)="AT":A$(86)="RADON":B$(86)="RN"
244 A$(87)="FRANCIUM":B$(87)="FR":A$(88)="RADIUM":B$(88)="RA"
245 A$(89)="ACTINIUM":B$(89)="AC":A$(90)="THORIUM":B$(90)="TH"
246 A$(91)="PROTACTIUM":B$(91)="PA":A$(92)="URANIUM":B$(92)="U"
247  A$(93)="NEPTUNIUM":B$(93)="NP":A$(94)="PLUTONIUM":B$(94)="P
U"
248 A$(95)="AMERICIUM":B$(95)="AM":A$(96)="CURIUM":B$(96)="CM"
249 A$(97)="BERKELIUM":B$(97)="BK":A$(98)="CALIFORNIUM":B$(98)="
CF"
250  A$(99)="EINSTEINIUM":B$(99)="ES":A$(100)="FERMIUM":B$(100)=
"FM"
251 A$(101)="MENDELEVIUM":B$(101)="MD":A$(102)="NOBELIUM":B$(102
)="NO"
252 A$(103)="LAWRENCIUM":B$(103)="LW":A$(104)="RUTHERFORDIUM":B$
(104)="KU"
290 A=RND(104):IF  B(A)=1 THEN  290
300 PRINT@  384,  "GIVE THE  SYMBOL  FOR ";A$(A);"            "
310 INPUT C$
320 IF C$=B$(A) THEN  500 ELSE 1000
500 B(A)=1:U=U+1:FOR X=11 TO U:SET (X,7):NEXT:FOR X=11 TO U:RESE
T(X,7):NEXT:IF U=115 THEN 1500
510 D=RND(5):ON D GOTO 520,530,540,550,560
520 PRINT@448, "LUCKY GUESS":GOTO 700
530 PRINT@448,"ARE YOU CHEATING?":GOTO700
540 PRINT@448,"KEEP UP THE  GOOD  WORK":GOTO700
550 PRINT@448,"PUT  THE PERIODIC  TABLE AWAY!!!":GOTO700
560 PRINT@448,"SHOW OFF":GOTO700
700 FOR Z=1  TO  600:NEXT:PRINT@448,STRING$(130," ");
990 GOTO 290
1000 V=V-3:FOR X=114 TO  V  STEP-1: SET  (X,7):NEXT:FOR X=114 TO
  V STEP-1:RESET (X,7):NEXT:IF  V<=10  THEN2000
1010 D=RND(5): ON  D  GOTO 1020,1030,1040,1050,1060
1020 PRINT@448, "OH NO":GOTO 1300
1030 PRINT@448,"NOT SO HOT":GOTO1300
1040  PRINT@448,"WHAT A DUMB GUESS":GOTO1300
1050 PRINT@448,"BETTER  STUDY":GOTO1300
1060 PRINT@448,"FOOLISH CHILD":GOTO1300
1300 FOR Z=1 TO 600:NEXT Z:PRINT@448,STRING$(130," ");
1310 PRINT@448, "THE CORRCT SYMBOL IS  ";B$(A):FOR Z=1 TO 600:NE
XT:PRINT@448,STRING$(130," ");
1490 GOTO  290
1500 CLS:PRINT@768,  "THANK YOU FOR HELPING CAPTAIN CHEMISTRY DE
FEAT THE EVIL ELEMENTS"
1501 PRINT@138,CHR$(136)+CHR$(183)+CHR$(191)+CHR$(187)+CHR$(132)
1502 PRINT@201,CHR$(191)+CHR$(175)+STRING$(3,191)+CHR$(159)+CHR$
(191)
1503 PRINT@264,CHR$(160)+CHR$(191)+CHR$(170)+STRING$(3,191)+CHR$
(149)+CHR$(191)+CHR$(144)
1504 PRINT@329,CHR$(131)+CHR$(168)+CHR$(191)+CHR$(131)+CHR$(191)
+CHR$(148)+CHR$(131)
1505 PRINT@394,CHR$(170)+CHR$(191)+CHR$(128)+CHR$(191)+CHR$(149)
1506 PRINT@456,STRING$(4,131)+CHR$(128)+STRING$(4,131)
1507 FORQ=1TO1000:NEXTQ:PRINT@160,"YOU HAVE EARNED YOUR CAPE";:F
ORQ=1TO1000:NEXTQ
1508 FORQ=10TO17:SET(26-Q,Q):SET(Q+23,Q):NEXTQ:FORQ=10TO19:SET(Q
,17):NEXTQ:FORQ=30TO39:SET(Q,17):NEXTQ
1510 GOTO2010
2000  CLS:PRINT@768,"BETTER STUDY!!!!!!!!!!!! THE CAPTAIN NEEDS
A FEW GOOD STUDENTS."
2010 FORQ=1TO3000:NEXTQ:CLS:INPUT"CARE TO TRY AGAIN";Z$:IFLEFT$(
Z$,1)="Y"THENRUN
2020 PRINT"GOODBYE, CAPTAIN CHEMISTRY NEEDS TO GO CLEAN HIS TEST
 TUBES NOW":FORQ=1TO2000:NEXTQ:CLS
```


See List of Advertisers on page 290

Celebrating In



Because your computer is

BUSINESS ANALYSIS

Business Analysis is a flexible, professional time series analysis and forecasting package. It features a unique graphing capability and can direct any chart or graph to your printer. Ideal for personal or business applications, *Business Analysis* can be used in sales and stock market analysis, forecasting, product or business planning, and family spending and energy consumption trends. Documentation included. Model I & III, Level II, 16K.

TRS-80,* Tape, #0140R-28...S$75.00
TRS-80,* Disk, #0152RD-28...$99.95

EASY CALC

Easy Calc is a unique program that will turn your computer into an electronic worksheet with all the speed and power of a programmable calculator. In addition to the regular calculator features, you can calculate percentages, summations of all rows of columns and other time saving functions. Your worksheets can be merged with Scripsit* word processor files for complete, professional reports. ENTER and SAVE your calculations with a touch of a single key! *Easy Calc*'s special library feature allows you to name each file with up to 28 characters. All options are MENU selectable and described in plain English. A detailed User's Manual is provided.

TRS-80,* Disk, #0369R3D-28 (Model III version)..........................$49.95
TRS-80,* Disk, #0269RD-28 (Model I version).............................$49.95

TLDIS/DLDIS

Tape and disk-based Labeling Disassemblers, *TLDIS* and *DLDIS* are three-pass disassemblers which assign labels (where appropriate) to routines in a machine-code program to give an output which is identical to a hand-assembled source code. Options enable you to send the disassembly to a lineprinter or to tape. *TLDIS* and *DLDIS* have many other special features and come complete with documentation. Model I & III, Level II, 16K.

TLDIS, TRS-80,* Tape, #0230R-28.......................................$14.95
DLDIS, TRS-80,* Disk, #0231RD-28......................................$19.95

THE ELECTRONIC BREADBOARD

The Electronic Breadboard permits the design and analysis of analog circuits. It can be used to evaluate voltages, currents, impedance and the frequency response of any circuit. This package is ideal for audio component repairmen, ham radio operators, hobbyists, electrical engineers, telecommunications engineers, audiphiles and students of electronics. It is a great training aid! Documentation is included. Model I & III, Level II, 16K.

TRS-80,* Tape, #0287R-28...$49.95

SUPER>>TERMINAL

The most flexible and powerful microcomputer terminal program available at any price, *Super>>Terminal*, allows you to communicate with almost any other computer system quickly and efficiently. It provides single key auto sign-on, file transfer capabilities which include prompted output and Xon/Xoff code recognition, and a continuous count of parity, framing and overrun errors. These and many more features make communications with *Super>>Terminal* fast, efficient and easy to use. Requires TRS-80* Model I or Model III with 32K memory and one disk drive. Documentation is included.

TRS-80,* Disk, #5700RD-28..$95.00

TYPING TEACHER

Transform your computer into both a typewriter and an instructor with this complete seven-part package. Includes on-screen diagrams and practice sessions. Your computer becomes a bottomless page for typing practice. The lessons are tailored to the keyboard, but train you to use any standard typewriter. Manual. Model I & III, Level II, 16K.

TRS-80,* Tape, #0099R-28...$12.95

SPACE SHUTTLE

Instant Software's *Space Shuttle* puts you in the command pilot's seat of America's first reusable space shuttle. Until commercial space flight becomes a reality, this is the closest you'll get to the ultimate flight. Complete with documentation. Model I & III, Level II, 16K.

TRS-80,* Tape, #0332R-28...$14.95

JET FIGHTER PILOT

After you've flown a few missions with the *Jet Fighter Pilot* package, you'll know you've earned your wings. *Jet Fighter Pilot* is a brilliantly realistic simulation of a real mission putting you in total control of the aircraft. All controls respond the same as they would on a real jet fighter. This combat flying program comes with comprehensive documentation. Model I & III, Level II, 16K.

TRS-80,* Tape, #0159R-28...$14.95

DANGER IN ORBIT

Danger in Orbit is a real-time, machine-code space game with variable levels of difficulty, superb high-speed graphics, sound effects and automatic scorekeeping. Destroy asteroids with your anti-matter cannon! Prevent alien spacecraft from invading the Terran Defense Network. Avoid alien attacks from hyperspace. Only lightning reflexes and nerves of steel will enable you to survive *Danger In Orbit*. Comes with documentation. Model I & III, Level II, 16K.

TRS-80,* Tape, #0237R-28...$14.95

SKYBOMBERS II

Skybombers II is an aerial war game pitting two nations against each other. Battlesounds and graphically displayed explosions give each battle a vivid reality. The scores for both countries are constantly updated at the bottom of the display screen. Experience air warfare at its best with *Skybombers II*. Requires 32K RAM, Applesoft in ROM and game paddles. Documentation is included.

APPLE,* Tape, #0183A-28...$9.95
APPLE,* Disk, #0271AD-28...$19.95

BALL TURRET GUNNER (with sound)

Ball Turret Gunner is a deep space, laser attack game. As a member of the *Ball Turret Gunner* service you're at the console of the spaceship: your mission—destroy the enemy! Multiple levels of difficulty, optional sound effects, superb graphics and complete documentation enhance this program. Model I only.

TRS-80,* Tape, #0051R-28..$9.95

only as good as the software

GEOGRAPHY EXPLORER:USA

Learn the vital facts about each state in our union in the most fun and fascinating way yet. The program displays computer generated maps of the United States, its seven regions and even its individual states. Questions may be multiple choice, recognition or fill-in format. A unique TEACHER mode allows multiple options of how the material is to be presented. As an added bonus, *Geography Explorer:USA* has the capability of supporting a light pen. The union of sophisticated software with the speed and novelty of the light pen represents a milestone in Computer Assisted Instruction. Fully documented, this program is educational for any age group. Model I or III, Level II, 16K.

TRS-80,* Disk, #0071RD-28. . . . . $49.95

MASTER REVERSI

Master Reversi is a state-of-the-art game that's simple to play, yet infinitely challenging. It is a tournament-winning program that has more features than any other reversi program on the market. *Master Reversi* offers a special library of tournament-level games for you to analyze. Whether you're having fun or studying for the world championship, *Master Reversi* is the game for you. Documentation included. Model I, Level II, 16K.

TRS-80,* Disk, #0378RD-28. . . . . $29.95

AIR FLIGHT SIMULATION

An actual simulated take off, flight and landing, *Air Flight Simulation* is both fun and educational. Basic flight instructions, explanations of basic aerodynamics, principals of flight and well illustrated documentation included. TRS-80* version requires Model I or III, Level II, 16K. APPLE* version requires 16K RAM and Applesoft.

TRS-80,* Tape, #0017R-28. . . . . $9.95
APPLE,* Tape, #0148A-28. . . . . $9.95

COSMIC PATROL

With *Cosmic Patrol* you get the best of the search and destroy space games. Impressive sound option and superb graphics make *Cosmic Patrol* the best of its kind. Defend your interstellar space and prey on the enemy with this fast, real-time action game program. Documentation included. Model I & III, Level II, 16K.

TRS-80,* Tape, #0223R-28. . . . . $14.95
TRS-80,* Disk, #0224RD-28. . . . . $19.95

SANTA PARAVIA AND FIUMACCIO

Build a powerful kingdom from your tiny city-state. Set in the year 1400 AD, in Italy, this challenging program allows you to test all your leadership skills. See if you have what it takes to be a wise and competent ruler. Well documented. TRS-80* version Model I & III, Level II, 16K. APPLE* version requires Applesoft in ROM.

TRS-80,* Tape, #0043R-28. . . . . $9.95
APPLE,* Tape, #0174A-28. . . . . $9.95
APPLE,* Disk, #0229AD-28. . . . . $19.95

Here's how to order:

RETAIL: Ask us for a list of the Instant Software dealers nearest you. If there is not one near you, let us know. Meanwhile, if you would like to order any of the products featured on these pages call toll-free 800-258-5473, or use the handy reply card in this magazine (a photocopy of it is acceptable). Just fill in the necessary information and return it to us. To receive a free copy of our latest catalog, just ask, or check the appropriate box on the reply card. Visa, Mastercard, American Express and personal checks or money orders are accepted. Sorry, no C.O.D.'s. Please add $2.50 per order for postage & handling.

DEALERS: If you would like to join the hundreds of Instant Software dealers in the United States, Canada and Europe who are already celebrating our software, fill in the coupon below (a photocopy is acceptable) or call toll-free 800-258-5473 and ask for our dealer sales department.

*TRS-80, and Scripsit are trademarks of Tandy Corp. APPLE is a trademark of Apple Computer Co.

Instant Software™

A SUBSIDIARY OF WAYNE GREEN INC., PETERBOROUGH, NH 03458

I want to join the celebration!

As an Instant Software dealer you can open up a new world of software for your customers. You'll receive the finest business, utilities, applications, games, simulations and educational software for most popular microcomputers. Find out how you can increase your profits as an Instant Software dealer. Inquire today—you owe yourself a celebration.

☐Call me directly.
☐Send me dealer information.

Dealer name ______________________
Address ______________________
City ____________ State ______ Zip ______
Your name ______________________
Telephone (AREA CODE) ______________________

C-28

Instant Software, Inc., Peterborough, NH 03458

Classroom networking comes of age.

Earth to Class, Listen Up!

Madeleine Fish
Grant Union High School
1400 Grand Avenue
Sacramento, CA 95838

Radio Shack's Network Controller ($500) makes it possible to use up to sixteen computers synchronously in a classroom.

The heart of the network is the controller box. On the backside are sixteen positions for plugging in connector cords. The other end of each cord connects to the tape position on each of the student computers. The controller box also connects to the tape output port on the teacher's computer. The teacher first loads a program from a disk or tape into the instructor's computer. The teacher then CSAVEs the program, sending it out through the tape port to the controller box. This amplifies the signal and sends it to the tape ports of the students' computers which have been set up with a CLOAD command. In this way the teacher can load sixteen systems simultaneously with the same program.

> **The Key Box**
>
> **Basic Level II**
> **Model I**
> **16K**
> **Radio Shack**
> **Network Controller**

A network system also makes communication in the opposite direction possible. Student stations selected via a numbered dial on the controller box can send programs to the teacher's computer. Using the system in this manner the teacher can save programs written by a student on the main system's floppy disk or obtain a hard copy on a printer connected to the instructor's computer system.

At Grant Union High School in Sacramento we have eleven TRS-80s (16K Level II) connected through a network system. We have used the computers in the math and science programs. The students write their own programs and perform drill exercises on the computer. The school library also contains three systems available for student use.

We have used the computer network to assist our students in mastering all levels of science instruction from general science through physics. College preparatory and basic level students also use the network. They work in groups of two or three per station.

In the physical sciences much of the material presented to the students requires that they practice solving problems. We have observed that students attempt many more problems when presented with them one at a time on the computer, compared to the traditional approach involving a page of problems. A student faced with a great many problems is often discouraged before attempting the first exercise. A computer provides a true second chance to solve a problem. The computer can inform a student that a solution is incorrect and provide a hint to help with a new approach.

During the past few years an increasing amount of commercial software has been directed toward the education market. It is still, however, difficult to find programmed material tailored to a particular school's curriculum or written at the target level. Finding funds for software is often a major problem for the classroom teacher. A teacher using computers for instruction must therefore be prepared to author many lessons.

Several moderately priced authoring programs allow a teacher to enter questions and answers. When the student uses these programs they require a two-step loading process. A general program is first loaded; the computer can then accept the second load containing the question and answer data. This procedure is not directly possible on the network system. The software requires considerable modification before it can be used with a network of computers.

Educators who are novices in computer programming, but wish to develop their own com-

COMPUTER BOOKS FOR BEGINNERS

Everything you need to know to get started programming your own computer. These handy program sourcebooks, each jam-packed with easy-to-understand info for beginners, are crammed with hundreds of tips, tricks, secrets, hints, shortcuts, and techniques, plus hundreds of tested ready-to-run programs. TRS-80 Color Computer. TRS-80 Pocket Computer. Sharp PC-1211 Pocket Computer. Casio FX-702P Pocket Computer. Four of the most popular computers for beginners.

Color Computer

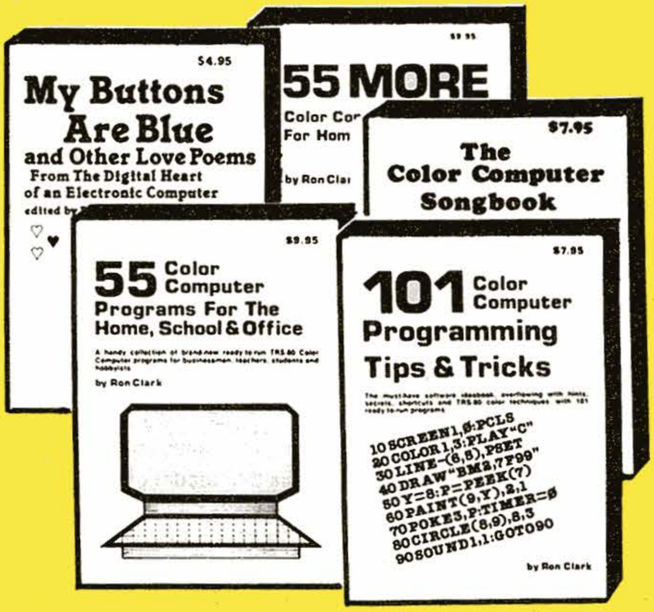
101 Color Computer Programming Tips & Tricks, learn-by-doing instructions, hints, secrets, techniques, shortcuts, insights, for TRS-80 Color Computer, 128 pages **$7.95**

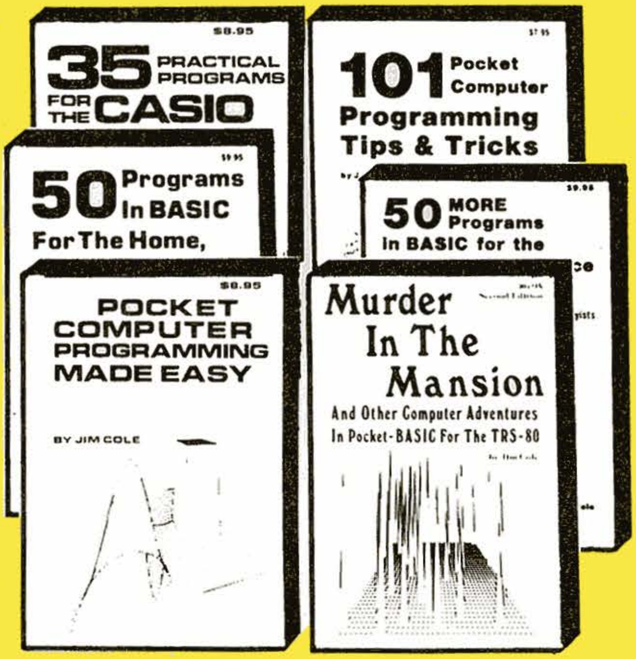
55 Color Computer Programs for the Home, School & Office, practical ready-to-run software with colorful graphics, 128 pages **$9.95**

55 MORE Color Computer Programs for Home, School & Office, sourcebook of useful, plug-in-and-run software with colorful graphics, for TRS-80 Color Computer, 112 pages **$9.95**

The Color Computer Songbook, 40 favorite pop, classical, folk & seasonal songs arranged for the TRS-80 Color Computer, ready-to-run music programs, 96 pages **$7.95**

My Buttons Are Blue And Other Love Poems From The Digital Heart Of An Electronic Computer, for poetry lovers, computer lovers, just-plain lovers, a high-tech classic, 66 heartwarming poems written by a TRS-80 Color Computer, great gift for someone close to you, includes the exclusive ARCsoft Poetrywriter™ program, 96 pages **$4.95**





Pocket Computer

50 Programs in BASIC for the Home, School & Office, useful plug-in-and-run software for TRS-80/Sharp PC-1211 pocket computers, 96 pages **$9.95**

50 MORE Programs in BASIC for Home, School & Office, sourcebook of tested ready-to-run software for TRS-80/Sharp PC-1211 pocket computers, 96 pages **$9.95**

101 Pocket Computer Programming Tips & Tricks, secrets, hints, shortcuts, techniques from a master programmer, for TRS-80/Sharp PC-1211 pocket computers, 128 pages **$7.95**

Murder In The Mansion and Other Computer Adventures, murder mystery, space, adventures, loads of fun, 24 game programs for TRS-80/Sharp PC-1211 pocket computers, 96 pages **$6.95**

35 Practical Programs for the Casio Pocket Computer, useful plug-in-and-run software for the FX-702P, 96 pages **$8.95**

Pocket Computer Programming Made Easy, new fast and easy way to make the world's smallest computer work for you, 128 pages **$8.95**

Program Worksheets

Handy printed forms make writing BASIC software easy and fun. Customized for your own computer system, or use the universal form good for any BASIC computer. 40-sheet tablets:

Color Computer Coding Form	**$2.95**
Pocket Computer Coding Form	**$2.95**
APPLE Computer Coding Form	**$2.95**
IBM Personal Computer Form	**$2.95**
Universal Coding Form	**$2.95**

Electronics Projects

25 Easy-To-Build One-Night & Weekend Electronics Projects, useful gadgets, readily-available parts, 96 pages **$4.95**

Order direct from this ad. Send check or money order. Include $1 shipping for *each* item ordered up to a maximum of $3. Or write for our free catalog. Mail orders to:

ARCsoft Publishers

Post Office Box 132Q
Woodsboro, Maryland 21798

(301) 663-4444

491

Customers outside North America wanting airmail send $6 per book. Foreign customers pay in U.S. dollars. For first-class mail delivery in North America include $2 per book. Maryland residents add 5% sales tax. Sorry, no COD. TRS-80 is a trademark of Tandy Corp./Radio Shack.

puter assisted instruction materials will find the program presented here useful. This program is easy for the teacher or parent to use and loads easily from tape or disk with a network configuration or a single computer.

The following program allows an instructor to write a series of questions, with one hint, in a succession of data statements. The answers may be numerical or alphabetic characters.

This program is very simple. A teacher with beginning programming skills can understand its logic and tailor the basic model to his own teaching situation.

The program reads the question, hint (if the first answer is incorrect) and answer from data statements. If the answer is a string of alphabetic characters it must agree exactly with the teacher's answer. If the answer is numerical, a factor of plus or minus 10 percent triggers a correct answer message. This can be easily changed. The student's score is displayed on the screen at the end of the program. The student can recycle through the questions or end.

The first data line contains the name of the exercise and the total number of questions. The format for entering the data is question, answer and hint separated by commas. Do not use commas or colons in the question, hint or answer.

You can enter the intial text material and instructions with a series of print statements. You might make the data statements the initial part of the program. Then delete the program beyond the data, text and instructions for storage and merge it with the body of the program for use.

If you use this program to prepare a computational exercise, you can enter the computer's calculator mode by Breaking the program and using the print command. Enter CONT to proceed with the program.■

```
10 REM PROGRAM TO FORMULATE A DRILL EXCERCISE
20 REM ALL QUESTIONS AND ANSWERS WILL BE ENTERED THROUGH DATA ST
ATEMENTS.
30 CLEAR 3000   'CHANGE DEPENDING ON NUMBER OF QUESTIONS
40 DIM Q$(20),A$(20),H$(20)     'SET FOR MAXIMUM OF 20 QUESTIONS
50 CLS
60 INPUT"FIRST NAME";N$
70 INPUT"LAST NAME";L$
80 INPUT "DATE(M/D/YR)";D$
500 CLS: PRINT"   TEXT MATERIAL"
1400 INPUT"PRESS ENTER WHEN READY TO GO ON";ZZ
1500 CLS: PRINT"  INSTRUCTIONS FOR COMPUTER PROGRAM"
1900 INPUT "PRESS ENTER WHEN READY TO GO ON";ZZ
2000 REM PROGRAM FOR RETRIEVING QUESTIONS AND ANSWERS FROM DATA
STATEMENTS.
2010 RESTORE:K=0
2020 READ P$,Z          'PROGRAM TITLE AND NO. OF QUESTIONS
2030 FOR X = 1 TO Z
2040 READ Q$(X),A$(X),H$(X)   'READS QUESTION, ANSWER AND HINT
2050 CLS
2060 PRINT@64, Q$(X);
2070 INPUT R$       'R$ IS STUDENT RESPONSE
2080 IF VAL(R$)=0 THEN 2110  'RESPONSE WAS A WORD
2090 IF ABS(VAL(R$)-VAL(A$(X)))<=.1*VAL(A$(X)) THEN GOSUB 5000:N
EXT X:GOTO 2130  'ALLOWS +/- 10% ON NUMERICAL ANSWER
2100 GOSUB 6000 :NEXT X:GOTO 2130
2110 IF R$=A$(X)THEN GOSUB 5000:NEXT X:GOTO 2130
2120 GOSUB 6000:NEXT X:GOTO 2130
2130 CLS:PRINT:PRINT:PRINT N$;" ";L$;"             ";D$"
";P$:PRINT
2140 PRINT "YOU SCORED "K" CORRECT OUT OF "Z" ON THE FIRST TRY":
PRINT:PRINT:PRINT"YOU HAD "(K/Z)*100"% CORRECT"
2150 PRINT: INPUT" DO YOU WISH TO REPEAT THESE QUESTIONS";C$
2160 IF LEFT$(C$,1)<>"Y" THEN END
2170 GOTO 2010
2180 GOSUB 6000:NEXT X
5000 REM CORRECT ANSWER SUBROUTINE
5010 PRINT@ 704, "CORRECT ANSWER ";N$:K=K+1
5020 FOR Y=1 TO 700:NEXT Y
5030 RETURN
6000 REM WRONG ANSWER ROUTINE
6010 PRINT@ 448,"TRY AGAIN- HERE IS A HINT-- ":PRINT H$(X)
6020 INPUT"ANSWER";R$
6030 IF VAL(R$)=0 THEN 6060
6040 IF VAL(R$)=0 AND A$(X)=R$ THEN 6100
6050 IF ABS(VAL(R$)-VAL(A$(X)))<= .1*VAL(A$(X)) THEN 6100
6060 IF R$=A$(X) THEN 6100
6070 PRINT@ 704,"ANSWER IS  ";A$(X)
6080 INPUT"HIT ENTER KEY WHEN READY TO GO ON";ZZ
6090 RETURN
6100 PRINT@704,"THATS USING ";N$;"'S BRAIN POWER"
6110 FOR Y=1 TO 600:NEXT Y
6120 RETURN
7000 DATA CHAPTER 1,2
7010 DATA WHAT IS THE MOLECULAR WEIGHT OF WATER,18,USE PERIODIC
TABLE FOR ATOMIC WEIGHTS OF H AND O.
7020 DATA WHAT IS THE NAME OF THIS COURSE,CHEMISTRY,CHECK THE TI
TLE OF YOUR TEXT BOOK
```

Program Listing

Education = MicroComputers.

E = MC*

*to the power of Instant Software

Einstein's equation helped explain the properties of relativity, but didn't do much to speed the learning process. Instant Software's equation *means* individual attention and greater comprehension. Everyone knows the best student-teacher ratio is 1:1. That was a problem in years past, but now it's as easy as C.A.I. (Computer-Aided Instruction). These Instant Software programs will guide and teach each lesson with the infinite patience only a computer can provide, thus allowing students to progress at their own pace.

Einstein had the right idea, but if he'd only known then what we know now. . . . . . .

REVISED Typing Teacher *(for the TRS-80™)**

A complete seven-part package that guides you from familiarization of the keyboard through typing words and phrases to mastery of touch-typing. Your video monitor becomes a bottomless page for typing practice, and your own private TYPING TEACHER, ready to teach when you're ready to learn. Model I, Level II, 16K; Model III, 16K. **Order No. 0099R -25 $12.95** plus $2.50 shipping.

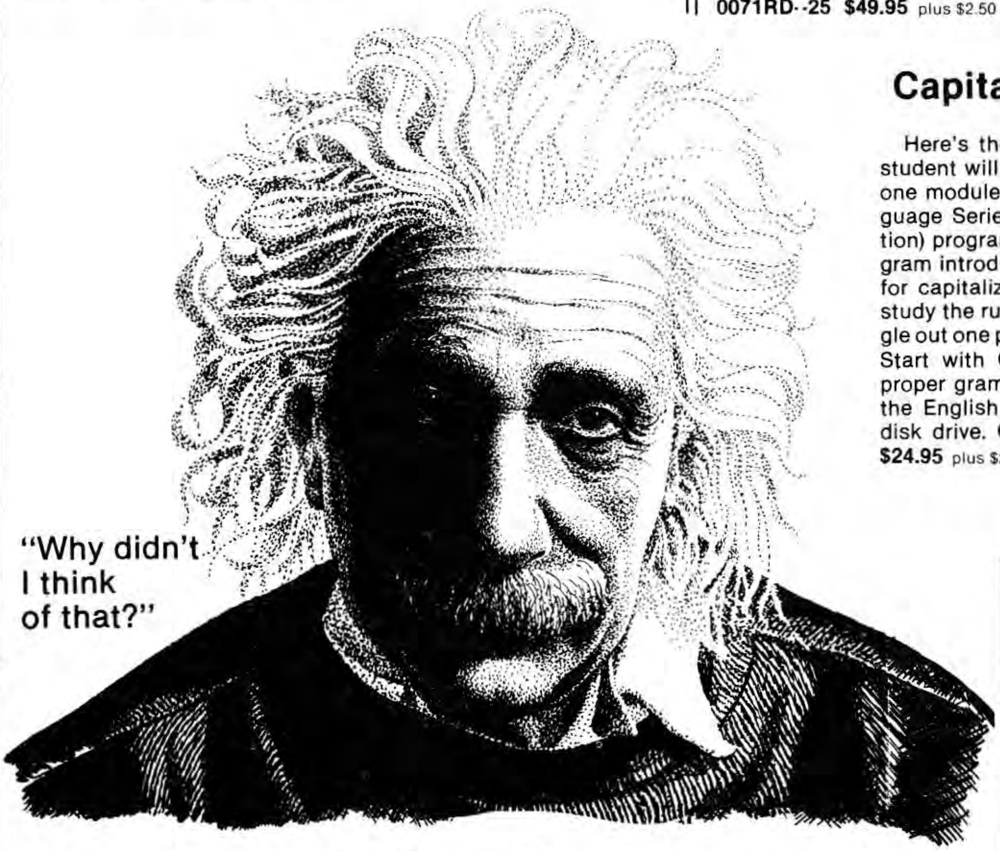
Triangle Trig *(for the TRS-80™)**

Instant Software is proud to offer TRIANGLE TRIG, the latest addition to our educational software inventory. This tutorial package is designed for everyone involved in learning mathematics from the junior high school level on up. It features two programs that are introductions to the mathematics of triangles, an essential component of the engineering sciences.

TRIANGULAR GEOMETRY includes twelve lessons that carefully teach all of the basic concepts that must be understood in order to deal with right and similar triangles. BEGINNING TRIGONOMETRY explains tangents, sines and cosines, and allows the student to use his new knowledge in practical situations. MOD I, 16K Level II, Mod III, 16K. **Order No. 0309R 25 $19.95** plus $2.50 shipping.

Geography Explorer: U.S.A. *(for the TRS-80™)**

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. Model I, Level II 16K, expansion interface with 16K, one minidisk drive. **Order No. 0071RD--25 $49.95** plus $2.50 shipping



"Why didn't I think of that?"

Capitalization *(for the Apple™)***

Here's the most patient grammar teacher a student will ever have. . . CAPITALIZATION. It's one module among several in the English Language Series of CAI (Computer Aided Instruction) programs from Instant Software. This program introduces and exercises the twelve rules for capitalization in English. The student may study the rules in order from 1 to 12, or may single out one particular rule for study and practice. Start with CAPITALIZATION and graduate to proper grammar and a better understanding of the English language. Applesoft Basic 32K, 1 disk drive. **Order No. 0339AD--25 (disk-based) $24.95** plus $2.50 shipping

TO ORDER:
See your local
Instant Software dealer
or call toll-free
1-800-258-5473
orders only
In New Hampshire
1-603-924-9471
Mon.-Fri. 8:00 am-4:30 pm E.S.T.

Instant Software™

~531

Peterborough, N.H. 03458 USA   A subsidiary of Wayne Green Inc.

*TRS-80 is a trademark of Radio Shack division of Tandy Corp.
**Apple is a trademark of Apple Computer Company

Don't lose your cool because the power company blows a fuse.

Battery Backup

Howard F. Batie
12002 Cheviot Drive
Herndon, VA 22070

A recent project involving the use of a TRS-80 computer to aid the handicapped has the need to provide a battery backup. The circuit described here features automatic battery takeover if the 110 volt line fails, and as an extra bonus, a cooler-running TRS-80 both under 110 volt and battery operation.

Many nursing homes, hospitals and special education centers have emergency power lines or generators which are exercised at regular intervals. When this is done, there can be a very brief period (fractions of a second to a few seconds) when the power is off altogether while switching to the alternate power. If computers are used as an educational/therapeutic tool or as a learning aid the power must be constant or the memory will be wiped out and the program will bomb. If tape or disk is used for storage, the program can always be reloaded; however, this can be a frustrating process for someone who is totally unfamiliar with a computer. Additionally, any files or information which were not yet saved on a permanent storage medium will be lost. This TRS-80 battery backup was designed to keep the computer memories up while switching over to the backup power source.

Operation

The TRS-80 battery backup circuit shown in Fig. 1 is built around Digital Innovations' DC-80 power converter designed specifically for the TRS-80. To trace its operation both from the 110 volt line and from the battery, it is easiest to consider each case separately. First, assume that the battery B1 is removed from the circuit. A standard 110 volt to 12.6 volt two-ampere filament transformer is connected to a full wave rectifier bridge formed of D1–D4. The output of this bridge, when filtered by C1, provides approximately 19 volts DC with no load, and about 15 volts with a load current of 1 ampere. D5–D8 provide a 1.4 volt drop which is essentially independent of the current drawn through the diodes ($2 \times 0.7 = 1.4$ volts). Thus, the 13.6 volt potential at point A would provide the input power for the DC-80.

Now if diodes D5–D8 are removed, and the battery is placed back in the circuit, the battery terminal voltage is provided to the DC-80 through diodes D9 and D10, giving an input voltage to the DC-80 of 11.9 volts ($13.6 - 0.7 = 11.9$). If D5–D8 are replaced, the 13.6 volt potential at point A from the AC supply is greater than the 11.9 volt potential from the battery; D9 and D10 are therefore back-biased and no current flows through them from the battery. Instead, the AC supply furnishes all power to the DC-80 and TRS-80. But if the 110 volt line fails, point A would fall to zero volts if it were not connected to D9 and D10. When the AC supply voltage at point A falls below 11.9 volts, the battery automaticlly takes over and furnishes current for the DC-80.

The relationship of the input voltage to the DC-80 as a function of the size of the filter capacitor C1 and the number of voltage-dropping diode pairs is shown in Fig. 2. The DC-80 input voltage from the AC power supply has both a DC and AC component. The DC component is primarily a function of the number of diode pairs used to drop the voltage of the full wave bridge output; the value of the filter capacitor C1 does not have a marked effect on the average DC voltage component above a value of about 1000 microfarads. However, since the DC-80 must furnish about 1 ampere to power the TRS-80, the AC (RMS) voltage component is dependent on the size of C1 (see the curved line of Fig. 2).

In order to prevent the current from being drawn from the battery while the DC-80/TRS-80 is being powered from the AC supply, the voltage at point A in Fig. 1 must remain above 11.9 volts. This means that the AC supply must furnish at least a voltage of the average DC voltage minus the AC component. With a 2200 microfarad capacitor for C1, no more than three diode pairs should be used with a 13.6 volt battery.

The diodes used are 50-volt silicon rectifiers rated at 3 amperes. In the rectifier bridge they run warm, but not hot to the touch, since current flows through each diode for only half of the duty cycle. However, in the voltage dropping diodes, DC current is flowing continuously. If D5 and D7 are removed, D6 and D8 are very hot to the touch after a few minutes. By adding D5 and D7 in parallel with D6 and D8, the current through each diode is halved, and they also run warm, but not hot. The same consideration applies to D9 and D10.

The length of time the TRS-80 can be

"...one DC-80 will power either a TRS-80 or an Expansion Interface..."

powered from the battery is determined by the battery ampere-hour rating. For instance, if a 100 A-h battery is used, about 100 hours of service is possible if the battery starts out fully charged. The DC-80 can accept input DC voltages down to about 9 volts. Since recharging the battery while it is connected to the DC-80 can damage the DC-80, disconnect the battery from the circuit of Fig. 1 before recharging the battery.

Construction

The components shown in Fig. 1 are assembled on a small perfboard which along with the transformer and DC-80 is housed in a single ventilated enclosure. The on/off switch S1 and AC power LED are optional. The 110 volt line is fused by F1, a panel-mounted half-amp fast-acting fuse. F2 is part of the DC-80 as supplied and should be retained. The DC-80 output DIN plug goes directly into the TRS-80 power jack, and the original TRS-80 power supply is no longer needed.

One final note: one DC-80 will power either a TRS-80 or an Expansion Interface (EI), but not both. If you intend to provide battery backup to both the TRS-80 and an EI, two DC-80's will be required. In addition, heftier components for Fig. 1 will be needed to handle double the current. In this case, I'd recommend that a 6-amp molded full wave bridge rectifier be used instead of diodes D1–D4, a 12.6 volt at 5A transformer for T1, at least three silicon 3A diodes in parallel for each diode pair shown in Fig. 1, and a computer-grade filter capacitor of at least 6000 microfarads rated at 15 volts DC.■

The DC-80 is available for U.S. $49.95 plus U.S. $3.00 per unit for shipping from
Digital Innovations
37 Stonybrook Drive
Kitchener, ONT N2M 4L6 Canada

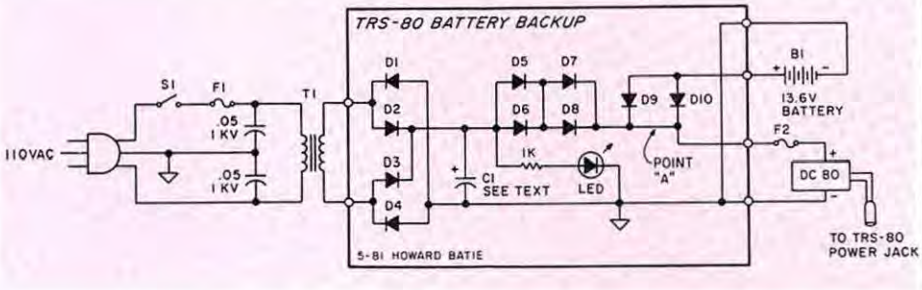


Fig. 1. Schematic Diagram.

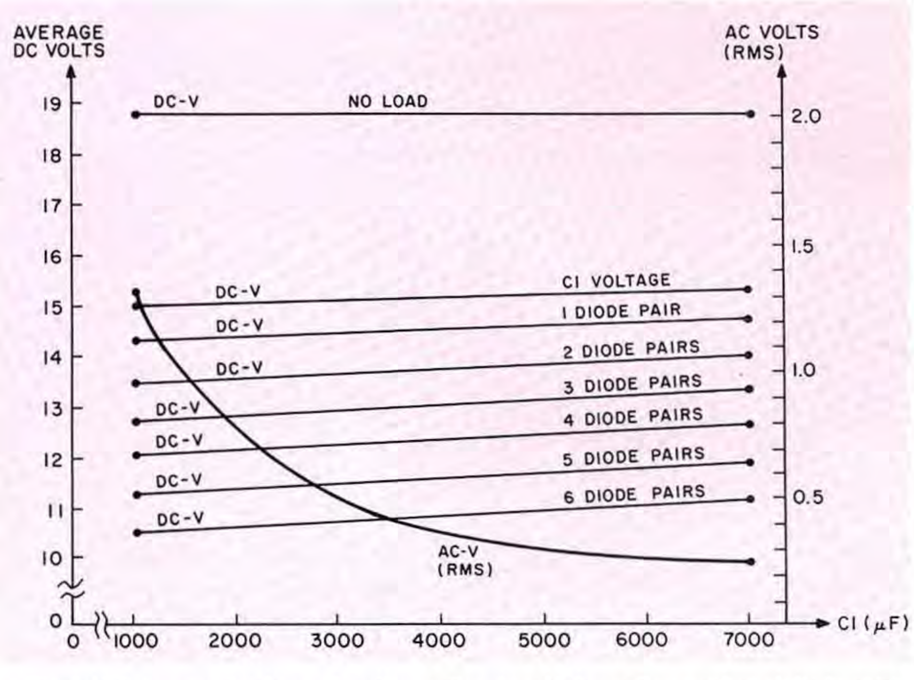


Fig. 2. Voltage at point A as a function of the size of C1 and the number of diode pairs.

✓See List of Advertisers on page 290

Microtechnology is making more possible for the physically handicapped.

Making More Possible

by Kerry Leichtman
80 Microcomputing *staff*

The personal computer is many things to many people. To the businessman it is an instrument of efficiency, to the games player it is a way to grope through dank underground corridors while hunting demons. The computer's uses are as varied as its users: hobbyist, economist, writer, scientist—the list goes on and on. Most generally, the machine is a resource of available programmed information manipulated by the operator to obtain certain results. The operator telling the machine what to do is the normal human to computer relationship.

The normal relationship? A computer manipulating a person must then be an abnormal relationship. Seems like it should be. But really, who is telling who what to do when a flashing prompt translates into, "You didn't shut the disk door properly, stupid."? Or who is controlling things when suddenly you find yourself, powerless, drifting in a distant galaxy surrounded by hostile alien warships? Although the computer would be nothing more than an expensive paperweight without human control, its use as a functional extension to do what a human cannot could be its greatest asset.

The advent of the microcomputer has transformed this electronic technology from specialized super-calculators into a powerful teaching tool. School systems are taking advantage of the microcomputer's convenience to help students grasp everything from English grammar to abstract equations. As an educational medium for ordinary students, the computer is proving itself to be efficient, productive and motivating. As an educational medium for extraordinary students, the physically handicapped, the computer is new hope, new independence and new opportunity toward lessening the gap between what they can and cannot achieve.

A New Age

Like everything related to computers, application for the handicapped is a new concept. The days of computer operators in long white lab coats scurrying around a maze of huge mainframes, carefully noting functions and data on clipboards, has long been over. It is no coincidence that as machines became more compressed, faster and less expensive, so did the white glove approach to them. Computer technology became accessible to virtually everyone—including the handicapped.

Johns Hopkins University, along with Radio Shack and the National Science Foundation, recently seized this opportunity of accessibility by encouraging development of handicapped applications through a contest: "Personal Computing to Aid the Handicapped. The Johns Hopkins First National Search." The fact that these three organizations have combined forces should be proof enough that microcomputers and the handicapped have a future together. The fact that the contest was sub-billed as being the *first* national search clearly indicates the organizers expect the relationship to be worthy of continuing effort.

Programs and hardware will undoubtably become more sophisticated as time goes on. But, as this is the start of the coupling of computers and the handicapped, the obvious has to be dealt with first. Some of mankind's greatest inventions and concepts did not exist until someone said, "Hey, I could drag this load further and easier if it had wheels." "What are wheels?" his less imaginative friend might have asked in reply. The would-be inventor could not describe the wheel as looking like a donut, as they did not yet exist; he had to invent the wheel. More realistically, you might not be reading this had Gutenberg not invented moveable type.

The handicapped person, with crippling disabilities such as cerebral palsy, does not have the ability to type with two fully functional hands. In some cases, limited to one or two fingers, a mouthpiece, or headgear, the handicapped computer user has the ability to poke only one key at a time. Conventional typing courses, however, require full use of eight fingers.

Ruth Lambert, Inventor

While it may seem obvious, until Ruth Lambert invented it, One Hand Touch Typ-

continued on page 138

Photo 1. Joel Stevey uses a magnet strapped to his watch to select the components of his message.

✓See List of Advertisers on page 290

H & E COMPUTRONICS INC.

• • EVERYTHING FOR YOUR TRS-80™ • ATARI™ • APPLE™ • PET™ • CP/M™ • XEROX™ • IBM™ • •

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. - * ATARI is a trademark of Atari Inc. - * APPLE is a trademark of Apple Corp. - * PET is a trademark of Commodore
* CP/M is a trademark of Digital Research - *XEROX is a trademark of Xerox Corp. - * IBM is a trademark of IBM Corp.

NEW

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software

BUSINESS PAC 100

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual.....
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

	NAME	DESCRIPTION
1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNU1	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAP1	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ CASSETTE VERSION	$99.95
☐ DISKETTE VERSION	$99.95
☐ TRS-80* MODEL II VERSION	$149.95

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

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ASK FOR OUR 64-PAGE CATALOG

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

24 HOUR ORDER LINE
(914) 425-1535

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE***

✓9

ACCOUNTS PAYABLE

Processes

- ★ Add, change or delete records within the Vendor File.
- ★ List the Vendor File.
- ★ Enter vouchers.
- ★ Automatically determine which vouchers to pay.
- ★ Print checks and a Check Register.
- ★ Produce the following reports:
  - (1) Open Voucher Report
  - (2) Accounts Payable Ageing Report
  - (3) Cash Requirements
- ★ At the end of a month, prepare the General Ledger Transfer File, passing the following information for each debit or credit transaction:
  - (1) Account Number
  - (2) Description
  - (3) Source Code
  - (4) Date
  - (5) Amount

File Information

There are two main computer files maintained within the Accounts Payable System, the Vendor File and the Voucher File.

VENDOR FILE
- Vendor Code
- Vendor Name
- Address
- Phone
- Year-To-Date Purchases
- Year-To-Date Payments
- Current Balance
- Last Payment
- Date of Last Payment
- Monthly Entry Flag
- Due Date of Month
- Debit Account Number
- Amount (Debit)
- Month Last Paid

This file may also contain information to enable generation of automatic vouchers for those items such as rent or bank payments that are paid every month.

VOUCHER FILE
- Voucher Code
- Voucher Date
- Amount Due
- Date Due
- Discount Percent
- Discount Amount
- Discount Date
- Invoice Number
- Invoice Date
- Status

Plus up to six account number-amount fields for General Ledger account numbers to which the amount due is to be distributed.

PAYROLL

Processes

- ★ Add, change or delete records within the Employee File.
- ★ List the Employee File.
- ★ Modify the Tax Information Files.
- ★ At the end of a pay period -
  - (1) Calculate Pay
  - (2) Print Checks
  - (3) Print Payroll Register
- ★ At the end of a month -
  - (1) Print the monthly summary
  - (2) Print the Unemployment Tax Report
  - (3) Prepare the General Ledger Transfer File, passing the following information:
    - Net Pay (Cash)
    - Employee FICA Withheld
    - Federal Tax Withheld
    - Insurance Deductions
    - Miscellaneous Dedutions
    - State Tax Withheld
    - Local Tax Withheld

  The gross pay for up to twenty payroll departments may also be passed to the General Ledger.
- ★ At the end of a quarter, print the 941A report information.
- ★ At the end of a year, print the W-2 forms.

File Information

There are two main computer files maintained within the Payroll System, the Employee Master File and the Tax File.

EMPLOYEE MASTER FILE
- Name
- Address
- Local Code
- State Code
- Marital Status
- Exemptions, Federal
- Exemptions, State
- Social Security Number
- Pay Period
- Pay Type
- Pay Rate
- Insurance Deduction
- Miscellaneous Deduction
- Date Employed
- Date Terminated
- Last Check Information

Payroll (con't)

And current, month-to-date, quarter-to-date and year-to-date totals for:
- Regular Earnings
- Overtime Hours/Earnings
- Other Hours Rate/Earnings
- Commission Earnings
- Miscellaneous Income
- FICA Deductions
- Federal Deductions
- State Deductions
- Local Deductions
- Insurance Deductions
- Miscellaneous Deductions

TAX FILE
- (for single and married persons)
- Federal Tax Information Tables
- State Tax Information Tables
- Local Withholding Tax Information Tables

An Overview of the Inventory System

Inventory is probably the most speculative of all of a company's assets. A true measure of the effectiveness of management is the ability with which it supervises the inventory control function.

The Peachtree Software™ Inventory Management System is designed to (1) give you better merchandise control, (2) allow you to lower your dollar investment in inventory, and (3) improve customer service and response.

The System maintains detailed information on each inventory item including the part number, description, unit of measure, vendor and reorder data, item activity, and complete information on current item costs, pricing, and sales. Transactions effecting inventory (sales, receipts, adjustments) may be applied at any time to insure the inventory data is always up to date and accurate.

As with all Peachtree products, the system is interactive, simple to operate, and provides reports that are up to date and comprehensive.

Particular features of the Peachtree Software™ Inventory Management System include:

- Interactive, menu-driven programs
- Self-instructing user documentation
- Long item number - up to 15 characters
- Departmentalizing of items
- Multiple pricing levels
- Processes items on reserve (committed but still in stock)
- Online item query at any time
- Comprehensive management reporting
- Automatic month end file backup
- Recovery routines for hardware failures
- Sample data for demonstration and training

How the System is Designed

The Inventory Management System operates with an **Inventory Master File** which allows for the creation of each inventory item and for the recording of transactions (sales, receipts, returns, reserves, and adjustments) to each inventory item.

The Inventory Master File contains the item number, description and various other data on item costs, prices, reorder levels, vendor refereence, and activity. The items within the Master File are entered, changed, deleted, and queried through the **Inventory Master File Maintenance** program. All data on all items may be listed by using the **Detail Inventory Report** program.

Transactions may be applied at any time to the Master File through the **Enter Inventory Transactions** program. An **Update Report** automatically prints during this entry process to provide an audit trail of all inventory acitivity.

Several reports are available for the maintaining of stock, analysis, and forecasting. These reports include the **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.

At the end of an accounting period (usually a month), and then again at the end of a year, the **End of Period Processing** program is run to update current balances and clear previous balances.

COMPUTRONICS

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

✓9

***** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE *****

H & E COMPUTRONICS INC.

• • • EVERYTHING FOR YOUR TRS-80™ • • •

And All Microsoft BASIC Computers

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

Announcing! DOS RANDOM ACCESS & BASIC FILE HANDLING

Written For TRS—80 and All Computers Using MICROSOFT basic

FINALLY IT IS HERE. At last someone wrote a book on disk random access and file handling. A book written for the non-programmer. Written for the businesman and professional who need to solve and write special programs for in house business problems.

Written for the hobbyist who wants to go beyond the cassette recorder and into disk storage and file manipulation.

This book handles a subject of reasonable complexity, so simple and down to earth, that anyone with some Level II experience can cope with the material.

This book is written using a simple program as a starting point. The programs grow in ability and complexity as the book progresses into the various aspects of file handling and record manipulation. Extensive effort has been made to keep the material coherent and every program line is explained in detail.

The programming material presented in this 150 page self-instruction tutorial will provide any non-programmer with the ability to write special programs for inventories, mailing list, work scheduling, record keeping, research project data manipulation, etc. The subjects covered in this edition are as follows . . . . .

(A) The writing of a Menu to summarize program functions.
(B) The writing of a screen format to accept record data.
(C) The creation of the basic record.
(D) The Fielding and LSET routines for buffer preparation.
(E) The writing of the record to disk in a Random Access mode.
(F) The retrieval of a record from disk in a Random Access mode.
(G) The ability to change or edit a record.
(H) The LPRINT capability from disk using three different formats.
(I) Deleting a record from a Random file.
(J) Sorting the Random file.
(K) Searching the Random file by name or other keyfield.
(L) The ability to search in a "NEXT or PRIOR" fashion.
(M) The ability to purge a disk file from deleted records.
(N) The ability to calculate with data from a disk file.
(O) The provision for future expansion of the data fields.
(P) The use of flags to prevent program crashes.
(Q) Date setting, printer on-line, and many other routines that make a program run like a commercial written program.

D.S.C., Publishing, Div. of, D.S.C., INCORPORATED.

DEALERS—CALL D.S.C. PUBLISHING DIRECTLY FOR DEALER PRICING—(203) 748-3231 or write to D.S.C. PUBLISHING, P.O. Box 769, Danbury, CT 06810-0769

COMPUTRONICS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

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

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

DOS RANDOM ACCESS & BASIC FILE HANDLING	$29.95

24 HOUR ORDER LINE
(914) 425-1535

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

See List of Advertisers on page 290

> *"The computers cannot be credited for Joel's intelligence, but they have created an outlet and a means to nurture it."*

continued from page 128

ing did not exist. Ruth's typing method is taught through a software program distributed by Instant Software. For a majority of the physically handicapped the keyboard is the only barrier between them and the computer. Ruth's method won her an honorable mention in the Johns Hopkins Search.

"The average person in society is not going to rearrange anything for you (the handicapped person). They're going to expect you to conform like everyone else," says Lambert. If the keyboard itself has to be rearranged to become accessible to the handicapped, not much has been gained. The idea is not to rearrange technology's tools; that is cost prohibitive. A businessman will be less likely to hire a handicapped receptionist if they need to spend thousands of dollars for a special typewriter, as well. But if the standard typewriter is all the equipment needed, then why not?

Ruth's method rearranges the person's orientation to the keyboard, not the hardware itself. In One Hand Touch Typing the home row is different. Rather than the eight fingered a-s-d-f-g and ;-l-k-j-h, it is four fingered f-g-h-j. Her method also incorporates those limited to headgear, mouthgear and other supportive equipment.

The software takes the handicapped student letter by letter through the keyboard, just as in a conventional typing course. Each lesson contains keyboard character identification, finger position display (home row), introduction of new characters, new character reach coordination, drill on new characters and integration of new with old characters. After using the program to master the keyboard, the handicapped person is able to use the computer for everything from programming to journalism. The possibilities are endless. "Nine times out of ten," says Lambert, "the handicapped person will measure up over and above any 'normal person' because of their desire to succeed."

Joel Stevey, User

One person with that desire is Joel Stevey. Joel is a 24 year old quadraplegic. He converses, writes and does all his work via computer. Joel's wheelchair has more than the usual equipment. Attached to it is the Autocom communication system by TSI (Telesensory Systems, Inc.).

The Autocom is a portable electronic communication aid built into a special wheelchair lap tray. The lap tray is divided into a grid of 128 squares. The squares represent individual vocabulary items. There are 59 levels to the Autocom, for a total of 7552 vocabulary items. Joel uses his mostly as a word processor.

Joel's father: "It is operated by microswitches under the squares. . . He has a magnet strapped to his watch. That magnet activates the switches. He can move that over the board. . . and that's how he talks. There is an LED readout which faces toward him and there is another one, which is on a flexible kind of gooseneck, directed toward the people talking with him."

"In fact, that was his invention. TSI bought the contract from the University of Wisconsin. Wisconsin sent them out here because they said Joel has done more with his board than anyone else. They asked Joel what the system lacked. He told them, 'People have to stand behind me to see what I'm saying.' And that's when they came out with this gooseneck idea."

The Autocom was developed at the University of Wisconsin. At the time, Joel's family did a lot of traveling. His father, a minister, served with the Army for 22 years. It was a case of being in the right place at the right time. Joel's speech therapist had heard about the development of the Autocom and arranged for Joel to get one of the 40 prototypes.

"When Joel first got the original Autocom he had a date. He told me later, 'This thing paid for itself already'," said Joel's father, John Stevey, in a telephone interview. "Joel now can have friends apart from Dad and Mom. He goes to classes at Nyack College. . . he can talk with kids without us being around. The computer provides him with an awful lot of freedom he didn't have before. . . He moves his hand slowly, sometimes he uses his nose, but he does it."

The Autocom was not Joel's first exposure to computers. In Virginia he went to Fairfax and Edison High School. They had a computer studies program using a Hewlett/Packard 2000S and 2000E hooked to an IBM. He belonged to a computer explorer scout post. Being on an Army base, the scouts had access to its sophisticated equipment and were permitted to use the night vision laboratory.

Joel's first computer was a TRS-80 Model I with 32K and two disk drives. "When he got the Radio Shack computer he rewrote some of their programs. He told me, 'Dad, this is wrong. This is not operating right.' And they really did have design problems and he knew enough to correct them."

Joel later sold his Model I and bought a Model III. During our conversation, Joel's father, John, continuously praised their local Radio Shack store for their helpfulness and desire to assist. "I can't say enough good things about the Radio Shack store at the Nanuet Mall."

Joel writes his own programs. He keeps his bank book and his address book in his computer, and he uses it to do all his correspondence. "It's really opened up a whole new world. . . This is what he does most of the time. He enjoys writing programs. . . We're looking around now and asking, 'How do we make it pay?' That's an aspiration he has even though he's limited. He feels the same need for independence that everyone else feels."

Aside from computers, Joel is extremely bright. In high school his IQ was measured at 140. During his last two years in high school he was listed in *Who's Who in American High Schools*. When he was born his parents were told he was a vegetable and they should put him away. The computers cannot be credited for Joel's intelligence, but they have created an outlet and a means to nurture it.

Two weeks out of every summer Joel is a counselor for handicapped kids at the Monadnock Bible Conference in New Hampshire. By working with the kids, his father said, "It says to them, 'Hey, one of our guys is making it.' "

Filling The Need

The research for this article was uncommonly easy to do. It seemed everyone I men-

Photo 2. Although the red LED lettering is easy for a person to read, the camera didn't pick up Joel's message: "This is my freedom machine."

MIKEEANGELO®*1 BRINGS

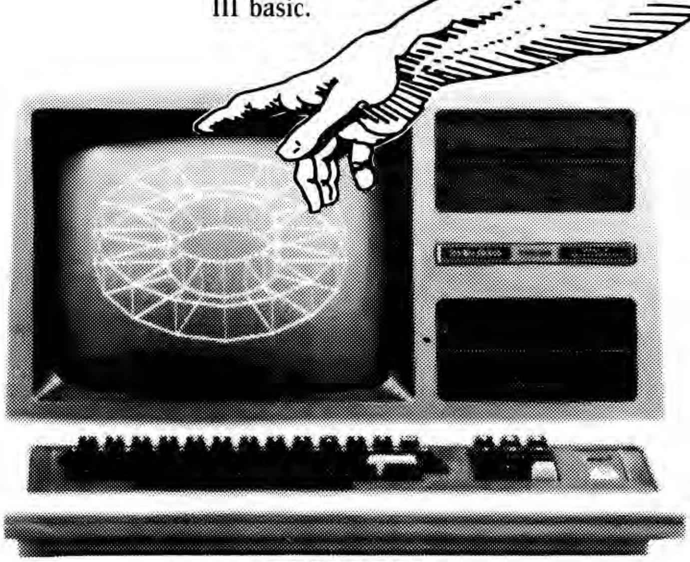
Graphic

LIFE

TO

TRS-80®*2

Model I and Model III
Must have Level II
Basic or Model
III basic.



WE'RE GOING TO CHANGE THE WAY YOU SEE THINGS!

Designing a computer that people could use in business and leisure, without spending alot of money was once a problem. Thanks to the TRS-80* there are now over 200,000 Microcomputers in homes and business today.

Unfortunately in keeping the price of the TRS-80 down, the Random Access Memory was kept down to only 17K. Yes. 17K. (16K User Memory and 1K Video Memory. 128 x 48 or 6,144 Independently controlled dots.) BUT WE'VE CHANGED ALL THAT.

NOW HIGH RESOLUTION GRAPHICS ARE HERE!

MIKEEANGELO is 16 times better with the Model III and 12 times better for the Model I.

That's 512 x 192 or 98,304 dots for Model III, and 384 x 192 or 73,728 dots for Model I.

MIKEEANGELO connects to your TRS-80 easily.*3 You can do it yourself with our instruction book in minutes and a handy tool kit is available, if you need one for $15. Please specify Model I or III TRS-80.

FEATURES YOU'VE NEVER SEEN BEFORE.

- Hardware total screen inverse video
- Two modes of density:
  384 x 192 or 192 x 192 for Model I
  512 x 192 or 256 x 192 for Model III
- Every dot can be set, reset or tested.
- 12288 bytes Video Memory
- Select normal, high or very high resolution with or without inverse video via output port.
- Disable Mikeeangelo's memory via output port.

*1 MIKEEANGLEO is a trademark for Mikee Electronics Corp.
*2 TRS-80 is a trademark for the Tandy Corp.
*3 Opening your TRS-80 will void the limited warranty.
*4 90 day warranty, full refund or replacement.

Order Now For Only **$340.** Complete

Order today complete with power supply, case, software and excellent manual for $340. Check, Money Order, Master Charge, Visa, Mastercard or COD's accepted. *4

Or write for free booklet and more information.

MIKEEANGELO®

MIKEE ELECTRONICS CORPORATION
P.O. Box 3813 Bellevue, Washington 98009
Telephone 206/451-0574

Stock to 30 days for delivery.
Washington residents must add sales tax.

"Dealers Inquiries Invited"

537

Introductory Special for 90 days only!

See List of Advertisers on page 290

"Simplicity is often overlooked in favor of the overly sophisticated."

tioned the project to knew of someone involved in the use or creation of computerized handicapped aids. I learned of Joel Stevey through a proofreader at *80 Microcomputing* who is involved with the Monadnock Bible Conference. Ruth Lambert works a few miles down the road at Instant Software. Wherever I turned, whoever I spoke to seemed to be able to supply me with information. I was soon overwhelmed with names to contact, phone numbers, brochures, pamphlets and other literature.

I found out about a braille display reader with a cassette input that forms words using braille cells on a display surface; digitalized speech programs; programmed teacher's aids to eliminate the more tedious duties from a teacher's time so they are free to spend more time with students; data inputing via bio-feedback devices; and hardware that allows the deaf to use the telephone.

Cost and practicality is a problem with much of what is being developed. Although the needs of the handicapped are special, to succeed they must help the handicapped user function as a normal user or person. It would be unrealistic to expect a computer to make a handicapped person normal, but it is not unreasonable to expect the computer to fill most gaps. For the computer application to be successful, it needs to be compatible with the non-handicapped society.

The specially designed car works because it functions the same as a standardly equipped vehicle. It can keep up with other cars on the road. The control mechanisms are specialized, but the car functions on a par with all others.

Much of the computer technology developed for the handicapped is only applicable in special environments. Simplicity is often overlooked in favor of the overly sophisticated. Work produced by government grants and large institutions is usually painstakingly slow. Software and hardware developers for the handicapped should take a hint from Ruth Lambert's idea: aim to make it easier for the handicapped to adjust to society.

Consider business, entertainment and other software. Though many companies put out quality products, most users also write their own. Part of the fun of owning a microcomputer is writing original software or improving existing programs. Throughout the research of this article I had a reoccurring thought: Why don't computer hobbyists write programs and design modifications for the handicapped? The only answer I could come up with was they were not aware of the need.■

80 Microcomputing *would be pleased to publish any well-written programs developed for use by the handicapped in future issues. The best programs will be featured as a "Program of The Month."*

"It takes squeezing to get out what is trapped inside."

Thoughts on Being Handicapped

by Joel Stevey

Communication is the most difficult problem in my life. Since cerebral palsy has robbed me of most of that function, I feel like a full bottle with its cork stuck. Most my life is spent listening and thinking, but I really long to contribute. Can you imagine how it is to love humor but not have the ability to tell a joke? I know many, but I cannot tell them. Oh well, that is just the beginning.

Not having speech is a particular problem. I am often included in groups physically, but how to get into the conversation is another matter. My contributions are slow and require assistance. Sometimes my message gets twisted in translation. That bugs me. I am sure it bothers others too. As a result, I am often ignored or another breaks in and the conversation moves on. There are some who exercise patience and understanding and refuse to be distracted. I appreciate them.

If I could do any one thing well, I would choose writing. I am perturbed when I have a good thought or idea and cannot record it. I am a night person and do much thinking in bed. Some of it I would like to write down. Closely related to this is the ability to use reference materials. I am hoping for future help in this area.

Typing is a very important part of my communication system. All my homework is done that way, and my finger gets tired. Have you ever tried a long division problem with a typewriter? Time is the major problem. Some days I type for three hours. Remember that I, too, make mistakes. Some day I will get a self-correcting typewriter, but until then, my folks help.

My frustration is illustrated in the first paragraph when I pictured myself as a bottle. Talking, working, playing, dating and many other areas are affected. I am also like a sponge. It takes squeezing to get out what is trapped inside. ■

Written by Joel as an English class assignment in 1976.

✓*See List of Advertisers on page 290*

The Key Box

Basic Level II
Model I

Can your computer help you learn Spanish or French?

RS-80Tay, Aysay Hatway?

```
0 '*** La Computadora Internacional
        for the model I, by jerold stratton
10 CLS:CLEAR7000:DIMW$(500),C$(1,9),C(500):S$=STRING$(120,32):P1=256:P2=516:RR$=
CHR$(191):GOTO300
99 '*** INPUT AND CHECK THE FILE
100 CLS:PRINT@89,"ENTER STUDY WORDS"
110 PRINT@P1,"FOREIGN TRANSLATION";:INPUTW$(I):PRINT@P2,"ENGLISH TRANSLATION";:I
NPUTW$:W$(I)=W$(I)+RR$+W$+"/":W$=""
120 IF"STOP"+RR$+"STOP/"=W$(I)THEN130ELSEPRINT@P1,S$:PRINT@P2,S$:I=I+1:GOTO110
130 CLS:INPUT"CHECK STUDY LIST (Y/N)";F$:IFF$="N"THEN200
140 FORJ=0TOI:A=J:GOSUB2000:PRINT@P1,"FOREIGN WORD:  ";:FORJ1=0TOC1:PRINTC$(0,J1
);"...";:NEXT:PRINTS$:PRINT@P2,"ENGLISH TRANSLATION:  ";:FORJ1=0TOC2:PRINTC$(1,J
1)"...";:NEXT:PRINTS$
150 R$=INKEY$:IFR$=""THEN150ELSEIFR$="E"THEN160ELSEIFR$="Q"THEN130ELSEIFR$="S"TH
EN170ELSENEXT:GOTO130
160 INPUT"FOREIGN CORRECTION";W$(J):INPUT"ENGLISH CORRECTION";W$:W$(J)=W$(J)+RR$
+W$+"/":PRINT@P1,S$;S$;S$:NEXT
170 PRINT@P2,S$:PRINT@P1,"";:INPUT"FOREIGN WORD TO BE SEARCHED FOR";A$:FORK=0TOI
:A=K:GOSUB2000:FORJ1=0TOC1:IFA$=C$(0,J1),J=K-1:NEXTJELSENEXTK,J
199 '*** STUDY THE FILE
200 Q=0:CLS:A=RND(I)-1:B=RND(2):IFBB$="FOREIGN",B=1ELSEIFBB$="ENGLISH",B=2
210 IF(VANDC(A)),GOTO200ELSEGOSUB2000:ONBGOTO230,250
230 PRINT@P1,C$(0,RND(C1)-1);S$:GOSUB280:FORJ1=0TOC2:IFA$=C$(1,J1)THEN290ELSENEX
T:PRINT"SORRY, CHARLIE.  THE ANSWER IS":PRINT:FORJ1=0TOC2:PRINTC$(1,J1)"...";:W=
W+1:NEXT:FORJ=1TO1500:NEXT:C(A)=0:GOTO200
250 PRINT@P1,C$(1,RND(C2)-1);S$:GOSUB280:FORJ1=0TOC1:IFA$=C$(0,J1)THEN290ELSENEX
T:PRINT"I'M AFRAID YOU GOOFED!  THE ANSWER IS":PRINT:FORJ1=0TOC1:PRINTC$(0,J1)".
..";:NEXT:W=W+1:FORJ=1TO1500:NEXT:C(A)=0:GOTO200
280 PRINT@P2,"";:INPUTA$:IFA$="STOP"THEN300ELSERETURN
290 PRINT@212,"CORRECT!!!":R=R+1:PRINT"PERCENTAGE="R/(W+R)*100:FORJ=1TO1200:NEXT
:IFNOTV,C(A)=1
291 GOTO200
300 CLS:PRINT"              1)  SAVE WORDS
            2)  INPUT FROM TAPE
            3)  ADD WORDS TO MEMORY
            4)  STUDY WORDS
            5)  CHANGE FORMAT
            6)  USE UNUSED QUESTIONS OR ALL
";:INPUTF:ONFGOTO400,500,100,200,350,320:GOTO300
320 V=NOTV:GOTO300
350 INPUT"FOREIGN, ENGLISH, OR NO PREFERENCE";BB$:GOTO300
400 INPUT"FIRST FOREIGN WORD TO BE SAVED";N$(0):INPUT"SECOND FOREIGN WORD TO BE
SAVED";N$(1):FORJ=0TOI:A=J:GOSUB2000:FORJ1=0TOC1:IFC$(0,J1)=N$(N),N(N)=J:N=N+1:I
FN>1THEN450
410 NEXTJ1,J:PRINTN$(N)" COULD NOT BE FOUND":GOTO300
450 INPUT"TAPE READY (Y/N)";F$:IFF$="N",300
460 PRINT#-1,N(1)-N(0):FORJ=N(0)TON(1):PRINT#-1,W$(J):NEXT:GOTO300
500 INPUT"TAPE READY (Y/N)";F$:IFF$="N"THEN300
510 INPUT#-1,J:FORI=ITOI+J:INPUT#-1,W$(I):NEXT:I=I-1:GOTO300
1999 '*** SUBROUTINE TO SEPERATE WORDS IN FILE STRING
2000 FORJ1=0TO9:C$(0,J1)="":C$(1,J1)="":NEXT:C1=0:C2=0:C=0:QQ=1:FORQ=1TOLEN(W$(A
)):JJ$=MID$(W$(A),Q,1):IFJJ$=RR$GOSUB2030:QQ=Q+1:NEXT:ELSEIFJJ$<>"/",NEXTELSEONC
+1GOSUB2010,2020:QQ=Q+1:NEXT
2001 RETURN
2010 C$(C,C1)=MID$(W$(A),QQ,Q-QQ):C1=C1+1:RETURN
2020 C$(C,C2)=MID$(W$(A),QQ,Q-QQ):C2=C2+1:RETURN
2030 GOSUB2010:C=C+1:RETURN
```

Program Listing

Jerold M. Stratton
148 S. Division
Hesperia, MI 49421

I designed this program to help me study Spanish and French. You can easily modify it for other languages or study needs.

The program first displays a menu. The first time you use the program, you must enter 3 (add words to memory). The computer asks for the foreign word and then the English translation. It will continue to ask for these until you enter Stop for both questions. I have provided a sample file shown in Table 1 to help you check out this program. Enter the words exactly as they are shown. A slash between words separates different definitions of the same word.

The computer asks if you want to check the file for errors. Enter Y and the first set of words ("Buenos dias..." and "Good morning...Good day...") appears. To change any set of words, type E and enter the correct words. Hit the space bar to advance to the next set of words. You can access the search function at any time while a set is displayed by hitting S. The computer will ask for

Professional/Scientific

WORD PROCESSING

Since introducing Qwerty 3.0 in September, people have been calling to ask if we were making ludicrous claims. The answer is NO! QWERTY 3.0 does all we claim it does and more! No other software of this type can match QWERTY 3.0

$$T_1^2 = \sum_{i=1}^{n} \frac{\alpha_i \sigma_i + \mu^2 \pi^2}{\lambda_i \psi_i - \rho^2 \theta^1}$$

It is the best. Period. We guarantee you will agree! If for some reason you find that this program does not meet your needs, return the entire package within 14 days for a prompt and cheerful refund.

(Actual QWERTY text above).

SOME FEATURES OF QWERTY 3.0

1. Automatically prints in proportional print, with a suitable format. Transitions between the three print styles are easy, including all expanded print modes. **2.** QWERTY 3.0 adds 75 new symbols, including upper and lower case Greek letters, mathematical symbols such as integrals and summations, arrows, brackets, and probability symbols. **3.** Any character can be used as a subscript or superscript, even simultaneously. Carats, bars, and tildes can be placed over any character, with precise position control. **4.** Underlining, with or without underlining of spaces, including long ratios and mathematical expressions. **5.** FOOTNOTES can be placed on any page so that they remain on the desired page, even if text is inserted later. **6.** TABLE commands enable positioning of the print head anywhere on a line. Invaluable in printing neat mathematical layouts, tables, columnar material, etc., in proportional print. **7.** PRETTY commands allow printing of repetitions of a chosen character. When combined with TABLE, decorative borders can be produced with ease. **8.** FOLIO format produces output in two or three columns per page, in either proportional or 16.7 cpi mode. Ideal for newsletters. **9.** Supplies a third output mode, in which only SCRIPSIT commands are obeyed. Allows printing of special QWERTY commands for future reference. **10.** PAGE END indicates where pages will end, and the page number, without printing the text. One can prepare an almost error-free document without ever using paper. **11.** Correction of SCRIPTSIT's errors and inconveniences, extensive documentation, and much more!

For cautious buyers, we offer the manual (over 70 pages) for $10. When you decide to buy QWERTY 3.0, we will credit the full manual price.

Qwerty 3.0 Disk, Manual, Reference Card, and Printer Table Rule **$49.95**

Qwerty Manual Alone **$10.00**

COMING SOON FOR DAISY WHEEL II & LINEPRINTER VIII

MED SYSTEMS SOFTWARE

P.O. BOX 2674 CHAPEL HILL, NC 27514

TO ORDER, CALL: 1-800-334-5470

128

See List of Advertisers on page 290

the foreign word and advance the display to the set you requested. You can change it, if necessary, by pressing E. To leave the "Check Study List" mode hit shift Q, and answer N to the next question.

You are now in the study mode. The computer will display a word or definition. If it is in a foreign language (for example, "el pueblo"), enter the corresponding English answer ("town" or "village"). If you answer incorrectly, the TRS-80 will display the correct answer. The computer displays your percentage of correct answers. To leave the study mode, enter Stop to any question, and the program returns to the menu.

Menu option 5, Change Format, allows you to have the computer give only the foreign translation or only the English word in the study mode. If you choose menu option 6, Use Unused Questions Or All, you will be asked questions which were not asked before, or which you answered incorrectly. Enter menu option 6 again to return to normal execution.

If you want to save the words on tape, choose menu option 1, Save Words. The computer will ask for the first and last words to be saved. To save everything, enter "Buenos dias" for the first and "Detras de" for the last word. If you enter "tener que" and "el pueblo", only "tener que", "pensar", and "el pueblo" are saved.

Buenos dias! Aprendan ustedes bien! ■

Foreign translation	English translation
BUENOS DIAS	GOOD MORNING/GOOD DAY
DE LA MANANA	A.M./IN THE MORNING
HASTA LA VISTA/HASTA LUEGO	UNTIL LATER/SO LONG/I'LL SEE YOU LATER
TENER QUE	TO HAVE TO
PENSAR	TO THINK/TO PLAN
EL PUEBLO	TOWN/VILLAGE
DETRAS DE	BEHIND
STOP	STOP

Table 1. Sample phrases

Variable	Function
A	the number (X) of the current W$(X) being studied
B	determines if the foreign or English will be asked
BB$	stores if you want only foreign or only English asked
C(X)	records whether question W$(X) was answered correctly
C$(X,Y)	stores the individual words separated from W$(X)
C1	the number of foreign words in W$(X)
C2	the number of English words in W$(X)
C	stores whether or not foreign or English words are being separated (subroutine at line 2000)
F$	for all input statements requiring a 'Y' or a 'N' answer
F	the answer to the menu
I	stores how many strings in the file
J,J1	used in For. . . Next statements
K	the For. . . Next variable in the search routine
N$(1),N(1)	first string to be saved (W$(n(1)) is first)
N$(2),N(2)	last string to be saved (W$(N(2)) = N$(2))
P1	position for PRINT@ statement
P2	another position for PRINT@ statements
Q	current position in W$(A), in subroutine at line 2000
QQ	position of the first letter in the current word being separated from W$(A)
R$	used for the INKEY$ statement
RR$	CHR$(191)—separates foreign section of W$(X) from English section
R	the number of questions answered correctly
S$	a string of spaces
V	stores whether only questions that were not yet asked, or were answered incorrectly will be asked (see text)
W$(X)	stores a set of corresponding foreign and English words
W	the number of questions answered incorrectly

Table 2. Variable functions

See List of Advertisers on page 290

AMERICAN

*New Retail Location
Southwood Shopping Plaza
Pryor, OK.

118 SO. MILL ST • PRYOR, OK 74361
PHONE (918) 825-4844

ALL LISTED PRICES ARE CASH DISCOUNTED. WE ACCEPT MC, VISA, AND COD (REQUIRES CERTIFIED CHECK, CASHIER'S CHECK OR CASH). FOR CHARGE ORDERS, PLEASE ADD 3%. FOR COD SERVICE, PLEASE ADD 3%. OKLAHOMA RESIDENTS PLEASE ADD 4% SALES TAX.

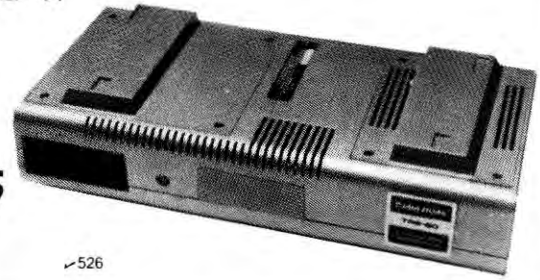
32K EXPANSION INTERFACES

SPECIAL OFFERING ★ **NEW LOW, LOW PRICE** ★

Due to a very special purchase, American Business Computers is able to offer a limited number of Radio Shack* Expansion Interfaces at the lowest price ever.

American Business Computers guarantees Expansion Interfaces to be Brand New — with original documentation and in perfect working condition.

$329.95



526

HARD DISK DRIVES

5 MEGABYTES FIXED
5 MEGABYTES REMOVABLE
STORAGE FOR TRS-80 MOD II

Call or write to American Small Business Computers and let us show you how we can give you almost unlimited disk storage on your TRS-80 MOD II. American Business Computers features the Cameo Disk Subsystems with associated software. **Dealer Inquiries Invited.**

527

SUPPLIES

PAPER FOR YOUR PRINTER

9 1/2 x 11 — Tear-off perforations and you have an 8 1/2 x 11 "cut sheet". 3200 sheets of 15 pound paper.....$39.95

VERBATIM DATALIFE DISKETTES

We feel this is the *BEST* Diskette on the market!

TRS-80 MOD 1 — *29.95*
TRS-80 MOD II — *49.95* *BOX OF 10*

528

OKIDATA MICROLINE 80 PRINTER

$399.00

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.

560

OLYMPIC DECATHALON*
FOR THE TRS-80*
MODEL ONE

Here's your chance to become an Olympic champion in the greatest of all athletic competitions—the Decathalon! This exciting skill game takes you through all ten events of the real competition from the pole vault to the discus to the 100-meter dash, testing your prowess in three diverse track and field disciplines—running, jumping and throwing. Each event is presented with incredible animated graphics that put you directly into the action. To meet the challenge of a champion is difficult; to win the gold medal, you'll need a good sense of timing, fast reflexes and good coordination. For 1-8 players.

562

ADVENTURE*
FOR THE TRS-80*
MODEL ONE

You are standing at the end of a raod before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully. There is a shiny brass lamp nearby.
Get ready to make the first decision that will affect your course through the "Colossal Cave" in the first complete version of Adventure ever implemented on personal computers! As you prove yourself sufficiently skilled in overcoming obstacles, more sections of the cave will become open to you. Proceed, explore the cavern, and collect your treasures. But watch out! Killer dwarves, green snakes, active volcanos, sulfur lakes, trolls, a five-foot oyster and other hazards abound. Each Microsoft Adventure session is unique, with over 400 messages and descriptions that can be combined in endless ways. Never before has this sophisticated fantasy/logic game, written for the DEC PDP-10, been **totally** implemented on personal computers.

*TM Microsoft Corp.
*TM Tandy Corp.

563

TRS-80™ Daisy Wheel Printer II

By Radio Shack

Yes, this is the **same** Daisy Wheel sold by Radio Shack for a couple of hundred bucks more. This is just one more reason why it **really** pays to shop American Small Business Computers.

$1649.95

918-825-4844

561

RETAIL STORE—SOUTHWOOD SHOPPING CTR—PRYOR, OK

Radio Shack may have trouble with this one.

Model II Compiler Basic

Model II Compiler Basic
Tandy/Radio Shack
Fort Worth, TX
$199

Larry Clark
Perimeter Data Systems
4449 Oak Trail Drive
Atlanta, GA 30338

As a professional programmer engaged in building large applications systems, I eagerly awaited Radio Shack's Compiler Basic for the Model II. I anticipated a significant performance advantage over the interpretive Basic that comes with the system. In addition, I felt that the ability to distribute programs in object form would provide increased protection against software piracy.

I knew that my existing programs would require modification. Radio Shack clearly indicates that the language differs from interpretive Basic, and suggests that it is primarily suitable for new application development. To achieve higher performance, I expected that the compiler language would require some additional specification. But I was utterly unprepared for what I encountered: gratuitous changes for the sake of change, callous disregard for human factors, and woefully disappointing performance.

The Development System

Programs are normally written and debugged using the Development System. This is a complete environment that closely resembles interpretive Basic—programs can be written, saved, executed, modified, etc., without returning to TRSDOS. There is also a stand-alone editor called BEdit, which allows more powerful editing operations, and a stand-alone run-time package that can execute previously compiled programs.

The editing commands in the Development System are annoyingly different from the familiar Edit command of interpretive Basic. In fact, there is no Edit command; its function is performed by Change, which is better suited to global changes than to intra-line corrections. The BEdit program uses another scheme (including an Edit command), but differs from both of the above mentioned, and from Edit-80, which it closely resembles. This makes it incredibly difficult to switch from the Development editor to BEdit. The difficulty is compounded if the user also works in interpretive Basic and/or Edit-80 (e.g., for Assembly-code or Fortran work).

If you have existing programs in interpretive Basic, forget trying to use them as a starting point. The file format used for storing source programs by Compiler Basic is different from either of the formats used by interpretive Basic. I managed to write a conversion program, but the language differences are so significant that I'd have been better off starting over.

Data Types

The compiler supports only two types of numeric data, integer and real. Integers occupy two bytes and perform as in interpretive Basic. Reals, on the other hand, are stored in floating packed decimal, and occupy eight bytes each. Each real number carries up to 14 significant digits. The use of a decimal base would seem a good choice for accounting work, since it eliminates the possibility of roundoff errors.

Strings have both maximum and current lengths. The default maximum is 255 (as with interpretive Basic). However, if the user knows that the string will never be that long, he can assign a shorter maximum to save space. Unlike interpretive Basic, strings always require enough storage to hold their maximum length. This avoids the interpreter's string compaction delays, but may prevent programs with large string arrays from fitting into memory.

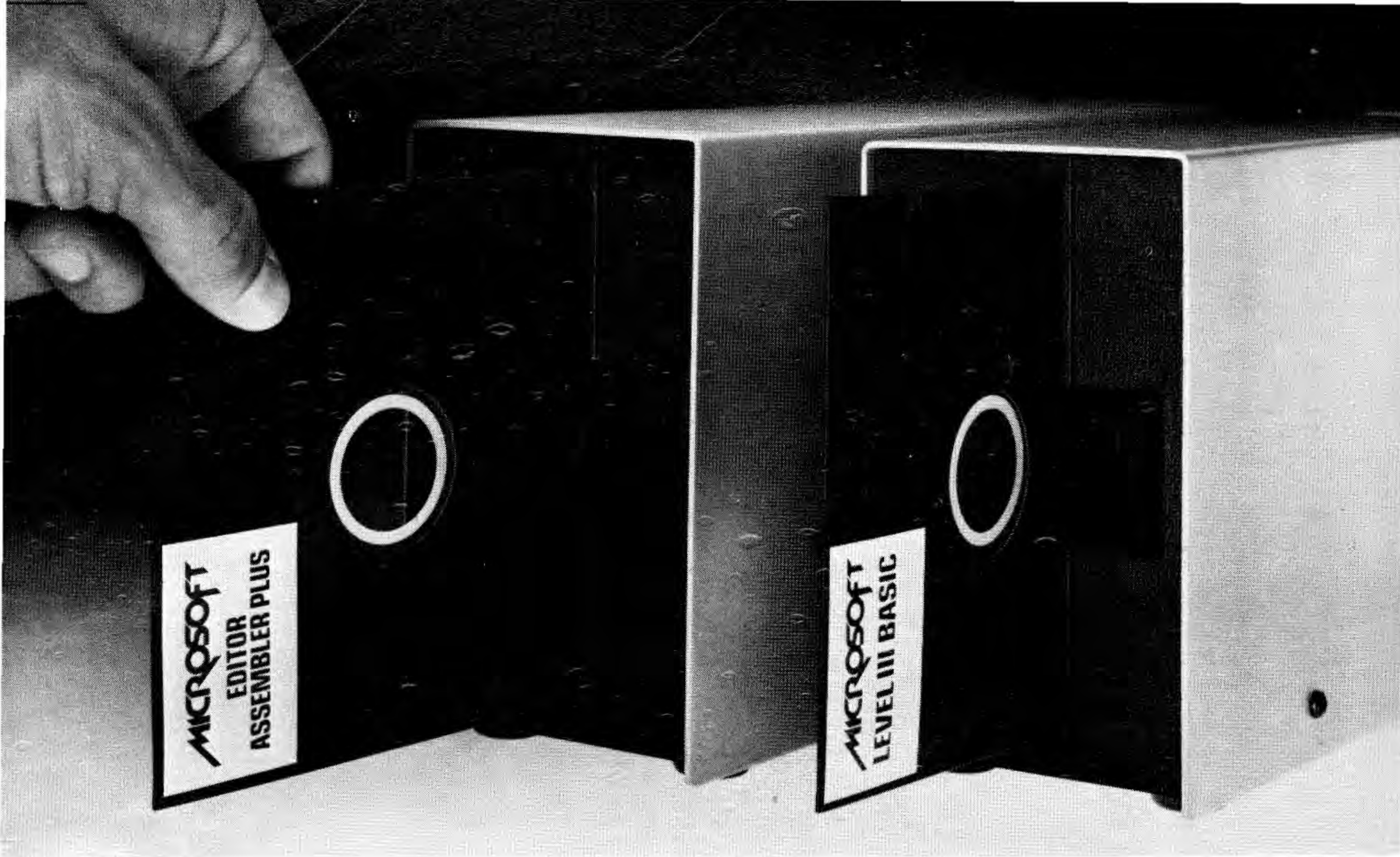
Variable names may be arbitrarily long, and the first six characters are significant. Thus, Joe and Joseph are different variables, although Joseph and Josephine are the same. A variable's type may be set implicitly (Real is the default), or specified by appending a type tag ($, %, or #). The first mention of a variable is binding—if the first line of a program refers to A$, the compiler treats every reference to A as a string, regardless of the presence of the $ tag, and the variables A% and A# are not available.

Arrays of all data types are available, but may not have more than two dimensions. Although relatively few programs need more than this, those that do become extremely awkward. The limitation seems arbitrary.

Needless Differences

Some of the other differences seem pointless. Their primary effect is to frustrate users who are accustomed to interpretive Basic.

- Quotation marks are not used around filespecs on commands such as Save and Kill.
- Load will load only an object file, not a source file. You must use Old for source files.
- LLIST <range> is replaced by List <range> {PRT}. The braces are mandatory.
- In concatenating strings, the plus sign has been replaced by an ampersand.
- In debugging a program, you cannot use Print <var> to see its value; you must Display <var> or DI <var>).
- RENUM is not recognized, but Renumber and RE are.
- Print@(6,3),<list> must be written as Print CRT(6,3);<list>.
- To determine the cursor position,



More powerful programming tools for the TRS-80.™ Now on disk.

Better results. Microsoft's Level III BASIC and Editor/Assembler-Plus are programming tools that help you write complex programs in less time, with less effort and utilizing less memory. Better programs. No matter what your programming skill. And for the first time these tools are available on disk.

Editor/Assembler-Plus. A powerful editing, assembly and debugging tool with many sophisticated features that make writing TRS-80 assembly language programs easier, faster and more efficient.

- Full disk capabilities.
- Assembly directly into memory. No need to save object code then reload for execution.
- Macro capability that allows you to define macros for commonly used sequences of instructions.
- Conditional assembly that allows you to generate more than one version of a program.
- Eight breakpoints at a time for program debugging.
- The INCLUDE statement that allows you to call additional disk files for assembly.
- Other features include extensive operators, automatic origin, symbol table printout, quash command, hex, decimal and octal constants, single step-through instructions in memory, five type-out modes, four type-in radices, plus extensive edit commands.

A more powerful BASIC. With Level III BASIC, you get power to perform tasks in BASIC that used to require assembly language. Plus, new ease-of-use features for your TRS-80.

- Advanced graphics. Develop charts, graphs, even animation in Level III BASIC. Draw a line, an outline box or a solid box by specifying just two points. Then save and recall it with BASIC commands.
- MENU. One command that allows you to construct an entire menu.
- CHAIN and COMMON commands allow you to call another program and pass variables to it.
- Powerful editing commands such as COPY/TRANSFER, FIND and CHANGE.
- DUMP command that makes debugging easier.
- Time-limit response. New INPUT # LEN and LINE INPUT # LEN commands allow you to set a time limit on response.
- RS-232 output from BASIC. With a single command.
- More. Level III gives you automatic line numbering, 26 user-definable single stroke instructions, and more.

Disk or cassette. Disk versions of Level III and Editor/Assembler-Plus are brand new. Cassette versions are also available with many of the same capabilities.

Talk to your Microsoft™ dealer. Ask for a demonstration of two of the most powerful tools you can get for your TRS-80: Level III BASIC and Editor/Assembler-Plus. On disk or cassette. From Microsoft.

TRS-80 is a trademark of Radio Shack, a division of Tandy Corporation.
Microsoft is a trademark of Microsoft, Inc.
Microsoft Consumer Products is a division of Microsoft, Inc.

579

MICROSOFT
CONSUMER PRODUCTS

10700 Northup Way, Bellevue, WA 98004. (206) 828-8080

"A close reading. . . reveals that Compiler Basic is not a true compiler."

ROW(x) and POS(x) have been replaced by CRTX and CRTY. (This would be fine if they hadn't reversed the normal usage of the coordinates.)

- The End statement marks the end of complication, not the end of execution.
- The INSTR function has been relabeled POS, and accepts only the two-argument form.
- The MID$ function has been renamed SEG$.

The above list is far from complete, but indicates the nature of the changes. If the compiler were to be used in a vacuum, the choice of syntax wouldn't matter. But every Model II is already supplied with interpretive Basic, and most potential users of Compiler Basic are probably already using it. (If they wanted to start from scratch, they would be better off using a language other than Basic.) So these changes simply make life needlessly difficult.

Input/Output

The file I/O commands are also markedly changed, but in this case there is some justification. Compiler Basic provides a rich variety of I/O options, including fixed or variable-length records; sequential, direct, or indexed access; and stream, formatted, or binary formats. However, most of the interesting combinations seem to waste an inordinate amount of space.

Direct access files automatically include two bytes of overhead per record—one for the record length and a second that's unused. (The record length indicates how much of the record is actually used, even though the physical records must be fixed in length.)

ISAM files also have two bytes of overhead per record, but round the storage to the next larger multiple of 32 bytes. Thus, a file with a record length between one and 30 bytes requires 32 bytes per record.

Numbers in formatted files are written in ASCII. Thus, an integer may take six bytes, including sign.

Real numbers within binary files carry a length byte in addition to the internal representation. This means that nine bytes may be (and therefore *are*) required to store a number.

The ISAM format was apparently dictated by considerations of (limited) compatibility with the Cobol compiler. Considering how much these two languages have in common, I wonder whether (or why) anybody would even want to try to marry them.

Program Segmentation

Perhaps the most welcome feature of Compiler Basic is that it allows true subprograms, which can be called by name (not by GOSUB <line>) and passed arguments. Except where declared as Common or as formal parameters, the variables in the subprogram have nothing to do with those in the calling program. "At last!", I thought, "I can actually write programs with a reasonable degree of modularity and structure." Wrong!

A close reading of the manual reveals that Compiler Basic is not a true compiler. It does not convert source programs into machine code, but rather, into an intermediate pseudo-code, which still must be interpreted at run time. (This is also true of most implementations of Pascal.) The pseudo-code is more compact than true object code would be.

Subprograms can be compiled either separately or in conjunction with the main program. If they are compiled separately, they can be loaded together before beginning execution, but the result can only be executed, not saved. The pseudo-code cannot be linked with true object modules. It is still possible to invoke machine code, but the technique is similar to the USR calls of interpretive Basic—the code must be preloaded into high memory that has been reserved, and the entry point addresses must be coded into the program.

The inability to combine independently compiled modules into a single executable whole makes the concept of separate compilation virtually useless. In designing large systems for beginners, I want to give them simple directions, such as "Enter the command Do Daily and follow the prompts." I certainly don't want to tell them to enter RSBasic, load a half-dozen modules, then say Run. Worse yet, if they have only the run-time system (sold separately for just such applications), they can't load multiple modules even if I wanted them to. So without a linkage mechanism, all necessary subprograms must be compiled at once.

This is easier said than done. The Development System requires room in memory for the entire source program it is compiling, plus the compiler itself. (I suspect it may also keep the object code in memory, since I was able to compile an oversize program by reducing some dimensions.) Consequently, the largest program that can be compiled in one shot is limited, and I reached the limit on my first attempt.

Chaining

So we can't combine modules that were compiled separately, and we can't compile a very big collection of modules at once. What's left? Chaining.

The program chaining facility is a mild improvement over what's available in interpretive Basic. It allows you to save the contents of certain variables in Common, where they can be retrieved by programs later in the chain. Except for this feature, it is no different from the Run program statement in interpretive Basic.

I thought there might be some smaller applications that could benefit from the performance improvements that come from compilation. To see how much improvement the compiler produced, I wrote a simple benchmark to find out.

Interpretive Basic	Compiler Basic
10 DEFINT A-Z	10 INTEGER
20 PRINT TIME$	20 PRINT TIME$
30 FOR I=1 TO 1000	30 FOR I=1 TO 1000
40 X=(I*3+6)/7	40 X5(I*3+6)/7
50 NEXT I	50 NEXT I
60 PRINT TIME$	60 PRINT TIME$
70 END	70 END

This seemed like a fair test, since both systems would be doing arithmetic using the same internal number format. Upon running these test cases, I was amazed by the relative performance of the two systems: Interpretive Basic, 13 seconds; Compiler Basic, 22 seconds (plus 14 seconds compilation).

Where I expected at least a threefold improvement (probably more like tenfold) in performance, I actually achieved a degradation of about 70 percent. Either Microsoft does things awfully well, or Ryan and McFarland (the authors of Compiler Basic) are doing something terribly wrong!

Conclusion

I have never bought a program with such high expectations, and wound up feeling so totally ripped off. It is inconceivable to me that a compiled program—or even a partially compiled one—can perform so poorly.

I am equally appalled by the unnecessary incompatibilities between this and the standard version of Basic for the Model II. It shows an utter lack of concern for the people who will inevitably use both.

If the compiler is intended for developing applications programs, the authors should recognize that program linkage is a necessity, and that disk space may be a precious resource. In any case, there is little rationale for needless overhead bytes.

After my first experience with the compiler, I put it on the shelf and dismissed it as useless. Several months later I brought it back out for another try, hoping against hope that I had missed something crucial. It appears that I hadn't.

There may be a few limited applications where Compiler Basic might be useful. If you know of one, please contact me. I'm willing to sell my copy at a sizable discount.■

Version 2.0 for Models I and III.

The New NEWDOS 80

Paul R. Prescott
737 Woodlawn Avenue
Dallas, TX 75208

NEWDOS 80 Version 2.0
Apparat, Inc.
Denver, CO
$149.00

Apparat revolutionized TRS-80 disk operation with their initial release of NEWDOS 2.1 several years ago, correcting TRSDOS bugs and adding several modifications. NEWDOS 80 followed in mid-1980, adding additional commands and routines to DOS and Disk Basic. With the release of NEWDOS 80 Version 2 for both the Model I and III, this group has produced an extensively revised system that can meet any challenger head-on! NEWDOS is a new operating system rather than a debugged and enhanced TRSDOS.

Apparat supplies seven other programs on the NEWDOS disk. Superzap allows direct disk access for applying corrections. Chainbld provides a mini-text editor for writing chain files. EDTASM is Radio Shack's Editor/Assembler with disk capabilities. LMOFFSET loads machine language programs from tape to disk and vice-versa, and applies a relocating appendage to keep them from crashing DOS while loading. Dircheck reads a directory to provide essential repair information. Disassem, a machine language disassembler, writes a source file readable by EDTASM to disk. H. S. Gentry's automatic spooler ASPOOL feeds a print file from disk to the printer while the computer processes another program. Version 1 owners note that none of these programs as supplied with Version 1 operate under Version 2. Read the documentation carefully when converting.

Documentation

The most serious shortcoming of NEWDOS 80 Version 1 was its documentation. The new Version 2 manual consists of 258 pages, including a badly needed index. Chapters are sectioned by DOS functions and Basic enhancements, instructions for other programs supplied with NEWDOS, and technical information on DOS routines for assembly programmers. A short chapter provides information on converting from Version 1 to Version 2, and converting from Model I to III. The operating system consists of 14 DOS overlay modules while Disk Basic is made up from one resident module and eight new overlays. Chapter 5 documents all modules; select only those modules necessary for each disk and kill the rest to provide more disk space.

Version 2 and Version 1 incompatibilities make some programs inoperative unless you incorporate changes. The same applies to TRSDOS, mainly with machine language programs that call certain DOS routines. New entry point addresses are documented if you need to make changes. Apparat also supplies zaps for Scripsit, Visicalc, APL80/CMD, and Racet's DSM and Infinite Basic so they function properly. As with other operating systems differing from TRSDOS, not all programs from other sources run correctly.

Technical jargon can be confusing even for knowledgeable operators. For example, Apparat coined the word "lump" to, in the words of the manual, "avoid using the words track and cylinder." Standard DOS operation for the Model I allows five sectors/granule and two grans/track, which gives 10 sectors/track. For the double density Model III this is three sectors/gran and six grans/track, or 18 sectors/track. NEWDOS 80 Version 2 maintains the five sectors/gran, but creates a "lump" which can be set by the user to contain 2-8 granules/lump, or 10-40 sectors/lump. Because the word "track" implies that a track could span several physical tracks on the disk Apparat created "lump." A lump can span several tracks, and a granule can start in one track and end in another. Double density and eight inch disks can use the maximum sector/track count and maintain the same directory structure. If grans/lump is set to eight, the directory can accommodate 7,608 sectors which, again in the words of the manual, "might suffice for a while."

The manual provides an extensive explanation of the disk directory structure including the GAT and HIT sectors, File Directory Entries, File Primary Directory Entries, and File Extended Directory Entries. Apparat changed these structures by defining previously unused bytes. Other operating systems defining these bytes differently are incompatible with NEWDOS. The File Control Block structure is similarly explained.

NEWDOS Zaps

A glossary of terms and a short error chapter appear at the end of the manual, along with an explanation of Apparat's updating procedure. Apparat provides NEWDOS owners with changes (zaps); to receive these return the registration card. Installation instructions are straightforward, but for those uneasy with zapping, Apparat provides an update service. Apparat will perform a full disk copy on your original NEWDOS 80 Version 2 disk from Apparat's master copy if you return your disk. Save any non-NEWDOS programs on another disk before the update or they will be lost.

With Version 1, after all new zaps were applied to a master system disk, the user faced the tremendous task of copying the zapped files to all other disks that contained them. Since many disks contain only part of the original system this took time. In NEWDOS 80 Version 2 Apparat has modified Copy format six to recognize a Copy By File (CBF) parameter. Files are copied one at a time rather than as a full disk backup. This copy format recognizes two new parameters: Destination Files Only (DFO) copies only those files on both the destination and source disks; and Inclusion List File (ILF), specified by filespecs that you create and save on the master disk. The zap duplicating procedure could not be simpler. After you apply the zaps an ILF specifies those zapped files. Then use the following command:

"You can set the number of tries at verifying a read or write to disk before declaring an error."

```
COPY,0,1,,NFMT DFO CBF ILF = filespec:0
```

NFMT prevents the destination disk on drive one from being formatted and CBF indicates file copy. With this command the computer checks the ILF to find out which files to copy; DFO checks the destination disk to see if it already contains that file. If so, the zapped file is copied; if not, it is ignored. No other files are touched. After one disk is updated, enter R from the DOS Ready message to repeat the copy command for another disk. Single drive owners could not use the CBF option with Version 1 but can use it with Version 2. Several disk swaps are necessary but the procedure is greatly simplified. This new zapping ability alone justifies the price of NEWDOS 80 Version 2!

System Customizing

System is a standard library command allowing you to specify options activated during each reset. Version 1 allowed 20 such options; Version 2 includes 36 for the Model I and 33 for the Model III. A few Version 1 options are no longer defined.

Several options are available through a built-in keyboard intercept routine included in NEWDOS 80 Version 2. JKL prints the screen to a printer, 123 invokes Debug, and DFG invokes Mini-DOS (more on this later). For older Model I's a debounce routine makes separate software unnecessary. You can selectively enable or disable all four of these options. Disable the entire intercept routine via a System option or by pressing the up-arrow key during a reset/power-up.

You can specify the number of disk drives to avoid wasting time searching for files on non-existent drives. Set any drive as the default drive for the DIR command. When you save a file the system searches all drives to see if it already exists. If not, you can specify the drive number to begin searching for free space. No drive less than this will be written to. This feature prevents writing a file to a system disk in drive 0 if you forget the write protect tab.

You can set the number of tries at verifying a read or write to disk before declaring an error. The usual number is 10 but five saves time. You can also set the number of write-with-verify tries. With this option you can force the system to rewrite and try again. If you set this at two and the other at five, the system writes a file and attempts verification five times. If an error occurs the process repeats, *including* a second write, before declaring an error. Previously, if the error occured after five verifies, I/O stopped and you had to try again.

You can protect high memory at each reset rather than specify a memory size. You can make any key repeat, and adjust the time from first hitting a key until repeat starts. If your printer cannot accept characters above a given ASCII code you can replace them in the LPRINT stream with spaces or periods. You can enable a DOS command that repeats the last DOS command by entering R; this is a great timesaver when you want to format more than one disk or make multiple copies. To make disk backup easier, you can perform full disk copy commands without passwords even if passwords are enabled for program use.

You choose a blinking cursor and specify the cursor character. You can force the system to perform a separate verify of all tracks after Format is finished in addition to the track verify performed after each track is formatted. In double density this helps to ensure that a disk is usable. By setting another option to Y or N the system knows whether a lowercase modification has been installed. If so, a lowercase driver can be activated, again eliminating separate software. A system shift-lock can be set to force all uppercase. Since I use The Patch for lowercase I cannot test these functions. You can tell the system to ask for the date and time at each power-up. These questions can be asked or bypassed at each reset as well. Version 2 permits active CPU speed-up mods during disk operations; you may set timing loops internal to the system according to the actual speed-up time.

In a business environment the user may not understand how the system works. To avoid accidental Break and Clear functions, disable these keys. To protect programs or data, enable passwords and place the system in a run-only mode which disables Break, prohibits any Basic direct commands, and disables 123, DFG, and JKL. You must then use an auto command to select and run the desired program or execute a chain command file; the "DOS ready" message forces an endless loop in which case pressing R forces a reset. As with all other systems you can disable the Auto command by holding Enter at power-up or reset. You can also disable the Auto command by run-only, or you can selectively disable it by setting another option to N. If the system is brought up via a Chain file the commands are displayed as they are executed. To provide secrecy for program names and procedures, another option automatically eliminates all display, including DOS and Basic messages, until a program turns the display back on.

DOS Features

Machine language programmers can invoke a Debug routine by pressing 123 simultaneously or by executing Debug from the "DOS is ready" message to inspect or change memory or disk or to single step the program. In addition your machine language program can access a DOS Call routine to execute any DOS command or other program. This routine is documented, including its entry point and use of registers. NEWDOS allows you to insert an interrupt routine into the DOS interrupt chain.

Mini-DOS allows you to interrupt a main program to invoke one or more DOS library commands and then return to the main program where you left it. For example, by pressing DFG and entering Mini-DOS, you can examine directories and kill files while executing Scripsit even though these features were left out of Scripsit itself. You can then return to Scripsit by entering MDRET, or abort and return to the "DOS ready"

```
LOAD GSF/CIM
BASIC 1, 62168
DEFUSR = &HFE80
CLEAR
RUN "DIR/GSF"
```

Fig. 1. Chain File

```
DRIVE   1   A1A        08/23/81        39 TRKS          37 FDES            85 GRANS
                       EOF      LRL        RECS GRANS   EXTS               SIUEC....UAL
SUPERCOP/OBJ           05/088   256          06     2      1               ....C....UA0
EDTASM/CMD             35/000   256          35     7      1               ............0
SUPERCOP/SRC           28/134   256          29     6      2               ..UE........0
DISASSEM/CMD           25/000   256          25     5      4               ............0

NEWDOS/80 READY
```

Fig. 2. Directory Command Display

See List of Advertisers on page 290

"Mini-DOS supports chain files invoked from DOS or from a Basic program."

message by entering MDBORT. Caution: Do *not* interrupt disk I/O. You can execute only library commands, not programs, and use all commands except Append, Chain, Copy, and Format. A special Mini-DOS command, MDCOPY, copies one file at a time.

Mini-DOS supports chain files invoked from DOS or from a Basic program. In many instances a standard series of keyboard or disk input is necessary. The computer automatically processes these inputs in sequence whenever a chain file is called. An example from my own use is in Fig. 1. I wrote a disk directory program using the GSF sort routine. Entering the GSF initialization commands each time before running the program required looking up the correct procedure and a lot of typing. Now, by invoking the chain file via the Auto command CHAIN,CHAIN/JCL this is automatically completed at reset. Chain with NEWDOS is more versatile because you can separate each file into sections identified by section ID's. The appropriate section is then invoked by CHAIN,FILESPEC1,SECTION #. One file on a disk is used by several programs; separate sections are called as needed by one program. Chaining also allows message display and user input requests. DOS command DO performs exactly like Chain. You can manually pause chaining with the right arrow key, restart it with Enter, and cancel it with the arrow key. Apparat includes a Basic program, CHAINBLD/BAS, for use in writing Chain files. You can also use Scripsit or Electric Pencil.

Library Commands

NEWDOS 80 Version 2 supports all the usual TRSDOS library commands. Many are revised and several new commands have been added.

Use Attrib to prevent or allow DOS to allocate more space to a file during pro-

"...additions to Basic editing and programming features provide my Model I with most of the Model III capabilities."

gram execution, and to reallocate unused file space at a program Close to free the space. DOS marks each updated file in the directory. You can also use an Attrib parameter to remove or create this mark. You can remove the marks for all updated disk files at the same time by the PROT command with parameter RUF. Update marking is helpful in maintaining backup disks.

Blink enables and disables cursor blinking; Break does the same for the Break key. Clear clears all device routing, removes all user interrupts from the timing chain, can reset memory protection, and zeros memory in specified addresses. CLS performs the same as the Basic CLS.

The Create command allocates disk space for a file before it is used in a program. Designate in advance the file's name, logical record length, and allocation status. You can write a specified number of records consisting of 00.

DIR has parameters in addition to A, I, and S. U displays only those files marked as updated; "/ext" limits the display to files having the stated filename extension; and P sends the directory to the printer. With single drive systems it was not possible to obtain a directory on a formatted data disk since system files are not present. Now, entering DIR $ with a system disk on drive 0 prompts you to mount the target disk before the directory is shown. Under NEWDOS the A option provides much more information, as shown in Fig. 2. The last column provides for 12 flags; only eight are presently defined. These flags disclose password protection and protect level.

Killing unwanted files has always been a chore. NEWDOS provides Purge with two parameters to make life easier: USR leaves untouched all system and invisible files, while "/ext" removes all files with the specified extension. Purge is not automatic. It first displays each file specified by the parameters; you must respond Y to kill the file or N to leave it. A Q response ends purging.

Apparat has at last added a somewhat restricted device routine to Version 2. Routing is provided for the keyboard, printer, display, and null (transfer nothing) for the Model I; RS-232 input and output are added for the Model III. Routing can be to a specified memory address giving a user routine location. The best way to understand device routing is with examples. ROUTE PR DO: LPRINT'S go to the display (DO), not the printer (PR). ROUTE DO DO PR: display output goes to the display and the printer. ROUTE PR NL: discards printer output, NL = null. It is no longer necessary to have separate Print and LPRINT statements transmitting the same information.

NEWDOS does not provide a Backup function; instead, six different forms of the

"Because I often leave a GOSUB reference with an undefined line number. . . renumbering failed frequently unless I included dummy lines.

Copy command are available. The first four provide for single file copy, while five and six are for full disk copies. Form six, the most useful, allows a Copy By File (CBF) parameter. Check File With Operator (CFWO) displays each file name and prompts you for Yes copy, No copy, Restart, or Quit. The UPD parameter copies only those files marked updated. In addition to the ILF files mentioned in connection with zapping, Copy recognizes Exclusion List File (XLF). Any file name in an XLF file is not copied; all others are. Transferring programs from Version 1 to Version 2 disks is thus simplified. Via an XLF file on the NEWDOS disk your programs can be copied from old disks to Version 2 disks; all Version 1 programs are ignored.

For the Model III only, you can specify printer lines per page and characters per line with a Forms command. Setcom activates and deactivates the RS-232 interface, and can set or change word length, baud rate, stop bits and parity. Setcom directs whether the input routine should wait until an input byte is received or an output byte is sent.

Basic Enhancements

Apparat has enhanced older Basic abilities and added new ones. Specify the number of files, memory protect size, and a Basic command sequence from DOS or by an Auto command. Two direct editing commands have been added: a colon scrolls one display page toward the start of the text, while @ does the same toward the end. DI and DU move lines from one area to another. A Renew command recovers programs lost by an accidental New.

Chain programs without losing variables or file areas by appending ",V" at the end of a Run or Load command. All variables remain unchanged for the new program to use except DEFFN variables; open file areas are left open. Merge can be called from within a program and functions for ASCII or packed files while protecting variables.

Version 1 contained a built-in Renum command with an annoying feature: Renumbering failed if any undefined line numbers were present. Because I often leave a GOSUB reference with an undefined line number until I write the actual subroutine, renumbering failed frequently unless I included dummy lines. Now you can use Renum with a new X option. As long as the undefined line number is not within the range of lines renumbered the procedure will function. The Reference command is similarly enhanced; it still provides a cross reference table to variables or line numbers and you can now use it to find strings, any Basic key word, or a series of keywords such as GOTO or LPRINT.

NEWDOS 80 Version 2 incorporates

See List of Advertisers on page 290

"If you are a Version 1 owner who gave up on these new files, try again."

Basic CMD functions similar to Model III TRSDOS. Command C removes spaces or remarks; E displays error messages; J converts calendar dates from Julian to day-of-the-year and back; and T and R disable and enable interrupts on the Model I. T and R are not implemented for the Model III. As in Version 1, you can call CMD "DOSCMD" from Basic to execute most DOS library commands and then return to the Basic program.

Apparat provides a greatly enhanced Basic sort routine with command O. You can implement a direct sort on up to nine single or multi-dimensional arrays. In addition, an indirect sort builds a new integer array which forms a table of pointers to elements of other arrays in sorted order without changing the other arrays. You can then use the new arrays to read the other arrays in sorted order. Both sorts function in ascending or descending mode.

Further refinements to delight any veteran programmer are available through nine CMD F functions. To bail out of complex programming and prevent program crashes, F=POPS purges all For. . . Next and GOSUB-Return controls; F=POPR purges only the current GOSUB level and For. . . Next controls at the same level; F=POPN purges only the current For. . . Next control; and F=POPN vn purges the For. . . Next associated with variable vn and any controls set while the vn loop was active. Further F functions change string area size without clearing variables, clear specified variables, keep specified variables and clear all the rest, delete program lines without stopping execution, and swap values between two variables.

Apparat's new system also incorporates a Basic program single stepping routine to help debug programs. When you invoke this routine direct command or during program execution, the program pauses at the end of each line and displays the next line number on the screen. Enter executes the next line and updates the displayed number. You can specify the line number at which to commence single stepping. This routine is strictly line oriented; one full line executes with each Enter. If there are multiple statements per line the routine executes all statements and pauses at the next line. This may not be as powerful as Boss but is extremely useful.

Disk File Enhancements

TRSDOS defined sequential and random disk file structures. NEWDOS 80 Version 1 redefined these files and added two other file types with a total of five subtypes. Sequential files became print/input files; random files became field item files. Apparat added "marked item" files with subtypes MI, MU, and MF, and Fixed Item files with subtypes FI and FF. Because Apparat's documentation was poor, I doubt that very many Version 1 users took advantage of the vast file structure improvement.

Version 2 leaves all five file types. The new documentation is much better. Twelve pages and two separate appendices providing numerous examples of actual use accompany Version 2. Understanding these file structures is still not easy and requires time and effort, but no more than the original TRSDOS manual. If you are a Version 1 owner who gave up on these new files, try again. Their ease of use with experience will surprise you.

Disk Drives

PDRIVE allows various combinations of disk drive sizes and densities. Version 2 supports up to four physical drives; the PDRIVE table maintained by the system allows for ten. I do not know if we can expect more in the future, but the possibilities are exciting.

Version 2 supports standard single density disks along with PERCOM, LNW, and Apparat double density boards on 35, 40, 77, and 80 track single or dual headed five inch or eight inch drives. You can implement any combination. Version 2 treats double sided drives as a single disk with one directory and the tracks divided between the two sides. Mixing such drives with single sided drives may present problems, but the documentation says this has been done. Dual head drive users should contact Apparat for further information. Version 2 does not accept five inch double sided double density disks created with Version 1. You must transfer the files to new disks for use.

Double density presents certain problems with data transmission. To ensure that a formatted disk is stress tested, Apparat supplies two optional zaps using the byte 6DB6 for formatting in place of the standard Model I E5E5 or Model III 5B5B patterns. The manual states that up to 30 percent of disks not certified for double density may fail using the worst case 6DB6 formatting. Since I just added double density none of my old disks are DD certified. Using 6DB6, NEWDOS 80 Version 2, and the new Percom Doubler II, only three percent failed. Apparently Version 2 and the Doubler II are a perfect team!

The PDRIVE command specifies each drive's type, track count, head step rate, grans per lump, starting lump for the direc-

See List of Advertisers on page 290

tory, and directory size between two and six granules. You can set step pulses between tracks to allow 80 track drives to read 40 track disks. This is enough to accomodate any drive available.

Model III TRSDOS disks can *not* be read by NEWDOS 80 Version 2 due to a difference in disk structure. However, the Model III Convert utility works on Version 2 five inch single density disks if the directory is properly laid out. Also, Version 2's Copy command, combined with the proper PDRIVE specification, allows you to transfer files between the two systems. You can swap disks between Model I and Model III if you follow certain limitations and specifications.

Choosing an Operating System

Choosing an operating system in 1979 was easy since only NEWDOS 2.1 and TRSDOS were available. With the addition of LDOS, ULTRADOS, VTOS 4.0, and DOSPLUS the choice is more difficult.

NEWDOS 80 Version 2 represents an improvement over Version 1 and is a worthwhile investment. Its extensive additions to Basic editing and programming features provide my Model I with most of the Model III capabilities. Apparat goes to a great deal of time and effort to remove bugs from NEWDOS, but does not supply phone support personnel. Apparat also charges for new on-disk copies of NEWDOS zaps.

There are trade-offs in any operating system. The NEWDOS routing capabilities should suit most needs although they do not approach the niceties provided by the LDOS phantom device routing. LDOS also allows you to customize key entry. However, LDOS does not supply Disk Basic, choosing instead to apply patches to the user's TRSDOS Disk Basic which provide features incorporated by NEWDOS. To my knowledge no one supplies a facility in any way similar to NEWDOS Mini-DOS, or the various disk file structures briefly mentioned. NEWDOS support of the Percom Doubler II makes double density operation easy for the Model I. Also, NEWDOS will not crash a disk if a file is killed while open. This can happen with LDOS, and is completely disastrous with TRSDOS.

The best way to choose between all alternatives would be to use them, but expense prohibits this. For the same reason no one has yet provided a comprehensive side-by-side comparison article. NEWDOS 80 Version 2 still represents the state of the art. ■

See List of Advertisers on page 290

WINCHESTER HARD DISK for TRS-80™

Add 6.3 meg to 19 meg to your TRS-80

Integral Winchester Business System includes: 48K MODEL III, DOS, DISK III™, 6.3 MEG HARD DISK (internal).

$4995

$2895

6.3 MEGABYTE WINCHESTER HARD DISK SUBSYSTEM With Chassis, PS, DOS.
9.5 MEG $3395.00

VR Data Winchester HD Subsystem Features:

- TRSDOS Compatible Operating System
- State-of-the-Art Circuitry
- Full Calibration, Test and Burn-in
- Up to 4 drive configuration
- Heavy duty Power supplies
- Warranty-120 days-parts & labor
- 115/230 VAC, 50/60 Hz.

Optional Mod III Configurations:

9.5 Meg HD (internal)	add $500.00
80 tk 1 side floppy	add $120.00
80 tk 2 side floppy	add $240.00

Optional Winchester Subsystem configurations:

2 x 6.3 meg drives	$4495.00
2 x 9.5 meg drives	$5495.00

Model III Communications

COMING SOON

D-CON **$299.**

Integral Modem
NEEDS NO RS232
— Direct Connect
Programmable Dialing
AUTO ANSWER/ORIGINATE
Easy Installation

HERE NOW

VR-RS232C **$75.**

Direct Replacement For R.S. RS232
Fully Tested & Burned-In
Easy Installation
120 Day Warranty
Programmable Pincut
Prototype Area

Phone 800-345-8102 • in PA 215-461-5300

VR Data **VR Data Corporation**

For the Models I and III.

Radio Shack's Compiler Basic

Compiler Basic
Tandy/Radio Shack
Fort Worth, TX
Models I & III
$149

Richard C. McGarvey
221 Hirschfield Drive
Williamsville, NY 14221

Before you buy the Basic Compiler program, go to your Radio Shack store, read the documentation and spend several hours testing the software. You may find, like I did, that the program is not what you want.

What's in a Name? In this case—everything! This new software is not a Basic language compiler. It is Compiler Basic, an entirely new and different type of Basic that is designed to be compiled by this program.

Compiler Basic is composed of several programs that work together in the development and operation of a compiled program. The first part is RSBasic, the Basic language that the Compiler recognizes. It is similar, but not identical to Level II Basic. RSBasic does not support many Level II instructions. RSBasic also has many new and unique commands.

The difference between Basics is only a minor inconvenience. The important point is will you use the compiler often enough to warrant learning the new Basic? RSBasic is not compatible with Level II Basic. Level II Basic programs cannot be compiled without substantial changes.

The second part is the Run-Time program. That's right! You must have a Run-Time program in the computer before you can run any compiled program. This is an important point and I will cover it more fully later.

The third part of Compiler Basic is the compiler, a function of RSBasic. The compiler is invisible to the programmer. With the Run-Time, DOS and RSBasic programs in memory a 32K machine has about 1,500 bytes of programming room left. A 48K machine would have a little over 16K of programming room. Even with 48K you would have to keep your programs modest in size and use overlay or multi-program styles to implement a complicated programming application.

The fourth part of Compiler Basic is the TRSDOS operating system, version 2.3B. It is not compatible with any other DOS. Radio Shack supplies an upgrade module that will permanently change any previous TRSDOS to version 2.3B. If you use a different operating system forget it. If you have implemented speed modifications on your computer you must disable them to get the slow TRSDOS to operate.

The package contains additional modules such as BEDIT, an editor module for editing the Basic source code. These modules are mostly controlled by RSBasic and are fully covered in the documentation.

Documentation

Compiler Basic documentation is extensive. A binder 3½ inches thick holds a manual 1½ inches thick. The greatest part of this publication explains RSBasic.

I cannot say how well written or how complete the documentation is. I had no trouble understanding the program's operation but I did not use it during any major programming efforts. Interested buyers should stop by a store and read the documentation carefully before making the purchase. Be sure you can achieve your programming goals with the Compiler before you purchase it.

Speed?

I always thought that a compiled program was faster than its high-level language source. This does not seem to be the case here. ZBasic, Microsoft Basic Compiler and Accel 2 give varying speed increases. ZBasic is the fastest but also the most limited. Microsoft seems to be the slowest but the most versatile. Accel 2 appears to be the most well rounded compiler; it sacrifices some freedom to increase speed and some speed to retain versatility.

Expecting greater speed, I compiled a short test program that would indicate a start and stop point for timing the program's operation. While this simple program was not a very good test of the compiler it did give interesting results. The compiled program took 24.5 seconds to run. The same program under normal Disk Basic took 12.6 seconds.

If speed is your goal stay away from this compiler. Speed is not the goal of Radio Shack's Compiler Basic and it does not appear to be a secondary benefit.

Radio Shack says that this Compiler is business oriented and intended for use as a tool for the development of new software. They are serious about this and unless your goal is new business software then look elsewhere.

Operation

Operation is fairly simple. The manual says you will be prompted for date and time when you boot the system but in practice no such prompting occurs, at least not by the package I examined.

Next you load RSBasic. With features such as the Auto command you can enter your Basic program from the keyboard. If you have a Basic program already on disk you can load and finish it in RSBasic. RSBasic loads Level II Basic programs if they are on a 2.3B disk. It will not compile Level II programs.

After you write your program you can run it. This command first compiles and then runs the program for debugging. Next you can compile the program by issuing the Compile command. This compiles and saves the compiled program to disk under the name you give it.

One drawback is that the so-called compiled version you save on disk is not really compiled. The compiler in RSBasic compiles the program into *intermediate code*. Each time you want to run your program you must run it with the Run-Time program which compiles the intermediate code into final code. This leads to some problems when you try to sell your programs.

Selling Your Creations

Radio Shack is aware that you might

"To be fair, the program seems to function well and it appears to be properly documented."

wish to sell the programs you create with Compiler Basic. They do not charge royalties for the use of their compiler. Of course TRSDOS 2.3B and the Run-Time package are needed to operate the program you sell and they are copyrighted. You cannot sell copies of the Run-Time program without violating the copyright.

Radio Shack has generously provided a way around this dilemma. You can sell your program in one of two ways. First, you can buy a Run-Time disk for the Model I or Model III. These cost $20 each. You then put your program on the Run-Time disk and sell the Run-Time disk with your program on it. This method is fine if you plan to sell your program for a high enough cost to cover the $20 per copy Run-Time fee. If you plan to develop a complete business system that will sell for several hundred dollars you can simply mark it up $20 per copy.

The second way is to sell your program on a plain disk and tell your customer that he must buy the Run-Time disk in order to use it. Either way the cost of the Run-Time disk is passed on to the consumer. Once the customer has the Run-Time disk, regardless of which method is used, he can make as many copies of it as he wishes. Each customer need only pay for one Run-Time disk.

Remember, selling your program on a copy of the Run-Time disk violates the copyright.

Flexibility

I do not rate this program highly for flexibility. First, it recognizes only the exact syntax of RSBasic. If you have spent a great deal of time writing Level II programs expect to spend the same amount of time to make them work in RSBasic.

Second, you are restricted to TRSDOS 2.3B. No other DOS works. You cannot move the finished program to a different DOS. You could modify another DOS, however the time and expertise necessary to modify the receiving DOS to run correctly would make it an unlikely undertaking.

Third, you must have the Run-Time disk to run the finished product. This limits the salability of the product you create by establishing a minimum price of $20. Your profit is then tacked on top of that.

If Radio Shack has intended to create a very complicated way of making Basic programs secure from theft they have succeeded. With Compiler Basic you can write a Basic program that cannot be listed. The original source code will be almost totally indecipherable.

Value

Is the program valuable to the average programmer? Consider the following points:

- Cost—Compiler Basic costs $149. If you wish to sell your programs you must figure on a minimum cost of $20 plus profit per program disk.
- Flexibility—You are limited to one Basic, one DOS and you must have the Run-Time program resident in memory in order to operate your resultant program. If you wish to alter your program you must load the original Basic source and then edit or add to it as necessary. Others cannot alter your program to fit their needs. This lack of flexibility may turn off many prospective buyers.
- Security—If you lose the original Basic source you are out of luck. No one can tamper with your program but that is not necessarily a benefit. If the person buying the program wishes to make alterations he cannot, but he can make limitless copies of the program. While your source is safe the resultant program is not. There are many security tricks to protect Basic code. None are foolproof but all are free if you read the right magazines.
- Use—This program is for developing new, business-oriented software. As I pointed out earlier, the salability of such software is in doubt. If, on the other hand, you are writing the programs for personal use, then why not just use Level II Basic? It appears to be faster and you already know it!
- Size—While the Run-Time program and the application program can be quite large in theory, in practice they are limited by the size of the compiler. You cannot develop a program that will run in a 32K machine with room to spare on a 32K machine. In fact, a 48K machine can compile only a 17K program. This means that the largest program that can be compiled will be small when in actual operation. A serious business system is about 32K. To use this efficiently the application program should be as complete as possible. The Compiler limits the program size so that each program will have to access the disk to get overlays or additional programs. Since you cannot merge compiled programs you must write your programs in a segmented style. While most programs can be kept under 17K it is not always the best programming method. This is especially true of larger business applications.

I think Radio Shack's new Compiler Basic is a poor investment. To be fair the program seems to function well and it appears to be properly documented. If you need this type of program then Compiler Basic is a good risk. It does exactly what it claims. But average programmers might purchase the program expecting something that they will not get. ■

✓See List of Advertisers on page 290

LOAD 80

This one will boggle your mind!

Tumblin' Dice

Ronald H. Bobo
3246 Gravois
St. Louis, MO 63118

Boggle, a word game from Parker Brothers, is addictive. You can play it alone or in competition with others. You need a pencil and paper for each player, sixteen dice whose sides bear letters of the alphabet rather than numbers, and a three-minute egg timer.

To play the game shake the dice within their transparent container. Next form as many words as possible using the letters on the tops of the dice. When all the sand runs through the glass, stop and figure your score.

The dice are left in the container to maintain the 4 by 4 layout of letters. Words may be formed by combining adjacent letters forward, backward, up, down, or on the diagonal. No single letter may be used more than once in a word, but where letters are duplicated, both may be used.

Why go through all that strenuous exercise shaking dice when your TRS-80 can do it for you? Where is the egg timer when you really need it? Let the TRS-80 keep time. In the heat of the game, it's fairly easy to forget the egg timer when you are concentrating on the dice. With the TRS-80 keeping time, it's almost impossible not to notice when time is up.

I wrote Boggler on the Model I, but it also should run on the Model III. The Basic is straightforward and will work on tape or disk systems.

The program contains complete rules for Boggler. After the title screen, you may go through the instructions or skip to the main routine. Following the instructions, enter the number of players. I suggest no more than ten players in a game, since the array is not dimensioned.

After you enter the names, the program prints sixteen letters in a 4 by 4 format using special effects. You can check the time on the clock printed underneath the letters.

At this point, start to form your words and write them down on paper.

The program will give warnings at two minutes 30 seconds and at two minutes 50 seconds. After the three minutes are up, the program prints further instructions. Comply with these instructions and touch any key to return to the scoring table, where scores are figured and entered into memory.

Once all scores have been entered, you may play another round or quit. If you play more than one round, the computer keeps a score for each round and a cumulative score.

When you quit, the program sorts the cumulative scores. It then lists players, their scores, and announces the winner. It will also announce if there has been a tie for first.

All single-key input is by INKEY$, making it easy to step through the program. ■

The Key Box

Basic Level II
Model I
Tape or disk

Program Listing

```
10 REM * BOGGLER *
20 REM * SUGGESTED BY THE PARKER BROTHERS' GAME "BOGGLE"
30 REM * PROGRAMMED BY RONALD H BOBO, ST LOUIS, MO, JAN 1979
40 REM * REVISION 2, AUGUST 1981
50 CLS
60 CLEAR 200
70 RANDOM
80 PRINT@475,"BOGGLER":PRINT@542,"BY"
:PRINT@600 ,"RONALD H BOBO"
90 FOR X=25 TO 100:SET(X,15):SET(X,36):NEXT
100 FOR Y=15 TO 36:SET(25,Y):SET(100,Y):NEXT:
FOR T=1 TO 1500:NEXT
110 CLS:PRINT@128,STRING$(63,"=");:PRINT@832,STRING$(63,"=");
120 PRINT@450,"WOULD YOU LIKE INSTRUCTIONS?  (TYPE Y FOR YES, N
FOR NO)...
130 A$=INKEY$:IF A$="" THEN 130 ELSE IF A$="Y" THEN GOSUB 1050 E
LSE CLS
140 GOSUB 1720
150 DIM A$(96)
160 REM * READ CHARACTER SET
170 FOR C=1 TO 96:READ A$(C):NEXT C
180 DATA A,A,A,A,A,A,A,A,B,B,B,C,C,C,D,D,D,D,E,E,E,E,E,E,E
190 DATA E,E,E,F,F,G,G,G,H,H,H,I,I,I,I,I,I,J,K,K,L,L,L,L,L
200 DATA M,M,M,N,N,N,N,N,O,O,O,O,O,O,P,P,P,Q,R,R,R,R,S,S,S,S,S
210 DATA T,T,T,T,T,U,U,U,U,V,V,W,W,X,Y,Y,Y,Z
220 REM *  PRINT FIRST ROW
230 P=4
240 PRINTCHR$(23)
250 PA=24
260 V=RND(96)
270 IF A$(V)="" THEN 260
280 A$(V)=A$(V)+CHR$(32)
290 FOR X=PA TO PA+7*64 STEP 64:PRINT@X,A$(V);:FOR Y=1 TO 25:NEX
T:PRINT@X,"  ";:NEXT X:PRINT@X,A$(V);
300 PA=PA+4
310 A$(V)=""
320 P=P-1
330 IF P<>0 GOTO 260
340 REM *  PRINT SECOND ROW
350 P=1
360 V=RND(96)
370 IF A$(V)="" THEN 360
380 A$(V)=A$(V)+CHR$(32)
390 ON P GOTO 400,420,440,460
400 FOR X=576 TO 602 STEP 2:PRINT@ X,A$(V);:FOR Y=1 TO 25:NEXT:P
RINT@X ,"  ";:NEXT X:PRINT@X,A$(V);
410 A$(V)="":P=P+1:GOTO 360
420 FOR X=638 TO 610 STEP -2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:P
```

Program continues

Two GREAT programs by Bruce Hansen!

ADVENTURE the system

BRAND NEW!

Create your own adventures that look and perform exactly like those written by Scott Adams! The Adventure System is more than just a program -- it's a unique experience in creativity! Included in The Adventure System package are:

- An adventure editor which allows you to design, debug, read and write your own adventure creations!
- An adventure driver for executing your adventure creations!
- Two full length adventures: "Miner's Adventure" and "Burglar's Adventure"
- A 50+ page manual in a handsome notebook describing the adventure language and how to use it; includes a free mini-adventure!
- An invitation to join the adventurer's user group which will entitle you to free program updates, user tips, a special discount on adventures, a user newsletter, and special help getting the adventures you write marketed

The Adventure System actually utilizes a unique language that permits you to input criteria for building an adventure game to your specifications. You are limited only by your imagination!! Challenge, amaze and entertain your friends with your own adventure programs! Adventure characters can include your family or friends; adventures can take place at home, work, campus or your own city!

The Adventure System sells for
just $39.95
Please specify Model I or III
Requires 1 drive and 48K RAM

For the cost of three adventures, TAS lets you create dozens !! In stock and ready to ship NOW.

TASMON

The Alternate Source MONitor

Overview: TASMON is an interactive Z-80 monitor and disassembler. All versions come complete with tape and disk I/O, and allow programs to be easily converted from one medium to the other. Source code generated by TASMON can be loaded by all popular editor/assembler programs. TASMON features a command to fully relocate itself to any memory block you specify.

Purpose: The purpose of TASMON is to allow study, debugging, tracing and single-stepping of Z-80 object code. Special care has been taken to make the program easy to use by beginning Z-80 programmers. TASMON was developed as a result of the author's dissatisfaction with the seven monitors he purchased. TASMON has a 45+ page user manual documenting each of its features, and includes special user sessions which document the more complicated features.

"other excellent disassemblers are ... The Alternate Source's TASMON, which provides symbols and disk files. (The TASMON package is a powerful monitor, one of the best I've seen.)"
William Barden, Jr.

Command Summary:

- Replace registers
- Modify memory
- Hex memory dump
- ASCII memory dump
- Disassembled dump
- Disassemble to printer
- Dump screen to printer
- Sum hex values
- Subtract hex values
- Find 1-4 consecutive bytes
- Skip forward one instruction
- Back up one instruction
- Clear screen
- Relocate programs
- Move block of memory
- Load system tape
- Load /CMD disk file
- View/verify system tape
- View/verify /CMD disk file
- Write system tape
- Write /CMD disk file
- Disassemble to disk
- Disassemble to tape
- Set breakpoints in ROM
- Set breakpoints in RAM
- Set breakpoints (9 total)
- Display breakpoints
- Clear breakpoints
- Single stepping (two ways)
- Tracing at 8 speeds

Plus:

- Keep screen
- Split screen display
- Back/Forward pagination
- Break after n executions

Disassembles with labels!

TASMON I & III tape
TASMON I disk
TASMON III disk
$29.95

The Alternate Source is always accepting subscribers to their magazine. But they aren't always giving away free programs to subscribers. Just now and then. Before March 31, 1982 each person subscribing/renewing for a 24 issue period to TAS, and mentioning this advertisement, will receive absolutely free, a program called **FLY.** You will find this program to be literally **full of bugs** yet containing no errors. **Swarming** with action, it's sure to amuse you for hours on end. **FLY** is not for sale anywhere. It is available only through this and similar advertisements, offer good until March 31, 1982. Twenty four issues of TAS are $36.00, **FLY** is free. You'll swat yourself if you miss this one!

Name: ____________
Address: ____________
City: ____________
State, Zip: ____________

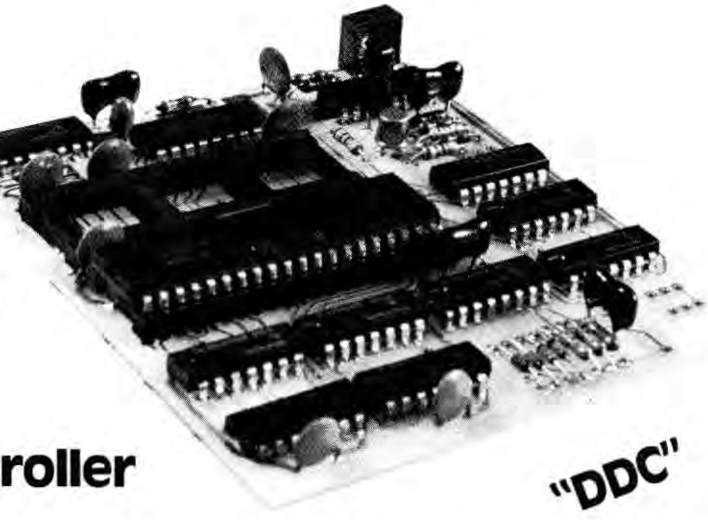
The Alternate Source
1806 Ada Street
Lansing, Michigan 48910
517-487-3358
517-485-0344 ✓138

- ☐ Rush me a TAS I disk!
- ☐ Rush me a TAS III disk!
- ☐ Rush me a TASMON tape!
- ☐ Rush me a TASMON I disk!
- ☐ Rush me a TASMON III disk!

Add $3.00 for packaging & postage.
Add $1.75 for COD shipments.
Add 4% for charge card orders.

```
RINT@X," ";:NEXT X:PRINT@X,A$(V);
430 A$(V)="":P=P+1:GOTO 360
440 FOR X=576 TO 598 STEP 2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:PR
INT@X," ";:NEXT X:PRINT@X,A$(V);
450 A$(V)="":P=P+1:GOTO 360
460 FOR X=638 TO 614 STEP -2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:P
RINT@X," ";:NEXT X:PRINT@X,A$(V);
470 A$(V)=""
480 REM *  PRINT THIRD ROW
490 P=1
500 V=RND(96)
510 IF A$(V)="" THEN 500
520 A$(V)=A$(V)+CHR$(32)
530 ON P GOTO 540,560,580,600
540 FOR X=640 TO 666 STEP 2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:PR
INT@X," ";:NEXT X:PRINT@X,A$(V);
550 A$(V)="":P=P+1:GOTO 500
560 FOR X=702 TO 674 STEP -2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:P
RINT@X," ";:NEXT X:PRINT@X,A$(V);
570 A$(V)="":P=P+1:GOTO 500
580 FOR X=640 TO 662 STEP 2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:PR
INT@X," ";:NEXT X:PRINT@X,A$(V);
590 A$(V)="":P=P+1:GOTO 500
600 FOR X=702 TO 678 STEP -2:PRINT@X,A$(V);:FOR Y=1 TO 25:NEXT:P
RINT@X," ";:NEXT X:PRINT@X,A$(V);
610 A$(V)=""
620 REM *  PRINT FOURTH ROW
630 PA=984
640 FOR P=1 TO 4
650 V=RND(96)
660 IF A$(V)="" THEN 650
670 A$(V)=A$(V)+CHR$(32)
680 FOR X=PA TO PA-3*64 STEP -64:PRINT@X,A$(V);:FOR Y=1 TO 25:NE
XT:PRINT@X," ";:NEXT X:PRINT@X,A$(V);
690 PA=PA+4:A$(V)=""
700 NEXT P
710 ' START CLOCK PROGRAM
720 PRINT@256,STRING$(31,"-");:PRINT@960,STRING$(31,"-");
730 C=0
740 D=INT(C/10):C=C-(D*10)
750 A=0
760 B=INT(A/10):A=A-(B*10)
770 FOR N=1 TO 500:NEXT
780 A=A+1
790 IF C=2 AND B=3 AND A=5 THEN PRINT@0,CHR$(30);
800 IF C=2 AND B=5 AND A=5 THEN PRINT@0,CHR$(30);
810 IF A>9  GOTO  830
820 GOTO 910
830 A=0
840 IF C=2 AND B=2 AND A=0 THEN GOSUB 2300
850 IF C=2 AND B=4 AND A=0 THEN GOSUB 2310
860 B=B+1
870 IF B>5 GOTO 890
880 GOTO910
890 B=0
900 C=C+1
910 PRINT@916,C;":";B;A;
920 IF C=3 GOTO 940
930 GOTO 770
940 PRINT"STOP"
950 FOR Z=1 TO 200:NEXT Z
960 PRINT@ 896,"CHECK ANSWERS TO MAKE SURE"
970 PRINT"ALL WORDS ARE CONTAINED IN"
980 PRINT"THE DISPLAY.  AFTER CROSSING"
990 PRINT"OUT DUPLICATES, PRESS ANY KEY"
1000 PRINT"TO VIEW SCORING TABLE."
1010 A$=INKEY$:IF A$="" THEN 1010
1020 GOSUB 1440
1030 GOTO 1870
1040 GOTO1040
1050 CLS:PRINT:PRINT:PRINT"EACH PLAYER WILL NEED PENCIL AND PAPE
R."
1060 PRINT:PRINT"OBJECT OF THE GAME IS TO LIST, WITHIN THE TIME
LIMIT,"
1070 PRINT"AS  MANY WORDS  AS YOU CAN FIND  AMONG THE RANDOM"
1080 PRINT"ASSORTMENT OF LETTERS."
1090 PRINT@896,STRING$(63,"=");
1100 PRINT@960,"PRESS  ANY KEY  TO CONTINUE...";
1110 A$=INKEY$:IF A$="" THEN 1110 ELSE CLS
1120 PRINT:PRINT"WHEN THE TIMER STARTS, EACH PLAYER LOOKS FOR WO
RDS OF"
1130 PRINT"THREE OR MORE LETTERS.  WHEN A PLAYER  DISCOVERS A WO
RD,"
1140 PRINT"IT IS WRITTEN DOWN.  PLAYERS WILL  HAVE APPROXIMATELY
"
1150 PRINT"THREE MINUTES FROM THE TIME THE LETTERS ARE DISPLAYED
."
1160 PRINT"A 'Q' IS ALWAYS TO BE CONSIDERED 'QU' SINCE 'Q' IS AL
WAYS
FOLLOWED BY 'U' IN ENGLISH WORDS.  THE 'U' WAS LEFT OUT SO AS
NOT TO DISRUPT THE DISPLAY WITH IMPROPER SPACING."
1170 PRINT@896,STRING$(63,"=");
1180 PRINT@960,"PRESS  ANY KEY TO CONTINUE...";
1190 A$=INKEY$:IF A$="" THEN 1190
1200 CLS:PRINT:PRINT:PRINT"WORDS  ARE FORMED FROM ADJOINING LETT
ERS.  LETTERS"
1210 PRINT"MUST JOIN  IN THE PROPER SEQUENCE TO SPELL  A  WORD."
1220 PRINT"THEY MAY JOIN HORIZONTALLY, VERTICALLY, OR DIAGONALLY
,"
1230 PRINT"TO THE LEFT, RIGHT,  OR UP-AND-DOWN.  NO SINGLE LETTE
R,"
1240 PRINT"HOWEVER, MAY  BE  USED MORE THAN ONCE WITHIN A SINGLE
"
1250 PRINT"WORD.  OF COURSE, ANOTHER  LETTER OF  THE  SAME TYPE"
1260 PRINT"MAY BE USED, SO LONG AS IT IS  ON A DIFFERENT ADJOINI
NG"
1270 PRINT"SQUARE."
1280 PRINT@896,STRING$(63,"=");
1290 PRINT@960,"PRESS ANY KEY TO CONTINUE...";
1300 A$=INKEY$:IF A$="" THEN 1300
1310 CLS:PRINT:PRINT:PRINT:PRINT"ANY  WORD,  INCLUDING PLURALS,
FORMS  AND TENSES"
1320 PRINT"IS ACCEPTABLE SO LONG AS IT CAN BE FOUND IN A STANDAR
D"
1330 PRINT"ENGLISH DICTIONARY.  WORDS WITHIN WORDS  ARE ALSO"
1340 PRINT"ALLOWED; E.G. SPARE:  SPA, PAR,  ARE, SPAR, PARE."
1350 PRINT:PRINT"WHEN THE TIMER STOPS,  EVERYONE QUITS WRITING.
 EACH"
1360 PRINT"PLAYER THEN READS ALOUD HIS/HER LIST OF WORDS.  ANY"
1370 PRINT"WORD APPEARING ON MORE THAN ONE LIST MUST BE CROSSED"
1380 PRINT"OFF ALL LISTS, INCLUDING THAT OF THE READER."
1390 PRINT@896,STRING$(63,"=");
1400 PRINT@960,"PRESS ANY KEY TO CONTINUE...";
1410 A$=INKEY$:IF A$="" THEN 1410
1420 GOSUB 1440
1430 GOTO 1520
1440 CLS:PRINTTAB(28)"SCORING"
1450 PRINT:PRINT:PRINT:PRINT"NO. OF LETTERS  3   4   5   6   7   8 OR
 MORE"
1460 PRINT:PRINT:PRINT:PRINT:PRINT"POINTS:          1   1   2   3   5
  11"
1470 FOR X=0 TO 80:SET(X,20):NEXT
1480 PRINT@896,STRING$(63,"=");
1490 PRINT@960,"PRESS ANY KEY TO CONTINUE...";
1500 A$=INKEY$:IF A$="" THEN 1500
1510 RETURN
1520 CLS:FOR X=1 TO 15:PRINTCHR$(23):PRINT@466,"REMEMBER":FOR Y=
1 TO 50:NEXT:PRINT@466,STRING$(8,32):FOR Y=1 TO 50:NEXT:NEXT X
1530 CLS
1540 PRINT:PRINT"MULTIPLE MEANINGS OF THE SAME WORD DO NOT EARN"
1550 PRINT"MULTIPLE CREDIT."
1560 PRINT:PRINT"THE SAME WORD FOUND IN DIFFERENT AREAS  OF THE
 GRID"
1570 PRINT"DOES NOT COUNT FOR MULTIPLE  CREDIT."
1580 PRINT:PRINT"THE  'Q' COUNTS AS TWO LETTERS, SINCE IT IS CON
SIDERED TO BE
FOLLOWED BY THE 'U'."
1590 PRINT:PRINT"FULL CREDIT IS ALLOWED FOR BOTH THE SINGULAR  A
ND PLURAL"
1600 PRINT"FORMS OF A NOUN  - PROVIDED THEY ARE LISTED AS SEPARA
TE WORDS."
1610 PRINT@896,STRING$(63,"=");
1620 PRINT@960,"PRESS ANY KEY TO CONTINUE...";
1630 A$=INKEY$:IF A$="" THEN 1630
1640 CLS:PRINT:PRINT:PRINT:PRINT"COMMON WORDS TEND TO BE FOUND
BY MORE THAN"
1650 PRINT"ONE  PLAYER.  IF YOUR WORDS ARE UNIQUE AND UNUSUAL, T
HEY"
1660 PRINT"ARE MORE LIKELY TO EARN POINTS."
1670 PRINT@896,STRING$(63,"=");
1680 PRINT@960,"PRESS ANY KEY TO CONTINUE...";
1690 A$=INKEY$:IF A$="" THEN 1690
1700 RETURN
1710 STOP
1720 CLS:PRINT@192,STRING$(63,"=");:PRINT@768,STRING$(63,"=");
1730 PRINT@470,"HOW MANY PLAYERS?"
1740 ANS$=INKEY$:IF ANS$="" THEN 1740 ELSE NN=VAL(ANS$)
1750 PRINT@470,CHR$(30);:PA=960
1760 PRINT@192,STRING$(63,"=");:PRINT@768,STRING$(63,"=");
1770 FOR PL=1 TO NN
1780 PRINT@464,"TYPE NAME OF PLAYER NUMBER";PL
1790 INPUT NA$(PL)
1800 PRINT@PA,NA$(PL);:IF NN=1 THEN PRINT" IS PLAYING ALONE";:EL
SE PA=PA+16:IF PL=4 OR PL=8 OR PL=12 THEN PRINT:PRINT@384,CHR$(3
0);:PRINT@448,CHR$(30);:PA=960
1810 NEXT PL
1820 PRINT@512,CHR$(30);:FORX=1 TO 500:NEXT
1830 PRINT@464,CHR$(30);
1840 PRINT@460,"PRESS ANY KEY TO PLAY...GOOD LUCK!"
1850 A$=INKEY$:IF A$="" THEN 1850
1860 CLS:RETURN
1870 FOR PL=1 TO NN
1880 CLS:PRINT@470,"ENTER ";NA$(PL);"'S SCORE..."
1890 INPUT SC(PL)
1900 TS(PL)=TS(PL)+SC(PL)
1910 NEXT PL
1920 CLS:PRINT:PRINT:PRINT
1930 PRINT"PLAYER","CURRENT SCORE","TOTAL SCORE"
1940 FOR PL=1 TO NN
1950 PRINTNA$(PL),SC(PL),TS(PL)
1960 NEXT PL
1970 PRINT@960,"TO PLAY ANOTHER ROUND, TYPE Y, OR TYPE ANY"
1980 PRINT"OTHER CHARACTER TO END THE GAME AND GET THE FINAL SCO
RE."
1990 A$=INKEY$:IF A$="" THEN 1990 ELSE IF A$="Y" THEN 2010
2000 GOTO 2020
2010 CLS:RESTORE:GOTO 160
2020 CLS:PRINT:PRINT TAB(25)"FINAL STANDING":PRINT
2030 FOR PL=1 TO NN-1
2040 FOR J=PL+1 TO NN
2050 IF TS(PL)>=TS(J) THEN 2090
2060 S=TS(PL):S$=NA$(PL)
2070 TS(PL)=TS(J):NA$(PL)=NA$(J)
2080 TS(J)=S:NA$(J)=S$
2090 NEXT J
2100 NEXT PL
2110 IF TS(1)=TS(2) THEN 2200
2120 PRINT TAB(15)"THE WINNER IS ";NA$(1);" WITH";TS(1);"POINTS.
"
2130 PRINT:PRINT
2140 FOR PL=1 TO NN
2150 PRINT"",NA$(PL),"",TS(PL)
2160 NEXT
2170 PRINT@960,"NICE GAME... LET'S PLAY AGAIN SOON."
2180 GOTO 2180
2190 STOP
2200 PRINT TAB(23)"WE HAVE A TIE GAME"
2210 PRINT TAB(22)"HERE ARE THE SCORES:"
2220 PRINT:PRINT
2230 FOR PL=1 TO NN
2240 PRINT"",NA$(PL),"",TS(PL)
2250 NEXT PL
2260 PRINT@960,"PLAY A TIEBREAKER ROUND? (TYPE Y OR N).
2270 A$=INKEY$:IF A$="" THEN 2270 ELSE IF A$="Y" THEN 2010
2280 GOTO 2170
2290 END
2300 PRINT@14,"30 SECOND WARNING";:RETURN
2310 PRINT@22,"10 SECONDS";:RETURN
```

Aerocomp's Proven Best-By Test! The "DDC"



Double Density Controller

★ Technical Superiority

At last! A double density controller for Model I with HIGHER PROBABILITY OF DATA RECOVERY THAN WITH ANY OTHER DOUBLE DENSITY CONTROLLER ON THE MARKET TODAY! The "DDC" from Aerocomp. No need to worry about the problems that keep cropping up on existing products. AEROCOMP'S new analog design phase lock loop data separator has a wider capture window than the digital types currently on the market. This allows high resolution data centering. The finest resolution available with digital circuitry is 125 ns (nano seconds). The "DDC" analog circuit allows infinately variable tuning. Attack and settling times are optimum for 5-1/4 inch diskettes.
The units presently on the market use a write precompensation circuit that is very "sloppy". Board to board tolerance is extremely wide - in the order of ± 100 ns. The "DDC" is accurate to within ± 20 ns.
The bottom line is state of the art reliability!

★ Test Proven

Tests were conducted on AEROCOMP'S "DDC", Percom's "Doubler A"* and "Doubler II"* and LNW's "LNDoubler"** using a Radio Shack TRS80*** Model I, Level 2, 48 K with TRS80 Expansion Interface and a Percom TFD100* disk drive (Siemens Model 82). Diskette was Memorex 3401. The test diskette chosen was a well used piece of media to determine performance under adverse conditions. The various double density adapters were installed sequentially in the expansion interface.
The test consisted of formatting 40 tracks on the diskette and writing a 6DB6 data pattern on all tracks. The 6DB6 pattern was chosen because it is recommended as a "worst case" test by manufacturers of drives and diskettes. An attempt was then made to read each sector on the disk once - no retrys. Operating system was Newdos/80, Version 1.0, with Double Zap, Version 2.0. Unreadable sectors were totalled and recorded. The test was run ten times with each double density controller and the data averaged. Test results are shown in the table.

★ Features

TRS80 Model I owners who are ready for reliable double density operation will get (1) 80% more storage per diskette, (2) single and double density data separation with far fewer disk I/O errors, (3) single density compatibility and (4) simple plug-in installation. Compatible with all existing double density software.

★ Value

$149.95 for the BEST double density controller on the market.

$189.95 for "DDC" complete with DOSPLUS 3.3D

$239.95 for "DDC" complete with LDOS

★ TEST RESULTS ★

MFR & PRODUCT	SECTORS LOCKED OUT (AVG)
AEROCOMP "DDC"	0
PERCOM "DOUBLER II"	18
PERCOM "DOUBLER A"	250
LNW "LNDOUBLER"	202

Note: test results available upon written request. All tests conducted prior to 8-25-81

Aerocomp's 14 day money back guarantee applies to hardware only.
Specials will be prorated. Shipping $2.00 in Cont. US. See opposite page for details.

Data Separators

The advances that make the "DDC" great are incorporated in the new AEROCOMP Single Density Data Separator ("SDS") and Double Density Data Separator ("DDS").

★ Has your original manufacturer left you holding the bag?

If you already own a Percom "Doubler A", "Doubler II" or LNW "LNDoubler", the AEROCOMP "DDS" will make it right. Look at the test results:

MFR. & PRODUCT	SECTORS LOCKED OUT	
	WITHOUT "DDS"	WITH "DDS"
PERCOM "DOUBLER II"	18	1
PERCOM "DOUBLER A"	250	0
LNW "LNDOUBLER"	202	0

Note: Same test procedures as "DDC".
* Trademark of Percom Data Co.
** Trademark of LNW
*** Trademark of Tandy Corporation

★ "DDS" $49.95
(Use 1791 chip from your DD Controller)

★ "DDS" with disk controller chip included **$79.95**

★ Disk controller chip $34.95

(Shipping $2.00 Cont. US - see opposite page for details)

Plugs directly into your existing Double Density Controller.

Do you need a Single Density Data Separator?

The internal data separator in the WD1771 chip (R/S Expansion Interface) is NOT recommended by WD for reliable data transfer. Do you have any of these problems: Lost data, tracks locked out, CRC errors, disk retry? YOU NEED ONE!

★ "SDS" $29.95
(For Mod. I; shipping $2.00)

See opposite page ▸▸▸▸▸▸

TRS 80*

DISK DRIVES

40 & 80 TRACK

as low as $299.95

NEW LOW PRICES

Thanks to you our sales volume has allowed us to cut costs and we're passing the savings along. We offer the best disk drive value in the market place. Reliability, features and cost tough to beat. We deliver...and we stand behind our products. AEROCOMP is the only manufacturer to offer FREE TRIAL! Buy Aerocomp drives today. You deserve the best.

BEST FEATURES

- ★ Fast 5 ms. track-track access time.
- ★ Operates single or double density.
- ★ "FLIPPY" feature for lower media cost (40-1 & 80-1).
- ★ Head load solenoid.
- ★ Disk ejector.
- ★ Easy entry door.
- ★ <u>NEW</u> EXTERNAL DRIVE CABLE CONNECTION. (no longer need to remove the cover to hook up cable)

EPSON MX80 PRINTERS

$469.00

TRS80 Cable (I & III). $29.95
APPLE Cable & Interface. . $96.00
RS232 Interface. $69.95
Ribbon Cartridge, Black. . . $13.95
Add $6.00 shipping (Cont. U.S.) does not include cable.

MX80 F/T

All the features of MX80 plus Friction Feed. Shipping & cables as above.

$579.00

ORDER TODAY!

- ★ 40-Track Drive. $299.95
- ★ 80-Track Drive. $429.95
- ★ 40-Track "FLIPPY" Drive. . . . $329.95
- ★ 80-Track "FLIPPY" Drive. . . . $449.95
- ★ 40-Track Dual-Head Drive. . $449.95
- ★ 80-Track Dual-Head Drive. . $579.95

All above drives are complete with silver enclosure, power supply and external drive cable connection. 115 VAC, 50/60 Hz. 230 VAC 50/60 Hz available on special order.

- ★ 40-Track Bare Drive. $269.95
- ★ 80-Track Bare Drive. $399.95
- ★ 40-Track "FLIPPY" Bare Drive $299.95
- ★ 80-Track "FLIPPY" Bare Drive $419.95

★ SPECIAL PACKAGES ★

STARTER A. $339.95
40-Track Drive, 2-drive cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

SRARTER B. $369.95
40-Track "FLIPPY" Drive, cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

COMBO C. $465.00
40-Track "FLIPPY" Drive, 2-drive cable, **LDOS**, Freight & Ins.

COMBO D. $585.00
80-Track "FLIPPY" Drive, 2-drive cable, **LDOS**, Freight & Ins.

COMBO E. $709.00
Two 40-Track "FLIPPY" Drives, 4-Drive cable, TRSDOS 2.3 Disk & Mannual, Freight & Ins.

COMBO F. $999.00
Two 80-Track "FLIPPY" Drives, 4-drive **LDOS**, Freight & Ins.

COMBO G. $909.00
Two 40-Track Dual-Head Drives, cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

COMBO H. $1299.00
Two 80-Track Dual-Head Drives, cable, **LDOS**, Freight & INS.

DOSPLUS 3.3 or NEWDOS / 80,2.0 available with any package.

Add $5.00 per drive for shipping & handling (Cont. US).

FREE TRIAL OFFER

Order your AEROCOMP Disk Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return it, packed in the original shipping container, for a full refund. (Special packages will be prorated.) We have complete confidence in our products and we know you will be satisfied! **ORDER TODAY!!**

WARRANTY

We offer you a 120 day unconditional warranty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes nescessary, our service department is fast, friendly and cooperative.

100% TESTED

AEROCOMP Disk Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!!

ORDER NOW!!

To order by mail, specify Model Number(S) of Drive, cable, etc. (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request C.O.D. shipment. Texas residents add 5% sales tax. Add $5.00 per drive for shipping & handling (Cont. US). Please allow 2 weeks for personal checks to clear our bank. No personal checks will be accepted on C.O.D. shipments-cash, money orders or certified checks only. You will receive a card showing the exact C.O.D. amount before your shipment arrives. Be sure to include your name and shipping address. You will be notified of the scheduled shipping date. Your bank credit card will NOT be charged until the day we ship.!!

DRIVE CABLES

2-DRIVE. $24.95
4-DRIVE. $34.95

WRITE AEROCOMP TODAY FOR MORE VALUES !!!

CALL TOLL FREE FOR FAST SERVICE (800) 824-7888, OPERATOR 24

FOR VISA/MASTERCHARGE/C.O.D. ORDERS

California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.

TOLL FREE LINES WILL ACCEPT ORDERS ONLY!

For Applications and Technical information, call (214) 337-4346 or drop us a card.

Dealers inqiries invited

AEROCOMP
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

✓387

✓See List of Advertisers on page 290

Speed up writing machine-language programs.

Microsoft Macro Assembler

G. Gratzer
75 Laval Drive
Winnipeg, Man. R3T 2X8
Canada

For some time I have been working on a book developing Fast Basic, a systematic merging in Basic with assembly-language subroutines. The Assembly-language subroutines are easy to write if you can overcome your impatience: The code is extremely repetitious. It was a natural thought then to use a macro assembler to write these subroutines, so I ordered one made by Microsoft.

It accepts files prepared by the Editor, so I put the EDIT/CMD file on my NEWDOS/80 disk and called the Editor. EDIT-80 then asks for a file and I typed TEST Break. According to the manual, Break creates a new file while Enter calls an old file. Unfortunately, hitting Break produced no response. I suddenly remembered NEWDOS/80 handles Break in a strange way. After running into problems with a modified NEWDOS/80, I decided to use the Microsoft system under TRSDOS 2.3.

The Editor is easy to use. It has two modes: Interline Editing and Alter. Interline editing has 16 commands: Insert, Delete, Print (on the screen), List (print on the lineprinter), reNumber, Replace (for delete and insert), and some others.

Alter put me into the editing mode. Just as with all the commands, Alter can be followed by any range of line numbers. The Alter mode has about 20 commands including insert characters, spaces; move the cursor; and delete characters or text. It took me awhile to get used to the Alter mode. The line being edited is shown only up to the cursor and I could not see what I wanted to edit. If the cursor was too far down the line, I automatically used the back arrow to backtrack (as in the Interline mode), which deleted characters.

When I finished the correction, I filed the document with Exit. I got an Illegal Command message. I checked the manual again. "E writes the edited text to disk and exits the editor." So I tried it again. After a few tries I gave up.

I soon realized I should not rely on the command summary; it is full of holes. The manual points out that Exit requires a new file name when the file edited is not new. I overlooked this.

My computer hung up a couple times when I used List to list lines on the screen. (Shift, down arrow, Obliterate was supposed to take me out of this, but it did not.) The Print instruction was strange. Without parameters it printed the next 20 lines.

I could not make the Obliterate command work. O <text> Break in Alter mode is supposed to delete all text up to <text>. My computer just hangs up.

To sum it up, this is a good editor with some very nice features, such as replacement of a string by another one in any range of line numbers. None of the instructions are compatible with the Level II Editor or with the Editor of the Radio Shack Editor/Assembler. And do not rely too much on the command summaries. Aside from this, the manual is readable, and some of the commands are illustrated with examples.

Macro Assembler

I could not make any sense out of the Macro Assembler manual. I could not understand the modes of symbols (absolute, data relative, program relative and common). I could not understand the related pseudo-ops ASEG, CSEG, DSEG and Common. But the worst part of the manual is easily the explanation of macros. Only one example is given to explain what a macro call does and even that uses 8080 opcode. In fact, all through the manual the scanty examples use 8080 pseudo-ops.

I learned that Radio Shack sells the Microsoft Macro Assembler under license. I asked my Radio Shack manager to get me this package so I could compare it with what I had.

The Radio Shack package contained two disks with the same four files and with the file FORLIB/REL thrown in for good measure. To my great disappointment, the manual was the same, except that mine had a disclaimer that it is not intended as a tutorial (they are not kidding); and a reference is given to a Zilog manual: *Z80-RIO, Relocating Assembler and Linker*

User's Manual. A call to Zilog assured me a copy of the manual.

The Zilog Manual

I read the Zilog manual. What a pleasure! It explained the Macro Assembler in less than 30 pages. They even explained how the Assembler works. Everything was illustrated with examples, and for just $7.50.

The manual explains four major points: the concept of relocatable modules; the mode of an expression (absolute, relocatable, external); conditional assembly; and macros. It does much more; it gives an excellent explanation of pseudo-ops, listing format, etc. But let us concentrate on the first four points.

When assembling a program with the Radio Shack Editor/Assembler, the assembly takes place from the location specified by the ORG (origin) statement. For instance, take the following simple example from the book:

```
00100          ORG   7000H
00110          LD    A,31H
00120          LD    (3E20H),A
00130   LOOP   JP    LOOP
00140          END
```

and in assembled form:

```
7E00              00100       ORG7E00H
7E00  3E31        00110       LD  A,31H
7E02  32203E      00120       LD  (3E20H),A
7E05  C3054A      00130LOOPJP  LOOP
7E00              00140       END
00000 TOTAL ERRORS
LOOP7E05
```

Thus LOOP stands for 7E00 plus five. This number five is the offset. If the ORG is changed to 8000H, Loop would be 8000 plus five.

A relocatable program typically has no ORG statement. The above program is assembled from 0000, and Loop has a value 5 and is marked with an R for relocatable. Similarly, each line is marked where change is needed when the origin becomes known.

Thus, assembly does not produce object code but relocatable code. It is the task of the Linker to take relocatable files and load them into core. Wherever the Linker finds a code marked as relocatable, the linker adds X to it, where X is the location from which the program is loaded. This X is either given by the user or generated by the Linker as it puts one relocatable segment after another.

This makes it easy to understand some of the rules of forming expressions. If Loop is relocatable, we cannot use 2*LOOP in an expression because adding X will not make this right. (We would have to add 2*X, and our Loader cannot do this.)

A label or a number can thus be relocatable or absolute. A label can also be external, meaning defined in another module. Such labels have to be defined in another module as global. For example, in module 1 we define: GLOBAL LOOP; then in module 2 we declare: EXTERNAL LOOP. Then module 2 can use the value for Loop (this is assigned to Loop when module 1 is loaded).

Conditional assembly takes the form:

```
COND   expression
.....
ENDC
```

If the expression evaluates to non-zero the code is assembled (up to ENDC) otherwise assembly continues with the first line after ENDC. This is useful when writing code that is to be applicable in various setups. Its real usefulness is, however, in macros.

To illustrate macros look at Program Listings 1 and 2.

Program Listing 1 is an Assembly-language subroutine for single-precision addition in a Basic program. It is immediately apparent that lines 210-240, 300-330, 500-530 are identical. This is exploited by the macro in Program Listing 2. The macro has a name: Load and three dummy variables: What, Into, From. When Load is called these three are specified, e.g., LOAD X,SA,HL, and the body of Load is copied into the code with What replaced by X, Into by SA, From by HL.

Conditional assembly can be illustrated by the following macro type:

```
MOVE   MACRO   X,Y,SAVE
       COND    'SAVE?' = 'S'
       PUSH    BC
       PUSH    DE
       PUSH    HL
       ENDC

       COND    'SAVE?' = 'S'
       POP     BC
       POP     DE
       POP     HL
       ENDC
```

If this macro is called MOVE AB,SUP,S, the registers are saved before the code is executed, while MOVE AB,5 will not save the registers. Since macros without a conditional can reproduce only one pattern, we gain much greater economy of coding by using macros with conditionals.

```
100   VARFTR   EQU    260DH
110   ADDSP    EQU    0716H
120   LDSAHL   EQU    09B1H
130   LDRAHL   EQU    09C2H
140   LDHLSA   EQU    09CBH
200            ORG    7E00H
210            LD     HL,VARX    ;LOOKING FOR X
220            CALL   VARPTR     ;DE POINTS AT X
230            EX     DE,HL      ;NOW HL POINTS AT X
240            CALL   LDSAHL     ;LOAD X INTO SA
300            LD     HL,VARY    ;LOOKING FOR Y
310            CALL   VARPTR     ;ADDRESS IN DE
320            EX     DE,HL      ;NOW HL POINTS AT Y
330            CALL   LDRAHL     ;Y INTO RA
400            CALL   ADDSP      ;CALL SINGLE PRECISION ADDITION
500            LD     HL,VARZ    ;LOOKING FOR Z
510            CALL   VARPTR     ;ADDRESS IN DE
520            EX     DE,HL      ;HL POINTS AT Z
530            CALL   LDHLSA     ;LOAD RESULT INTO VARIABLE TABLE
540            RET
600   VARX:    DEFM   'X'
610            DEFB   0
620   VARY:    DEFM   'Y'
630            DEFB   0
640   VARZ:    DEFM   'Z'
```

Program Listing 1

```
100            .REQUEST  LABELS
110            EXTERNAL VARPTR, ADDSP, LDSAHL, LDRAHL, LDHLSA
140   LOAD     MACRO   WHAT, INTO, FROM
150            LD      HL,VAR&WHAT
160            CALL    VARPTR
170            EX      DE,HL
180            CALL    LD&INTO&FROM
190            ENDM
200            ORG     7E00H
210            LOAD    X,SA,HL
300            LOAD    Y,RA,HL
400            CALL    ADDSP
500            LOAD    Z,HL,SA
540            RET
600            IRP WHAT,<X,Y,Z>
610   VAR&WHAT: DEFM   'WHAT'
620            DEFB    0
630            ENDM
640            END
```

Program Listing 2

The Microsoft Manuals: Another Try

I tackled the Microsoft manuals again, fortified by my understanding of the Zilog manual. It became immediately clear that this is not an implementation of the Zilog Assembler, but a reworked version of an 8080 assembler. Thus the terminology used is 8080 terminology with some Zilog terminology thrown in to confuse the user. For example, to understand the three modes of expressions is hard enough, but to understand Global is much more difficult since three names are used for this concept: Global, Entry and Public. To confuse matters worse, there are two kinds of relocatables: one for code and one for data. How these are handled is not explained until you get to the Link manual, and there you carefully read the fine print explaining the two switches, P and D.

Nevertheless, I proceeded as follows. First, I rewrote the manual for the Z-80 terminology:

where you find the pseudo-op	substitute
IFT	COND
ENDIF	ENDC
PAGE	*EJECT
DB	DEFB (if a single byte is defined)
DS	DEFS
DW	DEFW
DB	DEFM (if used for text)
SET	DEFL
PUBLIC	GLOBAL
EXT or EXTRN	EXTERNAL
the opcode	
MVI	LD
MOV	LD
INX	INC
JNZ	JR,NZ

Rewrite the few examples to avoid loading into ROM. (The typical example loads at zero of 800H.) Luckily, there are so few examples that the whole rewriting is finished in less than an hour.

In the Macro Assembler, the labels have five attributes (modes): absolute, code (program) relative, data relative, common and external. We can safely ignore common (this is included to interface with Fortran). The only difference between code relative and data relative is that the Linker will put the code relative segments one after another at the address specified to the Linker, and the data relative segments will come one after another at another address specified to the Linker.

The mode of the labels is determined by the segments they are in. A segment is declared absolute, code relative or data relative by the pseudo-ops ASEG, CSEG and DSEG, respectively. These pseudo-ops have no operands so they cannot be used to set where the segments will be loaded. Unfortunately, the default condition is CSEG and the default origin is zero.

Since the pseudo-ops are presented without example some experimentation is needed to find out how they work. I noted in my manual that the label in EQU and in Macro takes no colons (colons are compulsory for labels as a rule).

There are a number of pseudo-ops in the Macro Assembler that cannot be found in the Zilog manual. For instance, the Request file directs the Linker to the search file for external labels. We used this in Listing 2. This is very convenient; we can set up a file called Labels containing nothing but the definitions of often-used labels.

Conditionals have been expanded to eight formats and they also support Else statements. Having learned conditionals from the Zilog manual, I had no difficulty understanding these.

In addition to macros, the Macro Assembler has a number of useful block operations: REPT (repeat), IRP (indefinite repeat), IRPC (indefinite repeat character).

REPT is quite simple:

```
REPT 10
DEFS 5
ENDM
```

This simply generates DEFS 5 10 times. Instead of 10, we can also have expressions.

IRP is like a macro with a single dummy label; the code is re-

Expensive Expansion

EXPENSIVE – The *LNW System Expansion II* comes with a full 32K of 200ns RAM, and built-in RS232c 20 MA current loop serial interface. That's for starters. Next, consider our heavy gauge steel case, power indicator lamp, gold-plated connectors, FR-4 glass epoxy circuit board with solder mask and silk screen legends. Then there is the parallel printer port, screen printer port, real time clock, and extra heavy duty onboard power supply with over current protection, over voltage protection and thermal shutdown. If that's not enough then there is the floppy disk controller, guaranteed operation at a 4MHz CPU speed and our 6 month warranty. Every one of these features is *STANDARD*. This is true system expansion. You get every 'expensive' feature without spending more.

CHEAP – Our price is $399.95. Any way you compare, features or price, LNW's *System Expansion II* is the clear winner. The LNW System has been field tested for over two years with thousands of users. It works with any DOS, is 100% TRS-80 Model I compatible and it works 'right out of the box'. If there is any doubt in your mind as to whether you should buy ours or the 'other guys', just ask an LNW owner!

WE ARE #1 – Number one in price, features, reliability, performance and delivery. LNW is committed to 'expensive' features and quality at reasonable prices. LNW is committed to support, thorough documentation, and reliability.

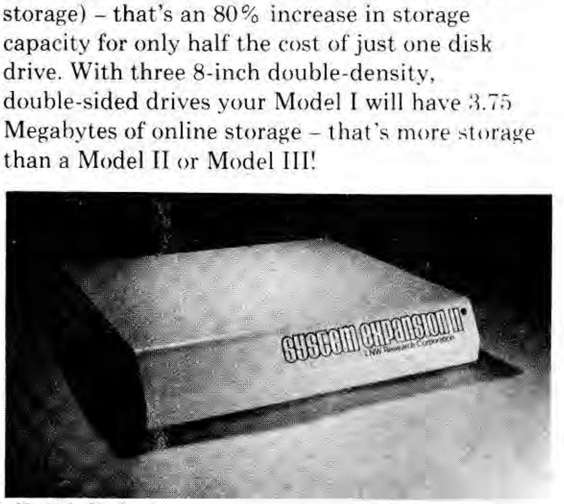
✓571

This has made us the number one manufacturer of system expansion units and accessories for the Model I computer.

EXPANSION OPTION – 8-inch drive capability is as easy as plugging in the *LNDoubler 5/8* option*. Now you can have any combination of single- or double-density, single- or double-sided, 8"* and/or 5" disks on-line! 8-inch disk storage increased to 591,360 bytes – 77-track single-sided, double-density or 1,182,720 bytes – 77-track double-density, double-sided.

The *LNDoubler's* unique 5/8 switch allows you to boot from 5- or 8-inch system disks and it's accessible from outside the interface. The $219.95 *LNDoubler 5/8* comes with a double-density disk operating system (DOS+ 3.3.9), *complete* with BASIC and utility programs... ready to run your software.

Each of your present 40-track, single-sided 5-inch drives will store up to 184,320 bytes (formatted storage) – that's an 80% increase in storage capacity for only half the cost of just one disk drive. With three 8-inch double-density, double-sided drives your Model I will have 3.75 Megabytes of online storage – that's more storage than a Model II or Model III!



LNW Research Corp.
2620 WALNUT Tustin, CA. 92680
(714) 641-8850 (714) 544-5744

*8" drive operation requires special cable, 8" double-density requires 3.55 MHz CPU speed-up modification or LNW-80 4 MHz computer.

peated as many times as necessary to go through the given list of parameters. This is illustrated in Program Listing 2. IRPC is similar but the list of parameters is replaced by a character string. Apparently, these can also call each other.

When I try out a new piece of software, I try to go in small steps. So to try out the Loader, I started with Program Listing 1 which has already been tested with the Radio Shack Editor/Assembler. I typed the program in under the name FIG3/MAC. There was a typing error, so I called it back again for correction. The second time around it could not be filed under FIG3/MAC, RENAME FIG3/MAC TO FIG3/MAC. I wish the Editor would give me a choice and let me retain the name.

I typed M80, to call the Macro Assembler. I typed =FIG3, which means the same as A/NO for the Radio Shack Editor/Assembler; it assembles FIG3/MAC (the default expression) but creates no code. M80 responds: No Fatal Error(s). I would prefer not to see the word fatal if there is no mistake. Encouraged, I typed FIG3,FIG3 =FIG3. This means: assemble FIG3/MAC, create a file FIG3/REL (the relocatable code), and another one, FIG3/LST, for the listing. I exited M80 with Break (this works about 50 percent of the time) and typed L80 to call the Linker. I typed FIG3 (the default is REL) and it loaded. But when I used Debug to see how it loaded I got a big zilch. I quit to reread the Linker manual.

The explanation was simple. The ORG statement does not override the default in the Macro Assembler and Linker: CREL, code relative. Since FIG3 contained no ASEG, despite ORG 7E00H the segment was assembled as code relative. Thus, as a default, a program will always load into ROM. I have seen many stupid defaults, but this must take the prize.

I added ASEG to FIG3, went through the same procedure, and it worked. I got more ambitious and tried out Program Listing 2, using the file name FIG4. This contains three important features of the Macro Assembler. I added ASEG to the code. It ran with no difficulty.

Rereading the Loader manual, I realized that everything that was left open by the procedure of building five kinds of relocatable modules has to be resolved by instructions to the Loader. Typing -P:address1,-D:address2 sets the starting address of the program (so the CSEG segments load from here) and of the data area (for the DSEG segments). By default the numbers you specify are decimals. Type -H, to make the addresses hexadecimal. So to create an object file for FIG4 we type: -H,-P:7E00,FIG4-N,-E. -N (N for name) creates an object file called FIG4/CMD (this is slightly misleading, CIM would be more appropriate), and -E exits to the operating system.

Summary

This is definitely not a friendly system. Each module has the same prompt (an asterisk) and there are four different exit procedures. Numbers are thrown at you without explanation, e.g., when running L80, and command lines take the place of prompting questions. Nevertheless, the system is simple enough to learn given the proper instructions.

A number of important features are missing: There are no local labels; macros cannot be called from another file; there is no library manager; and no runtime system is provided that would facilitate overlaying.

Compared with using a nonmacro assembler, however, working with this assembler is like driving a Cadillac. The software itself works smoothly.

To make this package acceptable, some improvements would have to be made. But even as it is, the Macro Assembler seems indispensible for anyone wanting to write Assembly-language programs with the ease only macros can provide. ■

"DOS RANDOM ACCESS & BASIC FILE HANDLING"

the HOW TO, SHOW & TELL, STEP by STEP book. Written for TRS-80 and all computers using Microsoft basic.
The breakthrough book for non-programmers... businessmen and professionals who want to solve in house problems by writing your own customized program...and the hobbyist with the desire to get beyond the cassette and into disk data storage and file manipulation. **"DOS Random Access & File Handling"** is the plain speaking, easy to read 150 page book for learning **HOW TO WRITE SPECIAL PROGRAMS** for inventories, mailing lists, work schedules, record keeping and more. Your acquaintance with Level I and Level II basic and this self-instruction tutorial will assure quick progress.

DSC Publishing
2 Dogwood Drive
PO Box 769
Danbury, CT 06810
Phone Order
(203) 748-3231

Yes, I want to expand my programming knowledge.

Please send me a copy of **"DOS Random Access & File Handling"** for 24.50 + 1.50 shipping ☐
Also send me the compatible program disk for 28.50 + 1.50 shipping. ☐
Overseas add 8.00 per book and disk airmail.

Name ____________
Address ____________
City ____________
State ____________ Zip ____________
Master Charge ☐ VISA ☐
Expiration Date ____________
Card # ____________
Check ☐ Money Order ☐
COD (2.00 extra) ☐
Signature ____________
Phone Orders (203) 748-3231
CT. orders add 7.5% sales tax.

IS THERE ANY OTHER SOFTWARE THAT ENABLES YOU TO:

Enhance DOS commands to rename, kill, copy,... multiple files with common name or extension?
Link EDTASM source files via new %INCLUDE command:
Assign and use symbolic labels in DEBUG?
Compile, link and execute FORTRAN program(s) by entering just one command (system will prompt you to insert proper diskette if necessary)?
Organize conditional chained execution?
Save hours of typing and have a good time with your computer?
These are but a few examples of what can be done with the

INTERACTIVE CONTROL LANGUAGE

ICL is a very fast machine-code program executing procedures written in the high level command language. (Variables and conditionals, &READ &IF &QUEUE &GOTO &JUMP &RESUME... 18 commands plus special features and number of options.) Under ICL procedures are invoked the same way as /CMD files or DOS commands.

ICL is designed to work as a "middleman" between the user and operating system and/or application programs. It has powerful facilities for control of program execution and data manipulation. ICL introduces virtual I/O for keyboard and display operations and provides uniform interface between different otherwise incompatible programs. ICL implements control interception scheme for interactive co-execution of programs and procedures, which enables enhancement and customization of existing software products in a simple way.

Superb documentation with detailed "how-to-use" and tutorial information, most useful procedures, system information and patches are provided.

This unique product is now available for NEWDOS2.1 or NEWDOS/80 — on 5" diskette + documentation at unbelievably low price of $34.95 (check or money order)

366 **XYZT COMPUTER DIMENSIONS INC.**
2 PENN PLAZA, suite 1500. New York, N. Y. 10121.

* Requires min 32K 1 Drive Model 1 (48K for FORTRAN).
** Specify NEWDOS2.1 or NEWDOS/80 diskette format (works with either DOS).
*** N.Y. state residents add 5% sales tax.

PRODUCTS FOR YOUR RADIO SHACK

COLOR COMPUTER

NEW!

MICROTEXT COMMUNICATIONS VIA YOUR MODEM!

Now you **can** use your printer with your modem! Your computer can be an intelligent printing terminal. Talk to timeshare services or to other personal computers; print simultaneously through a second printer port; and re-display text stored in memory. Download text to Basic programs; dump to a cassette tape, or printer, or both. Microtext can be used with any printer or no printer at all. It features user-configurable duplex/parity for special applications, and can send any ASCII character. You'll find many uses for this general purpose module! Available in ROMPACK, ready-to-use, for **$59.95.**

SOFTWARE DEVELOPMENT SYSTEM

The Micro Works Software Development System (SDS80C) is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack! Vastly superior to RAM-based assemblers/editors, the SDS80C is non-volatile, meaning that if your application program bombs, it can't destroy your editor/assembler. Plus it leaves almost all of 16K or 32K RAM free for **your** program. Since all three programs, editor, assembler and monitor are co-resident, we eliminate tedious program loading when going back and forth from editing to assembly and debugging!

The powerful screen-oriented Editor features finds, changes, moves, copys and much more. All keys have convenient auto repeat (typamatic), and since no line numbers are required, the full width of the screen may be used to generate well commented code.

The Assembler features **all** of the following: complete 6809 instruction set; complete 6800 set supported for cross-assembly; conditional assembly; local labels; assembly to cassette tape or to memory; listing to screen or printer; and mnemonic error codes instead of numbers.

The versatile ABUG monitor is a compact version of CBUG, tailored for debugging programs generated by the Assembler and Editor. It features examine/change of memory or registers, cassette load and save, breakpoints and more. **SDS80C Price: $89.95**

GAMES

Star Blaster — Blast your way through an asteroid field in this action-packed Hi-Res graphics game! Available in ROMPACK; requires 16K. **Price: $39.95**
Pac Attack — Try your hand at this challenging game by Computerware, with fantastic graphics, sound and action! Cassette requires 16K. **Price: $24.95**
Berserk — Have fun zapping robots with this Hi-Res game by Mark Data Products. Cassette requires 16K. **Price: $24.95**
Adventure — *Black Sanctum* and *Calixto Island* by Mark Data Products. Each cassette requires 16K. **Price: $19.95** each.

ROMLESS PAK I — is an empty program pack capable of holding two 2716 or 2732 EPROMs, allowing you up to 8K of program! The PC board inside comes with sockets installed, ready to go with the addition of your custom EPROMs. **Price: $24.95**
2-PASS DISASSEMBLER — with documentation package. 16K; cassette. 80C Disassembler **Price: $49.95**
CBUG — Machine language monitor. CBUG Cassette **Price: $29.95**
CBUG ON 2716 EPROM: Can plug into Romless Pak I. CBUG ROM **Price: $39.95**
PARALLEL PRINTER INTERFACE — serial to parallel converter allows use of all standard parallel printers. PI80C **Price: $69.95**
Assembly Language Programming, by Lance Leventhal. **Price: $16.95**
MEMORY UPGRADE KITS: 4-16K Kit **Price $39.95**. 16-32K (requires soldering experience) **Price: $39.95**
PARTS & SERVICES: SAMs, 6809Es, RAMs, PIAs. Call for prices.

WE SHIP FROM STOCK!

THE MICRO WORKS 109

GOOD STUFF!

Master Charge/Visa and COD Accepted

P.O. BOX 1110 DEL MAR, CA 92014 714-942-2400

See List of Advertisers on page 290

THE PROGRAM STORE

TIME DUNGEON

By Howard Berenbon from AOS
A delightful fantasy game with an educational slant. Explore the graphic dungeon, collecting gold and meeting alien travelers. When you enter a "Time Portal," you are transported to a special date in history. Should you answer the question posed you'll be rewarded in gold; a wrong answer costs you treasure. The questions may be true/false, multiple choice, or short answer, and they're not easy!

TIME DUNGEON - U.S. 16K tape $24.95
32K disk $29.95
TIME DUNGEON - WORLD 16K tape $24.95
32K disk $29.95

LANGUAGE TEACHER

By Cindy and Andrew Bartorillo
Learn the basics of a foreign language. LANGUAGE TEACHER offers hundreds of word combinations, verb conjugations and phrases. There is an option for having multiple-choice answers and for being retested on missed items. Full printer capability and a great deal of "human engineering" further enhance the programs. Teachers will appreciate the ample documentation and the ability to get printouts of quizzes. Currently available languages are French, Italian, German I & II, and Spanish I & II.

$29.95 for each program on 32K disks

SPEAK!

By William Neville III
Incredible new voice digitization/generation program that needs no voice synthesizer. You use your cassette recorder's microphone to input words or phrases. SPEAK! digitizes them and produces a disk file for later use in your own programs. The speech generated is sent out through the cassette port to any amplifier. Create your own talking games and voice prompted programs! While a model III is required for speech input, the resulting voice output will work with models I or III. You have to hear it to believe it!

48K disk...$39.95

TRS-80 Unless Otherwise Noted

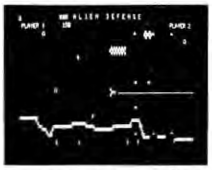
NOW FOR MODELS I OR III!

-80 SPACE RAIDERS

From Bosen Electronics
You are in command of the Starship "Defiant." The center of the screen is your "window" to the vastness of three dimensional space. Above and below it are readouts of critical information. Your orders are simple enough: Patrol the area and destroy all enemy spacecraft; return to base as needed for repairs and supplies. Carrying out these orders is more difficult!

An exciting and fast-paced game, -80 Star Raiders presents a flicker-free, animated view of the action from the pilot's perspective. Remarkably realistic.

16K tape...$24.95



ALIEN DEFENSE

By Larry Ashmun from Soft Sector
Piloting your ship across the horizontally moving terrain, you must battle the various enemy spacecraft. You are under attack almost constantly from missiles and bombs, and to make matters worse, your ground patrol people are being picked up by the alien landers. To save them, you must shoot the landers and swoop down to "catch" the falling man. This fast action game requires skill and rapid reflexes. The model III version makes excellent use of that model's special graphic features.

Model I, Tape: $19.95 Disk: $24.95
Model III, Tape: $19.95 Disk: $24.95

JABBERTALKY

From Mind Toys
A "programmable word game" that lets you compete against time or friends in solving your choice of anagram sentences or cryptograms. Skill levels range from simple to simply incredible! The sentences are generated by JABBERTALKY's "Language language," using the built-in vocabulary or your own.

16K tape...$29.95 32K disk...$29.95

OmniTerm

From Linbergh
An extraordinarily comprehensive smart terminal program for data communications. You can set all data transmission and reception parameters, define both input and output translation tables, and make all adjustments necessary to enable your TRS-80 to "talk" to just about any type of communications computer. Features include auto sign-on, prompted data transmission (a test editor is included), error checking & tabulation, autodialing (if your modem is so equipped), communication status checking, and much more. Also included are programs for preparing machine language for transmission, decoding received machine language programs, and special set-up tables for Compuserve, Source, and Dow Jones News Service. Complete with thorough documentation. Requires RS-232, modem.

32K disk...$99.95

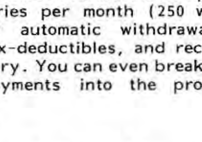
Also Available:
ATERM 1.4 — a less complicated smart terminal program...$19.95

By Chuck Acree from Acorn
A comprehensive genealogical program. It quickly and easily sets up a data base that holds name, date and place of birth, marriage and death information, plus a comment line for each ancestor.

YOUR FAMILY TREE will display/print a complete "pedigree" for any family member; a 3-generation chart may be displayed/printed showing the number of known ancestors beyond each branch of the tree. The program will also display a U.S. outline map showing migration across the country. You get full search capabilities on any key field. Capacities: 16K tape: 45-55 ancestors. 32K tape: 175, disk: 100. 48K tape: 300, disk: 225.

16K Tape or Disk...$29.95

MONEY MANAGER



By Andrew P. Bartorillo from Acorn
A complete management tool for the home budget, it accurately keeps track of your checkbook and provides an easy method of budget allocation. You can store information on up to 100 checkbook entries per month (250 with 48K), specify any automatic withdrawals, keep records of tax-deductibles, and record expenses by category. You can even break up charge account payments into the proper categories.

32K disk...$39.95

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

THE PROGRAM STORE
4200 Wisconsin Avenue NW, Dept. 8E02 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

✓17

Now you can run your Z80 at 2.5 MHz.

The SK-2 Clock Modification Kit

SK-2
Mumford Micro Systems
Summerland, CA
Model I
$24.95

C. H. Ballard
Box 824
Monticello, MN 55362

Many personal computer owners are disappointed with their machine's lack of speed. The TRS-80 has taken quite a bit of flak in this regard because the Z80 microprocessor does not run at its best clock speed of 2.5 Megahertz.

Some people substitute a new, higher-frequency oscillator crystal for the original crystal on the CPU. But you are stuck at the new higher speed forever, leading to a host of software problems (unreadable tapes and disk load errors).

Choosing A Clock Modification Kit

Before you attempt any modification, be aware that you will void Radio Shack's 90-day warrantee by opening the case. Also, do not attempt the mod yourself unless you are confident with your abilities.

In my search for a clock modification, I established the following criteria:

- The modification should require a minimum of modifications to the original equipment circuits. This makes it easier to restore the circuit to its original configuration if the need arises.
- The speed option should be switchable between low and high speeds to minimize the impact on software. Mechanical switching should enable the user to determine what speed the machine comes up in when the power is turned on. Software switching through an output instruction would be a nice feature.
- Disk system interaction should be accounted for: A hardware option should slow the system back down when disk accesses are made.
- The modification should indicate what speed the computer is running in at all times.
- The modification should cost as little as possible to operate.

With these criteria in mind, I began my search. My search gradually narrowed in on the SK-2 clock modification produced by Mumford Micro Systems. Before buying, I wrote for information on installation difficulty and was rewarded by return mail with a complete copy of the actual installation instructions.

First, decide which options you intend to implement on the SK-2. After removing the six screws holding the keyboard unit together, determine where to mount the SK-2 circuit board. I have Level II ROMs and Radio Shack's XRX-2 cassette modification board; only a small relocation of the Level II ROMs was necessary to mount the SK-2 board.

The options list of the SK-2 takes almost two pages. First, you have the choice of optional speeds. In its standard configuration, the SK-2 allows a 50 percent increase or a 50 percent decrease in selectable clock speed. (A decreased clock speed is useful for slowing down listings, debugging and for cutting down on keybounce problems during long input sessions.) You can have only one speed as an option. The 100 percent speed increase option requires some rework of the SK-2 board and probable replacement of certain circuits on the CPU (switching to a Z80A(B) and faster 4116 memories) that can be substituted in place of the 50 percent speed reduction.

I believe the 100 percent speed up should be offered as an SK-3 kit, with a new board not requiring rework and including a Z80A to complete the package. Implementing that option with the present kit is somewhat tedious.

Having chosen which speeds to include, you must make them available to the user. You can install a simple single-pole switch to select normal speed or an optional speed. If you want two optional speeds, you need another single-pole, double-throw switch. For software control alone, you can wire the SK-2 to accept an OUT 255,32 to change to the optional speed, or an OUT 255,0 to return to normal speed. The software option requires two more connections to the CPU.

As a grand-slam, you can choose the "Combination Option." This scheme requires replacing the single-pole, optional-speed switch with a single-pole, double-throw switch with a center-off position. In one on position, the CPU is at normal speed at all times. In the center-off position, the clock runs at the optional speed. When in the remaining on position, the speed is software selectable. I chose the combination option as well as two switch-selectable optional speeds (50 percent faster and 50 percent slower). This required mounting two small toggle switches (the type often used on computer front panels for address switches) on the front of the keyboard case bottom. When mounted in the case there was about ¼ inch of clearance between the switch terminals and the circuit board. I used a hot soldering iron to melt a hole through the case where I wanted the switch and then trimmed the still-warm plastic with an X-acto knife to remove burrs and established the final hole dimension. This method is less stressful on the circuit assembly than drilling on the case.

A disk-based system should automatically return to normal speed when any disk I/O occurs. This requires connecting the SK-2 to the motor-on signal from the expansion interface. The instructions suggest running a wire with a connector in the middle to the expansion interface. I felt it looked more professional to use two connectors—one on the keyboard and one on the interface. A connecting cable with mating connectors could be used rather than a loose wire dangling from the keyboard and expansion interface.

Finally, if you would like an indication that the CPU is not running at normal speed, consider

the LED option. You must supply and mount a small, low-power LED (the current limitation is not specified) and run leads to two points on the SK-2 circuit board. The LED lights any time the CPU is not running at normal speed. If you have more than one optional speed, I suggest labeling the switch positions.

Actual installation is relatively simple, although I would recommend you have a copy of the *Technical Reference Handbook* (Radio Shack Catalog No. 26-2103) for the TRS-80 when you do the installation. All options require that one trace be cut and four wires be soldered to four points on the CPU. If you want software-selectable speeds, two additional wires must be soldered onto the CPU. An excellent figure is provided for the first four connections and the trace cut.

The same cannot be said for the two software selection option connections; a clear figure would simplify the process both here and in the Expansion Interface option.

Using the SK-2

With my fingers crossed, I turned on the power for the infamous smoke test. No curly white wisps ascended from the case vents and the CRT lit up with the memory size question. (I left the Expansion Interface disconnected in case of trouble.) I entered a small test program from the instructions. I could get the printout to speed up and slow down by flipping the switches correctly. Once in a great while the CPU hung up when I switched from normal to high speed. The problem never occurred when switching from normal to low speed or when the speed change was software selected, so it may be a contact bounce problem.

My first real problem came when I attached the Expansion Interface to the keyboard; I could not run reliably at high speed. While frantically re-reading the instructions I found a note on page four of the instructions for owners of the new Expansion Interface. The cable from the keyboard to the new interface is not buffered (there is no black box in the middle of the cable). A restrictive hardware timing arrangement in the interface must be undone before the memory in the interface will run at the higher speed. Important information like this should be earlier in the instructions, tell you how to identify the problem interfaces, and help you locate the circuits to be modified with illustrations. In my case the circuit board was laid out so I could locate the circuits by counting chips; the *Expansion Interface Service Manual* (Catalog No. 26-1140) was also a help. The modification instructions were badly convoluted; they should have been step-by-step.

Once I had made the changes to the Expansion Interface, I no longer had a problem.

User Experience

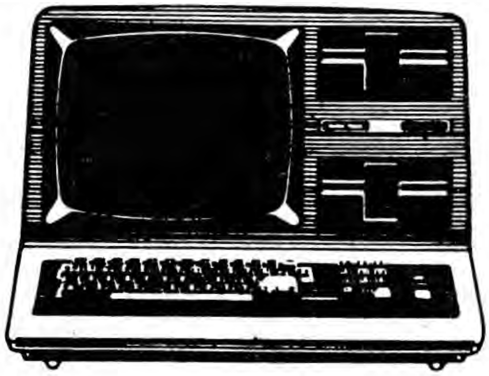
I have been satisfied with the performance of the SK-2. I now use software corrections to allow disk operation at higher speeds. I obtained these corrections from Mumford; by now they may be part of the standard instructions. The author of these corrections has also published them in an April 1981, *80 Microcomputing* article.

I am still able to occasionally hang the system during mechanical switching over to high speed, so for critical applications I use software switching. The LED option prevents grief when I forget to switch out of high speed and try to load an old cassette.

The Mumford Micro Systems SK-2 clock modification lives up to all its advertising. The company is responsive to questions and inquiries, both before and after the sale and orders are rapidly processed.

Consider the SK-2 as a cost-effective clock modification alternative that can be implemented in the software-controlled version.

If my experience with this modification could be of any use to you, feel free to write me at the address above or contact me through EMAIL on CompuServe at 70225,332.■



Tie 'em all together for $100!

Networking on a Shoestring

Donald R. Meinke
2507 Blaine SE
Grand Rapids, MI 49507

One method of stringing several TRS-80's together is to buy Radio Shack's Network I Controller. The controller costs $499. Since it basically contains an amplifier and a switching circuit, I built a system to do the same thing that costs only about $100.

As a teacher of physics and computer math at Creston High school in Grand Rapids, Michigan, I have 10 TRS-80s that I have networked together. The master computer (Photo 1) has an expansion interface with 48K of memory and a disk drive. I have run the master computer using only 16K of memory and Exatron's Stringy Floppy for mass storage.

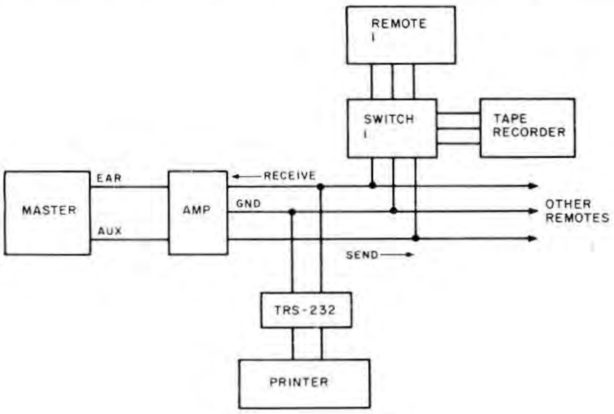


Fig. 1. Block Diagram of System

The system scheme is shown in Fig. 1. The ear and auxillary plugs for output to the tape recorder are plugged into a stereo amplifier. One channel of the amplifier is used to send signals to the remote computers and the other channel is used to receive signals from the remote computers. A three-wire conductor runs from the amplifier to the remote units. There is a common ground for the send and receive lines and a parallel circuit coming off the receive line that runs to a Small Systems Software TRS-232 interface which drives the printer. Any printer having an RS-232 or 20 ma interface can be used in this system. I am driving a Decwriter LA-36.

At each remote unit there is a switch that allows the student to control several options (see Photo 2 and Fig. 2).

Option 1

Option 1 allows the student to receive any program being sent by the master computer. The student must have his switch in the receive position and must also have typed and entered the command Load. I am using Level II Basic from Microsoft, so Load is a legal command. CLOAD would be used with Level III Basic. On the master computer SAVE "name" or CSAVE "name" would be entered. Of course the program to be sent should previously have been loaded into the master computer.

Pushing the Enter key on the master computer will send the

Photo 1. Master Computer: Model I 48K with single disk drive and Stringy Floppy.

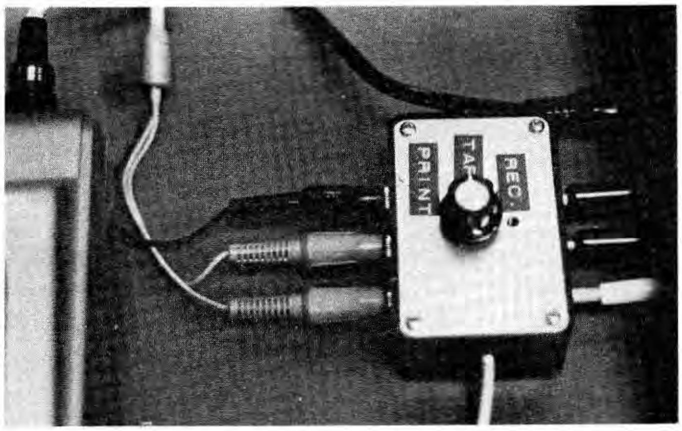


Photo 2. Switch: REC—Receive from master; TAPE—Tape recorder only; PRINT—Send to printer or master.

program out the auxilliary jack on the tape recorder interface. The signal enters the amplifier where it is boosted to the necessary level to be picked up by as many remote computers as are requesting that program. i find I need the sending channel amplifier volume control turned up halfway to send to all computers at once. Once the volume control is set, it needs no further adjustment. I have read that the volume multiplication factor needs to be 10 to 1.

Each remote computer will receive the program as if it were coming off the tape recorder. The asterisks will blink normally and the baud rate will be 500. In effect we are merely bypassing the tape recorder while sending a program, greatly increasing the reliability of the load. If the master computer successfully loaded the program, then the remote will also have a successful load. The only exception to this is if the master is using a DOS and the clock is not turned off through the CMD"T" function.

Option 2

To send a program back to the master computer the student must have his switch in the send mode. Here the role of the master and remote are reversed. LOAD or CLOAD has been typed on the master and the Enter key pushed. SAVE"name" or CSAVE"name" has been typed and the enter key pushed on the remote. The signal travels from the remote to the second channel on the amplifier. The volume control on this channel is also turned up one half. When the program has loaded on the master it is saved on disk or stringy floppy.

This system allows students working on projects in physics to send me their programs, and obviates the need for cassette tape.

Option 3

This option is very similar to Option 2. The student switch setting is the same, but the printer is turned on. Since the printer parallels the receive line for the master computer, the signal also passes through the TRS-232 interface. Now the student is able to print whatever is on his computer.

Our set-up is such that the student can LLIST his program to the printer or, using a special software command, List and run his program on the printer. The output will go simultaneously to the screen of the remote computer and the printer, so the student will be able to see exactly what is printing. The master computer printer also has a TRS-232 interface. It would be possible to have both printers Listing a program at the same time.

Option 4

One remote unit may send a program to one or more remote units by bypassing the master computer and routing the signal from the receive line into the amplifier and back out the send line. Any remote in the receive position with the Load command entered would receive the program.

Option 5

The computer math classes use tape for mass storage. The remote switches may select tape recorder only. Tape recorder use is completely cut off unless this switch selection is made. This is to prevent someone from leaving their switch in the send or receive position and then trying to Load or Save with tape. I don't want signals on the line that don't belong there.

Why Network TRS-80s?

The idea to network computers did not originate with the need to send, receive or print student programs. When I switched from a time-share system with two terminals to a system with 10 micros I created a problem for myself. Each of the nine remote micros has two resident programs in RAM; Level III Basic and a printer driver.

Level III offers most of the advantages of DOS except DOS input and output. There are several other advantages of Level III; two of these are important to my setup. Level III offers keyboard debounce and less sensitive cassette I/O, which has saved me a great deal of hassle with the old style keyboards on the Model I. Level III resides in low RAM.

The printer driver for the TRS-232, called Formatter (from Small Systems Software), allows great versatility in printing. This program resides in high RAM. Both these programs are in machine code which makes them very sensitive to the tape recorder volume setting. Each morning I would have to load nine micros with both these programs from nine different tape recorders. This was a time and load reliability problem. To avoid this I left all the computers on overnight which made me uneasy. With the network system I can load all the remote computers in 10 minutes by myself and in about five minutes with my lab assistant's help.

With machine code programs the System command must be used to load them. To send these programs from the master computer I load them from disk along with a third program called RSM (also from Small Systems Software). RSM is a monitor program that has a Punch command similar to the Punch command used by T-Bug. Once this is loaded into the master computer, the System

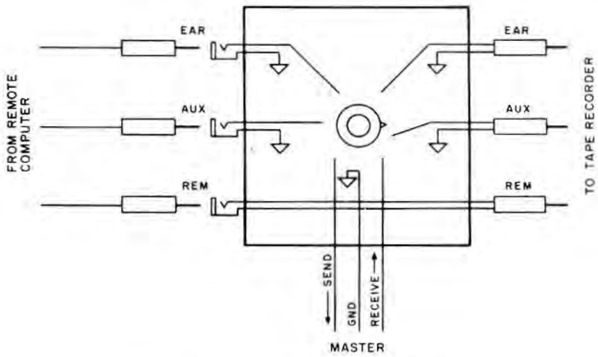


Fig. 2. Switch Box Wiring Diagram

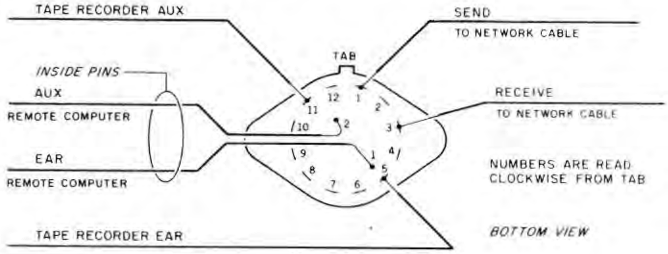


Fig. 3. Switch Wiring Diagram

Photo 3. Front of amplifier Selector to tape/tuner: Mode—Stereo; Tone—Fully counter-clockwise; Left—Receive channel; Right—Send channel.

command can be entered on all the remotes. The name given in response to the *? query can be anything; I usually use a single letter. Once all nine remotes are ready, each resident program is sent out by the Punch command with the beginning and ending memory addresses. This technique can be used with any machine code.

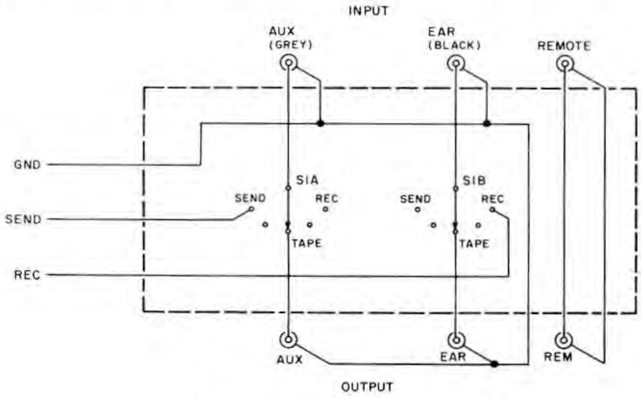


Fig. 4. Switch Schematic

Photo 4. Remote Micros

Wiring the Switch

Wiring the switches takes the most time in this project. You can eliminate the switches from the system and solder phono plugs into the line, so you can plug the ear and auxilliary plugs directly into the line from each remote. I originally designed the system this way, but it puts a lot of strain on the cassette cable. The students had to plug and unplug the cables to go back and forth between the recorder and the receive and send modes. It was also a lot of hassle to load the computers every morning by this method.

To wire the switch, refer to the schematic in Fig. 4 and the wiring diagrams in Fig. 2 and Fig. 3. I used a two-pole, six position switch. This many connectors are not needed but this switch was readily available at Radio Shack (catalog # 275-1386). When wiring the switch, remember to have a common ground for the ear, auxiliary and three-wire conductor from the master computer. A simple jumper wire between the ground connections on the ear and auxiliary jacks will suffice. Once the ground jumper is in, solder the send-receive ground to it.

2 miniature phono plugs
2 miniature phono jacks
1 subminiature phono plug
1 subminiature phono jack
1 rotary switch, 2 pole 6 position
1 component box

Fig. 5. Parts list

System Cost

The major cost of this network system comes from the purchase of a stereo amplifier, Radio Shack catalog #31-1982A, ($30). The other major cost is the remote switches, again purchased from Radio Shack ($58). I put the switches together myself so there was no labor charge. The last charge came from the purchase of two rolls of 75 foot three-conductor wire. One roll was used for the network and the other as hook-up wire for the switches.

Time Share?

It would be convenient if the system could use software for all the options mentioned above and also if the baud rate could be increased so programs could be saved on disk, eliminating tape completely. This would bring the system very close to being a timesharing one—not a timesharing system in the sense of using time from the master computer, but in being able to save and load rapidly to the master. This system would be helpful not only in the classroom, but anywhere a number of micros need to be combined. ■

SPM STATISTICAL PACKAGE FOR MICROCOMPUTERS

Med Systems is proud to announce SPM, the Statistical Package for Microcomputers. Written by Bruce Powel Douglass (see below), this is an integrated package of nine different programs providing a wide range of statistical analyses and data manipulations.

Each of the seven menu-driven SPM analysis programs can be run as a separate, self-contained analysis program. Each allows data entry and editing, with file size limited only by the computer's memory capacity. Files created by each program are optimized for speed in that procedure.

The disk version of SPM has two additional data handling programs which make it an integrated statistical package. In this version, files for one procedure can be transformed into files for any other procedures using FILETRAN. Previously entered files may be concatenated or new variables added using the EDIT program. Files can be merged and new variables created from functions of other variables. EDIT also provides 17 different data transformations, including linear, logarithmic, and exponentiation transforms, absolute values, and trigonometric functions. Transforms can be nested 30 deep.

The seven analysis programs in SPM provide a wide range of statistical analyses, from simple descriptive statistics to complex nonlinear regressions. Hard copy output is available from all programs.

The individual programs can be run in 16K. Tape versions are sold separately, see below. Tape versions save data to tape, disk versions to disk. WE HIGHLY RECOMMEND ALL DISK USERS TO BUY THE DISK VERSION. Since FILETRAN and EDIT are only available on disk, full use of SPM's many features can simply not be made using tape.

SPM offers features usually found only in mainframe statistical packages. In fact, it offers some not found in mainframe statistical packages! SPM is incredibly flexible, allowing transformations and combinations of variables to be created and analyzed. SPM is one of the finest professional microcomputer products ever offered.

ABOUT THE AUTHOR . . . Bruce Powel Douglass is a familiar name to most readers of '80 Microcomputing and other computing magazines. His software reviews, letters, and articles appear often, usually in regard to the scientific and statistical applications of microcomputers. He will soon have his own monthly column in '80 Microcomputing. Mr. Douglass is a candidate for a Ph.D. degree in neurophysiology at the University of South Dakota.

SPM (Seven Analysis Programs) TRS-80 Cassette: $74.95 (All nine programs) TRS-80 Diskette: $89.95

. . . and for the kids . . .

THE PLAYFUL PROFESSOR is a mathematical tutorial that provides instruction in addition, subtraction, multiplication, and division, with or without fractions. The program places the user(s) in a 60 room mansion haunted by an intelligent ghost who holds the key to the only exit. Correct problem answers are rewarded with moves through the castle. Incorrect answers prompt a full screen tutorial in a step-by-step, blackboard format that is easily understood by most children. Options include 3 difficulty levels, choice of problem type, one or two players, and split difficulty levels to allow parent and child to compete equally. A "Pass" feature allows input of a password to bypass the problems and leave only the game. Graphics are fast and extensive. All aspects of solving, including finding common denominators, reducing to lowest terms, and graphic long division are included.

The Playful Professor was judged one of the Top Ten Educational Programs by Robert Jackson, Mead School, as published in Microcomputers in Education, Vol. 1 No. 10. He says "every home with children and a computer should own this program. At the Mead School we use it daily."

"My 7 year old daughter had a fun way to review her math this summer, and my 4 year old daughter enjoys playing using the "pass" feature (what an experience it is to watch a 4 year old use a computer!)." S.C., Highland, Md.

The Playful Professor
TRS-80 Cassette: $14.95 TRS-80 Diskette: $17.95

MED SYSTEMS SOFTWARE

P.O. BOX 2674 CHAPEL HILL, NC 27514
TO ORDER, CALL: 1-800-334-5470

128

The Key Box
Basic Level II
Model I
32K RAM
TRSDOS

Eliza and Micromusic board a disk.

Two Transfers Please

Bettye Hollins
1525 Farnell Court
Decatur, GA 30033

Previous articles in *80 Microcomputing* illustrated procedures for transferring some of Radio Shack's Level II system tapes to disk. The utilities given by Irwin Rappaport in "Get T-Bug High," January 1980; "Relocate with PEEKPOKE," February 1980 and Robert Butler's "Edtasm on Disk," February 1980 make it possible to use Edtasm, T-Bug and Microchess 1.5 with the ease and efficiency of disk input. I had no success wishing Eliza and Micromusic to disk so I tried the procedures on these two system tapes.

The heart of the procedure is the simple, powerful block move but its application requires the starting, ending and entry addresses of the program you wish to transfer. To determine this requires a little work, but unless you love cassette input, it is worth the effort.

I examine memory before and after loading the program using T-BugHi which I use for addresses lower than 7300H. This is where T-BugHi resides. First, get into Level II Basic and press the Reset button to prevent the loss of Level II while moving back into DOS. Load T-BugHi and note the contents of several memory locations starting at 4000H. Level II Basic extends to 51C8H. The result is a disk version of Eliza and Micromusic.

To transfer Eliza to disk I used the procedure given by Robert

	Hex	Dec.	TRS-80 Dec.
Start Peeking	8300	33533	−32003
Stop Peeking	89FF	35318	−30218
Org. High Order Start	43	67	67
Org. High Order End	49	73	73
New High Order Start	83	131	131

Table 1

```
8100           00100          ORG     8100H
8100 210043    00110          LD      HL,4300H        ;SOURCE ADDRESS
8103 110083    00120          LD      DE,8300H        ;DEST. ADDRESS
8106 010007    00130          LD      BC,0700H        ;# OF BYTES TO MOVE
8109 EDB0      00140          LDIR                    ;MOVE IT
810B C3191A    00150          JP      1A19H           ;RETURN TO BASIC2
0000           00160          END
00000 TOTAL ERRORS
```

Program Listing 1

```
8A00           00100          ORG     8A00H
8A00 210083    00110          LD      HL,8300H        ;SOURCE ADDRESS
8A03 110043    00120          LD      DE,4300H        ;DEST.ADDRESS
8A06 010007    00130          LD      BC,0700H        ;# OF BYTES TO MOVE
8A09 EDB0      00140          LDIR                    ;MOVE IT
8A0B C30043    00150          JP      4300H           ;JUMP TO START
0000           00160          END
00000 TOTAL ERRORS
```

Program Listing 2

Starting Address 5000H
Ending Address 7586H

Org. 7FF0(32727)		Org. A800H	
Source Address	5000H	Source Address	8000H
Dest. Address	8000H	Dest. Address	5000H
No. of bytes	358F	No. of bytes	358F
Block Move #1		Block Move #2	

Save program using tapedisk

Starting Address 8000H
Ending Address A80FH
Transfer Address A800H

Fig. 1

```
88B9           00100          ORG     88B9H
7E88           00110 MSG      EQU     7E88H
7E00           00120 DUMP     EQU     7E00H
7E68           00130 DCB      EQU     7E68H
88B9 21887E    00140 SETUP    LD      HL,MSG          ;PUT QUERY IN HL
88BC CD6744    00150          CALL    4467H           ;DISPLAY ON SCREEN
88BF 21687E    00160          LD      HL,DCB          ;GET DCB ADDRESS
88C2 0620      00170          LD      B,32            ;LENGTH OF DCB
88C4 CD4000    00180          CALL    40H             ;INPUT
88C7 78        00190          LD      A,B             ;GET CHARACTER
88C8 B7        00200          OR      A               ;TEST FOR ZERO
88C9 28EE      00210          JR      Z,SETUP         ;GO IF NO INPUT
88CB 210C8A    00220          LD      HL,8A0CH        ;USER START
88CE 110F00    00230          LD      DE,000FH        ;# OF REC TO WRITE
88D1 CD007E    00240          CALL    DUMP            ;JUMP TO DUMP SUBROUTINE
88D4 C3FD88    00250          JP      88FDH           ;RETURN TO PROGRAM
0000           00260          END
00000 TOTAL ERRORS
```

Program Listing 3

You've Got TOTAL ACCESS ©

(specializing in TRS80 *)

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

Rose ©

The complete line of Radio Shack products is available through TA with standard RS limited warranty. Call me for price and delivery. Just cause you don't see it don't mean we ain't got it.
ROSE

Iv'e got

★ **DISK DRIVES**

- TA400 (40-T) $289
- TA800 (80-T) $419
- TA400 Flippy $319
- TA800 Flippy $439
- TA400-2 Dual Head $439
- TA800-2 Dual Head $549

Complete with silver enclosure & power supply.
All TA Drives are MPI.

★ **CABLES**

- 2-Drive $23.95
- 4-Drive $33.95
- Extender Cable $14.95

★ **BARE DRIVES**

- TA400B $259
- TA800B $389
- TA400B Flippy $289
- TA800B Flippy $409

★ **EPSON PRINTERS**

- MX80 .... $469.00
- MX80 F/T $579.00
- MX100 .... $799.00
- Parallel Cables $29.95
- MX 70-80 Ribbons $12.95

★ **OPERATING SYSTEMS**

- TRSDOS 2.3 Disk & Manual $19.95
- LDOS $139.95
- NEWDOS/80 .... $129.95
- DOSPLUS 3.3,3.3D $89.95

★ **IRON**

- Disk Drive Power Supply, Single $37
- Disk Drive Case (silver)and Base $19
- Memorex Diskettes (bx of 10) . $25

90 day warranty on **DRIVES.** Add $5.00 freight per drive in Cont. U.S. UPS COD charge $1.50. There is also a 15 day FREE TRIAL on TA drives. If not completely satisfied I'll refund your money (less shipping). I'll take exception to improper use or mishandling.

★ **RADIO SHACK COMPUTERS**

- 26-1061 MIII, LI, 4K $595
- 26-1062 MIII, LII, 16K $835
- 26-1065 MIII, 48K, 1 Drive $1695
- 26-1066 MIII, 48K-2 Dr-RS232 .. $2099
- 26-3001 4K Color Computer ... $315
- 26-3002 16K w/ext. Basic $485
- 26-3003 32K w/ext. Basic $566
- 26-3501 Pocket Computer $185
- 26-4002 MII, 64K, 1-dr $3288

★ **PERIPHERALS - Model I, II, III**

- 26-1140 OK Exp. Interface $249
- 26-1140 With 16K Mem. Tstd/Guar $299
- 26-1140 With 32K Mem. Tstd/Guar $349
- 26-1145 MI RS232 w/cable $85
- 26-1148 MIII RS232 w/cable $89
- 26-1172 D.C. Modem I $135
- 26-1173 D.C. Modem II $169
- 26-1206 CTR-80 Recorder $51
- 26-3008 CC Joysticks, pr $21
- 26-3010 13" Color Video $353
- 26-4150 MII 8.4MB Disk $3820
- 26-4151 MII Hard Disk No.2 ... $2970
- 16K 200nsec Memory Guar 1yr. $29

★ **OTHER PRINTERS & ACCESSORIES**

- 26-1158 Daisy Wheel II $1694
- 26-1455 Acoustic Cover $339
- 26-1447 Tractor Assy $208
- 26-1448 Sheet Feeder $1111
- 26-1165 Line Printer V $1581
- 26-1166 Line Printer VI $986
- 26-1167 Line Printer VII $339
- 26-1168 Line Printer VIII $679
- 26-1191 Flat Bed Plotter $1695
- 26-1195 Digitizer $379
- ANADEX DP-9500, 9501 .... $1295
- NEC 7730 Spinwriter, 55cps .. $2484
- Daisy Wheel Printer, 45cps ... $1779
- Tractor for above $237
- Sheet Feeder for above $1250
- 26-1308 Printer Stand $88
- 26-4302 Printer Stand $129
- 26-1401 or 4401 Printer Cable .. $32

★ **NEC COMPUTERS**

- PC-8001A Keyboard & Processor $1099
- PC8011A Expansion Unit $699
- PC-8031A 2-drive Expansion Unit $1099
- PC-8041A Green Phos. Video .. $225
- PC-8043A HighRes Color Mon .. $925

ROSE'S SPECIAL OF THE MONTH

★AEROCOMP'S★ ALL NEW DOUBLE DENSITY CONTROLLER

"DDC" $149.95

Here's a Double Density Controller for Mod. 1 that does away with all those data separation problems that seem to keep cropping up! Rose wouldn't fun ya now, would she? Don't be scared of Double Density any more. AEROCOMP has designed some advanced circuitry that even I don't understand - but I do know it works. Better order now. Quanities are going to be a bit short.

$189.95 complete with Dosplus 3.3D. How many do you want? 12?

ZENITH MONITORS

12" Screen * Green Phospher

This is the one you have been waiting for. Well built. Works great with the model I. Also has 40/80 column switch so the Apple folks can use it. 90 day warranty good Nationwide. Sell your old Model I monitor and be the first on your block to have a "real"green screen.

- 12" Zenith Green Phos. Mon $135
- Cable for Model I $7

★ **RADIO SHACK SOFTWARE**

I have most all R/S software instock and it can be yours at 15% off the R/S retail price. Call me for the latest availibility.

DOUBLE DENSITY DATA SEPARATOR FROM AEROCOMP

.... $49.95

If you own a PERCOM "DOUBLER A"*, "DOUBLER II"* or "LNDoubler"** upgrade it with the "DDS". Plugs right in. No more lock-out! You deserve one of these.

ORDER NOW! TOLL FREE 800-527-3582

Write or call ROSE TOLL FREE at 1-800-527-3582. Texas residents call 214-234-1770. Please use the toll free lines for orders and literature requests only. Technical help or service use the Texas line. Prices are mail order only. You pay by VISA or MASTERCARD, you can send check or money order (allow a couple of weeks for personal checks to clear) or order COD (we ship COD'S cash, certified check or money order only). 25% deposit required on all COD orders. Rose will take American money in just about any form. Add freight (UPS where possible) on all orders.Texas residents cough up 5% sales tax. Allow 2-4 weeks for delivery. Order today - I need the money!.

TOTAL ACCESS©
P.O. BOX 3002
RICHARDSON, TX 75080

214-234-1770

✓6

✓See List of Advertisers on page 290

Butler for Edtasm. See Fig. 1. Follow instructions A through E below for Micromusic, substituting the appropriate block moves. Eliza requires 16K, so you must have 32K to perform the transfer.

For Micromusic, a simple transfer is insufficient. The program itself uses cassette input and output for loading and saving the music you write. To change this requires appending a subroutine for disk input and output. By combining features from H. S. Gentry's "Spool and Despool" March 1980 *80 Microcomputing*, and William Barden, Jr.'s Dump/Read program, "Assembly Line" November 1980, *80 Microcomputing*, with a disassembled listing of the program using NEWDOS disassembler, I obtained a fully operational disk program.

The instructions are for a 32K system. In addition, you need a disk copy of Irwin Rappaport's PEEKPOKE program, February 1980, *80 Microcomputing*.

Loading and transferring the program.

At DOS ready, enter Debug and press Enter, Break. Enter D8100, press Enter. Enter M8100, press Spacebar.

Now enter the object code from block move #1, Program Listing 1. Enter G402D. Enter Debug (OFF). Enter Dir.

At DOS Ready Enter Basic2. Enter System. Enter music. At the second ? Type /33023 (8100H). Press the Reset button.

At this point you have two options. If you only want a disk version proceed to section E; if you want the full program, skip E and go to F.

Enter Debug, press Break. Display 8A00H. Enter block move #2 as in block move #1. Enter G402D. Debug (off). Enter TAPEDISK/CMD. Save the program using HIMUSIC:0 8300 89FF 8A00

The program now on disk will enter at 8A00H which will transfer the program from 8300H back to 4300H, and operate like the cassette version.

Relocation

At DOS Ready Enter Basic. Load and run PEEKPOKE. Answer queries with the data in the last column of Table 1. Relocation will take about 10 minutes and should display 124 changes. Unfortunately, more changes will have to be done manually. Press the Reset button. Enter Debug, press Break. Now make the changes listed in Table 2.

Disk save and load

Still in Debug Display 88B9 and enter the code in Program Listing 3. Display 8917 and enter C33889. Display 8938 and enter the code in Program Listing 4. These two programs request Filespec when saving and loading, and then jump to the Dump/Read subroutine.

Dump/Read subroutine

Now display 7E00H. Enter program in Program Listing 5. To enter the code starting at 7E68H using Debug, perform the following: Leave 32 spaces from 7E68H to 7E87H. At 7E88H, enter the ASCII code for the word, Filespec?. The code is 46, 49, 4C, 45, 53, 50, 45, 43, 3F. Enter 0D at 7E91 and 0F at 7E92 and the object code is complete. Enter G402D. Enter DEBUG (OFF). Enter TAPEDISK/CMD. Save the program using DKMUSIC/CMD:0 7E00 89FF 8300.

You now have a copy of Micromusic on disk with disk save and load. The program will save and load 960 bytes or 15 video lines of music. A word of caution: You must load DIR before loading the program or the Break key will not operate properly. Instead of going to command, it will drop out of the program and go to Memory Size?.

Running the program

Center the program under DOS, it will load in and the blinking asterisk will appear. The program operates as in the Micromusic instructions except for save and load. To save your music, press the Break key. The screen will respond with Command?; Enter Save or Load. The screen will respond with Filespec?. Type a one to eight letter title with or without an extension. You can use MUS as an extension to identify the songs on the disk.

I also write the filespecs of all my songs on the screen as though I were composing a song and save it under DIR/MUS. Anytime during the program I can see which songs are on the disk and how I saved them by pressing Break, entering Load, and answering filespec with DIR/MUS. Micromusic will attempt to play this directory, but no harm will be done. To update the list, load it, add the new filespecs and resave it. ■

Address	Contents	Change To
885E	4A	8A
8861	4A	8A
886E	4A	8A
88A2	4A	8A
88E1	4A	8A
8905	4A	8A
890D	4A	8A
8945	4A	8A
89FB	4A	8A

Table 2

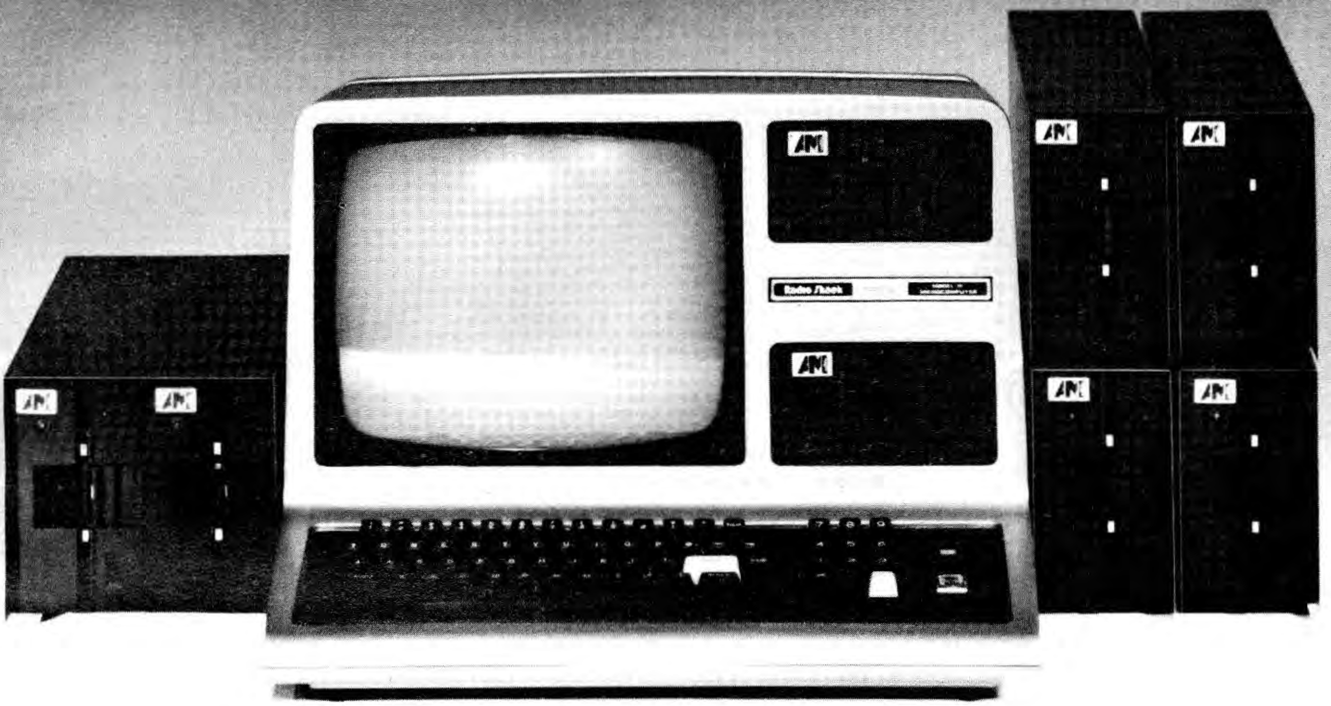
```
8938                  ORG    8938H
7E88          MSC     EQU    7E88H
7E39          READ    EQU    7E39H
7E68          DCB     EQU    7E68H
8938 21887E   SETUP   LD     HL,MSG         ;PUT QUERY IN HL
893B CD6744           CALL   4467H          ;DISPLAY ON SCREEN
893E 21687E           LD     HL,DCB         ;GET DCB ADDRESS
8941 0620             LD     B,32           ;LENGTH OF DCB
8943 CD4000           CALL   40H            ;INPUT
8946 78               LD     A,B            ;GET CHARACTER
8947 B7               OR     A              ;TEST FOR ZERO
8948 28EE             JR     Z,SETUP        ;GO IF NO INPUT
894A 210C8A           LD     HL,8A0CH       ;USER START
894D CD397E           CALL   READ           ;JUMP TO READ SUBROUTINE
8950 C3FD88           JP     88FDH          ;RETURN TO PROGRAM
0000                  END
00000 TOTAL ERRORS
```

Program Listing 4

```
7E00                  ORG    7E00H
7E00 D5       DUMP    PUSH   DE             ;SAVE # OF RECORDS TO WRITE
7E01 E5               PUSH   HL             ;SAVE START ADDRESS
7E02 21947E           LD     HL,BUFFER      ;GET BUFFER ADDRESS
7E05 11687E           LD     DE,DCB         ;GET DCB ADDRESS
7E08 0640             LD     B,64           ;64 BYTE LRL
7E0A CD2044           CALL   4420H          ;INITIALIZE
7E0D 2808             JR     Z,DMP020       ;GO IF NO ERROR
7E0F F680     DMP010  OR     80H            ;PUT ERROR CODE IN A
7E11 CD0944           CALL   4409H          ;ERROR DISPLAY ROUTINE
7E14 CD2D40           CALL   402DH          ;REBOOT
7E17 E1       DMP020  POP    HL             ;GET START ADDRESS
7E18 E5               PUSH   HL             ;SAVE START ADDRESS
7E19 11687E           LD     DE,DCB         ;GET DCB ADDRESS
7E1C CD3C44           CALL   443CH          ;VERIFY
7E1F 20EE             JR     NZ,DMP010      ;GO IF ERROR
7E21 E1               POP    HL             ;GET START ADDRESS
7E22 D1               POP    DE             ;GET # OF REC TO WRITE
7E23 1B               DEC    DE             ;DECREASE COUNT
7E24 7A               LD     A,D            ;GET MS OF BYTE
7E25 B3               OR     E              ;TEST FOR ZERO
7E26 2808             JR     Z,DMP030       ;GO IF DONE
7E28 014000           LD     BC,64          ;LRL
7E2B 09               ADD    HL,BC          ;GET NEXT START ADDRESS
7E2C D5               PUSH   DE             ;SAVE # OF REC
7E2D E5               PUSH   HL             ;SAVE NEW START
7E2E 18E7             JR     DMP020         ;CONTINUE
7E30 11687E   DMP030  LD     DE,DCB         ;GET DCB ADDRESS
7E33 CD2844           CALL   4428H          ;CLOSE
7E36 20D7             JR     NZ,DMP010      ;GO IF ERROR
7E38 C9               RET                   ;RETURN
7E39 3A927E   READ    LD     A,(NOREC1)     ;GET # OF REC TO READ
7E3C 32937E           LD     (NOREC2),A     ;STORE # OF REC TO READ
7E3F E5               PUSH   HL             ;SAVE START
7E40 21947E           LD     HL,BUFFER      ;GET BUFFER ADDRESS
7E43 11687E           LD     DE,DCB         ;GET DCB ADDRESS
7E46 0640             LD     B,64           ;64 BYTE LRL
7E48 CD2444           CALL   4424H          ;OPEN
7E4B 20C2             JR     NZ,DMP010      ;GO IF ERROR
7E4D E1       REA010  POP    HL             ;GET START
7E4E E5               PUSH   HL             ;SAVE START
7E4F 11687E           LD     DE,DCB         ;GET DCB ADDRESS
7E52 CD3644           CALL   4436H          ;READ ONE RECORD
7E55 20B8             JR     NZ,DMP010      ;GO IF ERROR
7E57 3A937E           LD     A,(NOREC2)     ;GET # OF REC TO READ
7E5A 3D               DEC    A              ;DECREASE COUNT
7E5B E1               POP    HL             ;GET START
7E5C 32937E           LD     (NOREC2),A     ;SAVE # TO REC TO READ
7E5F 28CF             JR     Z,DMP030       ;RETURN IF DONE
7E61 014000           LD     BC,64          ;LRL
7E64 09               ADD    HL,BC          ;GET NEW START
7E65 E5               PUSH   HL             ;SAVE NEW START
7E66 18E5             JR     REA010         ;CONTINUE
0020          DCB     DEFS   32
7E88 46       MSG     DEFM   'FILESPEC?'
7E91 0D               DEFB   0DH            ;PUT RETURN HERE
7E92 0F       NOREC1  DEFB   15             ;15 - 64 BYTE RECORDS
0001          NOREC2  DEFS   1
0100          BUFFER  DEFS   256
0000                  END
00000 TOTAL ERRORS
```

Program Listing 5

BORN TO RUN



Get your TRS-80™ up and running with special system and disk drive values like these . . .

Complete TRS-80 Model III microcomputer system with disk drives

A.M. Electronics Model III with 48K RAM, two Tandon 40-track double-density disk drives, A.M. Electronics controller board, DOS & manuals, complete and ready to run! *$1,895.00*

Price includes shipping anywhere in Continental U.S.

A.M. Electronics "Multiplier"

Tired of unreliable double density operation? With *Multiplier* you can upgrade from single-density without doubling your problems *$99.95*

Memory Chips

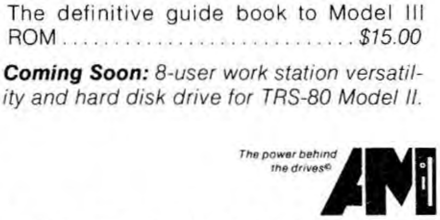
16K 200 nanosecond RAM chips *$2.00 each*

New! Teac double-head 40-track disk drive

Complete with power supply, case and extender cables *$340*

Teac or Tandon disk drives, complete with power supply, case & extender cables

40-track single side disk drive *$250*
80-track single side disk drive *$340*



TRS-80 Model III Commented

The definitive guide book to Model III ROM *$15.00*

Coming Soon: *8-user work station versatility and hard disk drive for TRS-80 Model II.*

A.M. Electronics business software systems put your TRS-80 to work for you

Manufacturing Inventory Control. Six-programs-in-one, for complete control of raw materials from ordering to after-production disbursement *$299.95*

Dealer/Service Center Record Package. Includes flexible, modular units for auto dealers, mechanics, TV repair centers, electronic technicans & appliance repair centers *$199.95*

A.M. Host. Allows your TRS-80 Model III to be run remotely via telephone *$29.95*

Inventory Control System. For complete control over a virtually unlimited amount of inventoried items. Completely menu-driven *$199.95*

All software compatible with Model I or Model III. Each package includes comprehensive, easy-to-follow documentation and support materials.

The power behind the drives©

A.M. ELECTRONICS, INC.

New Ann Arbor Address: 3446 Washtenaw Ave. 48104. (313) 973-2312. Open M-F 10-6, Sat. 10-5
Fowlerville*: 111 North Grand, Box 1071. 48836. (517) 223-7281.
Attention Dealers, OEM's & Distributors: Call us for details on our attractive pricing.

*Authorized Radio Shack Dealer. ™TRS-80 trademark Tandy Corp. Prices subject to change without notice.

452

You've got ten fingers and ten toes, so. . .

The Ten Key Tutor

Mel Knoyle
223 Marietta Street S.E.
Salem, OR 97302

We live in a world that seems to revolve around numbers. Every day we are challenged, frustrated, excited, frightened, thrilled or threatened by them. Ten key devices can enhance our digital dexterity and touch operation is not difficult. A Level II TRS-80 with a ten key pad on the keyboard, this program and a little practice will turn you into a proficient ten key operator.

The program monitors your progress as you practice. As accuracy and speed increase so will the difficulty of the exercises, until you successfully complete a column of ten five-digit numbers.

The Ten Key Tutor is relatively short. The program is designed in five separate modules to make later modifications easier. Each module has only one exit except the Initialization Section and lines 375, 380 and 385 where program repetition is controlled. Table 1 identifies all the program variables and Table 2 lists all lines referring to the variables.

The Tutor

Lines 1–9 display the initial screen with brief instructions. This section is not essential to the program.

Actual program execution begins with line 30. After initialization, the program repeats from line 40. This section develops and displays the exercises. If the previous results contained errors, the exercise is redisplayed. Otherwise a new set of numbers is printed.

INKEY$ is used instead of Input, allowing the program to keep track of response time. The timer begins when the first digit is entered and stops when Enter is pressed.

Lines 305–399 check for elapsed time and errors and print the results. Line 320 converts the random number limit to the number of digits per entry in the last exercise. Lines 355, 360 and 365 identify the errors.

Using The Ten-Key Tutor

On the ten key pad your index finger operates the 7, 4 and 1 keys and rests over the 4. Your middle finger operates the 8, 5 and 2 keys and rests over the 5. (The decimal point is not used in this program.) Your fourth finger rests over the 6 and operates the 9, 6 and 3 keys. Your thumb operates the 0 and your pinky the Enter key.

Type RUN and touch Enter. Read the screen and touch Enter again. Problem 1 should be on the left side of the screen. The cursor is near the center opposite the first number in the column. Begin entering the numbers. Keep a steady rhythm, don't look at the keys and keep practicing. ■

```
1 CLS
2 PRINT"**********************************************************
3 PRINT"*****          T E N   K E Y   T U T O R             *****
4 PRINT"***** Version 1.3   August 30,1980  by Mel Knoyle *****
5 PRINT"**********************************************************

6 PRINT@330,"A LIST OF NUMBERS WILL BE DISPLAYED"
7 PRINT@456,"POSITION YOUR FINGERS OVER THE KEY PAD"
8 PRINT@579,"ENTER THE NUMBERS KEEPING YOUR EYES ON THE SCREEN"
9 PRINT@719,"KEEP  A  STEADY  RHYTHM"
10 '*********************************************************
20 '*****                 INITIALIZATION                *****
30 ER=0:PN=0:X=9:M$="####,###"
40 PRINT@969,"TOUCH ANY KEY WHEN READY TO CONTINUE";
50 GO$=INKEY$:IF GO$="" THEN GOTO 50 ELSE T=0:A=0:B=0:CLS
100 '********************************************************
110 '*****                PROBLEM SET-UP                 *****
120 PRINT@0,"EXERCISE #";:IF E PRINT PN ELSE PN=PN+1:PRINT PN
130 FOR I=1 TO 10
140     IF E=0 THEN A(I)=RND(X)
150     PRINT@I*64,;:PRINTUSING M$;A(I):A=A+A(I)
160 NEXT I
170 PRINT@704,STRING$(9,"-"):PRINT@768,;:PRINTUSING M$;A;
200 '********************************************************
205 '*****                INPUT ROUTINE                  *****
210 FOR I=1 TO 10
215     PRINT@I*64+36,"? ";CHR$(95);
220     Q$=INKEY$:IF T THEN T=T+.02
225     IF Q$="" THEN 220ELSE T=T+.02
230     IF Q$=CHR$(13) THEN B$="":GOTO 245
235     IF LEN(B$)<5 THEN B$=B$+Q$:B(I)=VAL(B$)
240     PRINT@I*64+31,;:PRINTUSING M$;B(I);:GOTO 220
245     PRINT" ";:B=B+B(I)
250 NEXT I
255 PRINT@735,STRING$(9,"-");:PRINT@799,;:PRINTUSING M$;B
300 '********************************************************
305 '*****            PERFORMANCE EVALUATION             *****
310 PRINT"ELAPSED TIME =";FIX(T*2);"SEC."
315 E=B-A:IF E<>0 PRINTUSING"ERROR ="+M$;E:GOTO 355
320 XN=INT(LOG(X+1)/LOG(10))
325 IF T<XN+2 THEN X=X*10+9:PRINT"EXCELLENT":GOTO345
330 IF T<XN+4 THEN PRINT"GOOD":GOTO345
335 IF T<XN+8 THEN PRINT"KEEP A STEADY RHYTHM":GOTO345
340 PRINT"RELAX..TAKE A DEEP BREATH..AND TRY AGAIN."
345 IF X=327679 AND E=0 THEN GOTO 380
350 IF X>32767 THEN X=32767
355 FOR I=1 TO 10
360     IF A(I)<>B(I)THEN PRINT@I*64+40,"?";
370 NEXT I
375 ER=ER+E:GOTO 40
380 IF ER THEN PRINT"TRY TO IMPROVE YOUR ACCURACY":GOTO 30
385 IF PN>5 THEN PRINT"TRY TO INCREASE YOUR SPEED":GOTO 30
390 PRINT"CONGRATULATIONS, ";
399 PRINT"YOU ARE A PROFICIENT TEN-KEY OPERATOR!":END
999 '******************** END OF PROGRAM ********************
```

Program Listing

The Key Box

Basic Level II
Model I
Ten Key Pad

Variable	Definition
A	TOTAL OF PROBLEM VALUES
A(1-10)	ARRAY CONTAINING PROBLEM VALUES
B	TOTAL OF INPUT VALUES
B(1-10)	ARRAY CONTAINING INPUT VALUES
B$	HOLDS MULTIPLE DIGITS (INKEY$)
E	ERROR VALUE
ER	ERROR INDICATOR
GO$	PAUSE CONTROL (DUMMY VARIABLE)
I	LOOP COUNTER
M$	NUMERIC OUTPUT MASK
PN	PROBLEM NUMBER
Q$	INKEY$ INPUT VARIABLE
T	ELAPSED TIME COUNTER
X	RANDOM NUMBER LIMIT
XN	NUMBER OF COLUMNS

Table 1. Program Variables

Variable	Line Numbers
PN	30,120,385
X	30,140,320,325,345,350
M$	30,150,170,240,255,315
ER	30,375,380
GO$	50
A	50,150,170,315
T	50,220,225,310,325,330,335
B	50,245,255,315
XN	50,320,325,330,335
I	130,140,150,160,210,215,235,240,245,250,355,360,370
A(1-10)	140,150,360
E	140,315,345
Q$	220,225,230,235
B$	230,235
B(1-10)	235,240,245,360

Table 2. Variable Locations

Pop art's out, comp art's in.

The Random Picture Generator

James A. Swarts
758 West Dale Avenue
Muskegon, MI 49441

The TRS-80 is more than a microcomputer—it's an artist! With the proper software you can paint modern art masterpieces on the computer's video display in a matter of seconds. You can create everything from Indian art to spaceship portraits. Your imagination is the only limit.

The Random Picture Generator (see the Program Listing) is a very simple program! It was the first program I ran on my TRS-80. Later I decided to add a routine to produce reverse video pictures. I quickly wrote the necessary Assembly language program and converted it to numeric data which I POKE into high memory. I changed all the Set verbs to Reset in order to have dark pictures drawn on a light background.

The Random Picture Generator isn't a complicated program, but it is an entertaining program. It also proves that TRS-80 graphics aren't all that bad. ■

```
30 RANDOM:FORN = 1TO50:CLS
40 X = RND(128) - 1:A = RND(128) - 1
50 Y = RND(48) - 1:B = RND(48) - 1
60 FORI = 1TO10:D = RND(8):FORJ = 1TORND(20) + 5
70 ONDGOSUB150,170,190,210,160,180,200,220
80 IFX<0THENX = 127:ELSEIFX>127THENX = 0
90 IFA<0THENA = 127:ELSEIFA>127THENA = 0
100 IFY<0THENY = 47:ELSEIFY>47THENY = 0
110 IFB<0THENB = 47:ELSEIFB>47THENB = 0
120 SET(X,Y):SET(127 - X,Y):SET(X,47 - Y):SET(127 - X,47 - Y)
130 SET(A,B):SET(127 - A,B):SET(A,47 - B):SET(127 - A,47 - B)
140 NEXTJ:NEXTI:FORT = 1TO1500:NEXTT:NEXTN:END
150 X = X + 1:A = A - 1
160 Y = Y + 1:B = B - 1:RETURN
170 Y = Y + 1:B = B - 1
180 X = X - 1:A = A + 1:RETURN
190 X = X - 1:A = A + 1
200 Y = Y - 1:B = B + 1:RETURN
210 Y = Y - 1:B = B + 1
220 X = X + 1:A = A - 1:RETURN
```

Program Listing

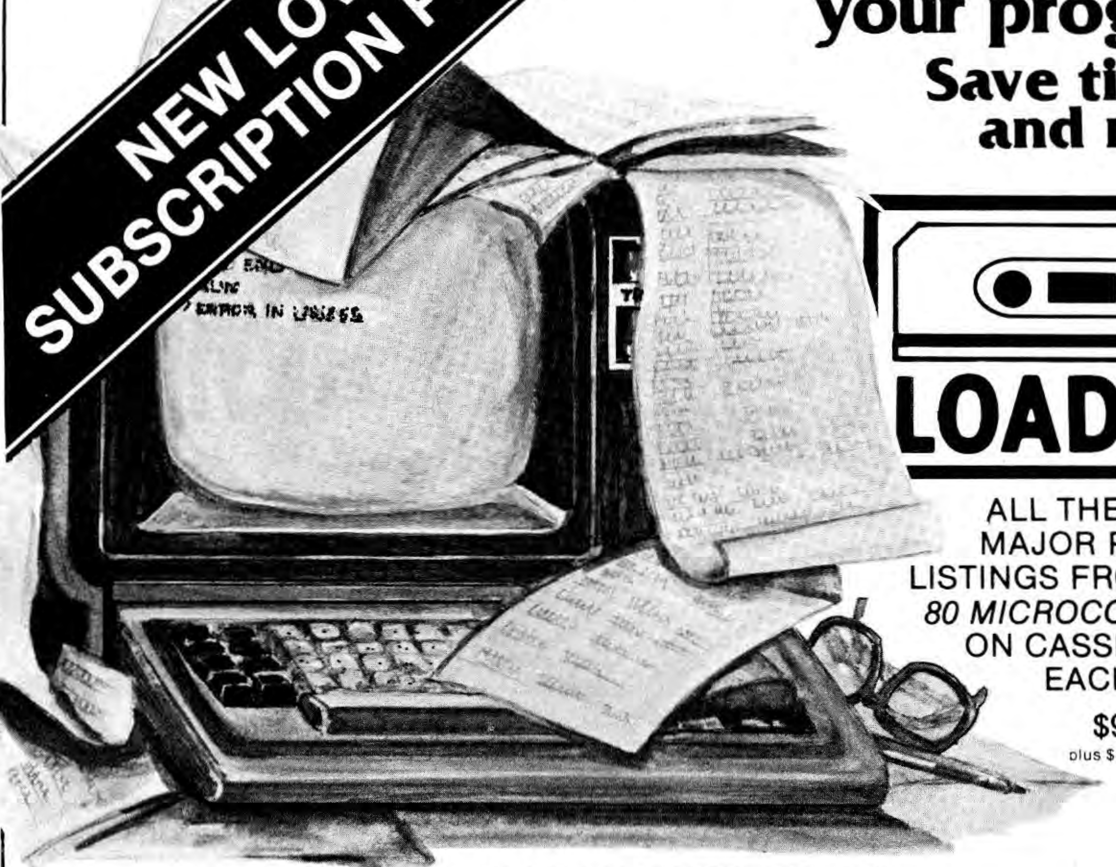
```
15 FORL = 32000TO32015:READP:POKEL,P:NEXTL:POKE16526,0:POKE16527,125
30 RANDOM:FORN = 1TO50:U = USR(0)
120 RESET(X,Y):RESET(127 - X,Y):RESET(X,47 - Y):RESET(127 - X,47 - Y)
130 RESET(A,B):RESET(127 - A,B):RESET(A,47 - B):RESET(127 - A,47 - B)
225 DATA 229,33,0,60,54,191,17,1,60,1,255,3,237,176,225,201
```

Modifications to the Program

NEW LOW SUBSCRIPTION PRICE

Tired of typing in all your programs?

Save time and money!



LOAD 80

ALL THE MAJOR PROGRAM LISTINGS FROM *80 MICROCOMPUTING* ON CASSETTE EACH MONTH

$9.95
plus $1.50 ea. postage & handling

SAVE HOURS OF TYPING & AGGRAVATION

Sign up for a subscription to LOAD-80. Save 30% off the single tape price.
Begin your subscription with the February 1982 cassette.

"Frankly, after hundreds of hours of frustration, I seldom ever try to keyboard a published program. Inevitably, I mess it up when I keyboard it. Who needs the aggravation?

"This is why I started a new series of cassettes called **LOAD-80**. Each cassette will have program dumps of the listings in an issue of *80 Microcomputing*. These listings are direct from the authors and tested by the *80 Microcomputing* staff. All but the very short program listings will be on the **LOAD-80** cassettes. Thus, you will be able to save hours of inputting programs and even more of debugging your keyboard errors."

Wayne Green

Wayne Green
Publisher, *80 Microcomputing*

The **LOAD-80** cassette is simply the program listings that appear in the articles in *80 Microcomputing*. It was created to save you the time involved in typing the listings yourself. Successful loading of the programs depends on reading the documentation in the articles. If you have your current magazine at hand when you load the cassette, you should have no difficulty. If you still have problems, please return the tape for replacement.

LOAD-80 began with the April 1981 issue. To order back issues, look for the back issue advertisement in this magazine or ask your local dealer.

To order **LOAD-80**, fill in the attached card and we will send you your **LOAD-80** cassette for the major programs in this issue. If the card is missing, fill in the coupon below (a photocopy is acceptable) and mail to: **LOAD-80**, 80 Pine Street, Peterborough, NH 03458.

☐ **YES,** I'll save 30%. Sign me up for a year of **LOAD-80** beginning with the February 1982 cassette... $99.97 a year

☐ **Visa** ☐ **Am. Ex** ☐ **MasterCard**
☐ **Check/Money order**

☐ **YES,** I want February **LOAD-80** for $9.95 plus $1.50 each postage and handling

Name: ______________________

Address: ______________________

City: ______________ State: ________ Zip: ____________

Signature: ______________ Card # ______________ Exp. Date: ____________

LOAD-80 is manufactured by Instant Software, a division of WAYNE GREEN INC., 80 Pine Street, Peterborough, NH 03458. There is no warranty expressed or implied that **LOAD-80** will do anything other than save you typing.

Foreign air mail, please inquire. **2/82**

LOAD 80

A program for coaches to help put together a winning unit.

Dream Team

Dennis Wangsness
2978 Cohansey Drive
San Jose, CA 95132

Getteam creates optimum teams based on information you supply about the skills and positions for the team. A soccer team needs 11 people. There are nearly 40 million ways to arrange those people among the 11 team positions. Sixteen people can be arranged in over 170 billion ways among the same 11 positions.

The Program

You will have to input information concerning skills, names of people, person's skill level numerically valued, names of tasks/positions to be filled, values of skills to tasks and order of importance of tasks.

Assume, again, we are placing people on a soccer team. Some of the skills to list are heading, dribbling, playmaking and passing. Tasks/Positions might be striker, center fullback, right wing and left halfback.

Quantities requiring numerically valued input should be of a common range. Names of team members, skills and tasks should be kept short for a neat output.

The program first asks if you want instructions. If you answer N it then asks if you want to recall saved information. An N response causes the program to proceed to data entry.

You will be asked to enter the names of the skills. When you have no more skills to list enter Last.

After the skill names, you are prompted for the name of a person. After each name the program requests a score for each skill. The cycle repeats until you enter Last.

Next is task/position names and scores. This sequence proceeds as did the names and ability levels. The scores indicate the relative importance of skills to positions. Again, enter Last when no further input is needed.

When keyboard input is completed you are asked if you want to change anything. A response of Y produces a menu of things to change. After your changes, the program calculates the strategy matrix. This grades each person's abilities at each position. The matrix can be displayed or printed.

You will be asked to specify the task order of importance. If you choose not to specify, it is assumed that you have input the tasks in order of importance. If not, input your choice from the displayed possibilities.

The Best Team

The program calculates your best team by picking the best player for the most important position, the best of the remaining players for your second most important position, and so on, until all positions are filled. If desired you can display or print the team listing.

Next the program asks if you want a computer optimized team. If you respond Y the players are arranged and rearranged until the strongest team is found. Again you are given the chance to display or print.

Once the computer optimized team is picked you can change the task order of importance. Do not be surprised if the computer optimized team is similar for each choice of task order of importance. The optimized team is usually very balanced.

At the end of the program you are able to save the information. ■

Program Listing

```
4 '**************************************************
5 '***************** GETTEAM/BAS ********************
6 '**************************************************
10 CLEAR 1000
15 DEFINT A-Z:CH=1
20 DIM S$(20),N$(20),PS(20,20),T$(20),ST(20,20),PT(20,20),I(20),
J(20),I$(20)
25 CLS:INPUT"DO YOU NEED INSTRUCTIONS (Y/N)";Q$
30 IF Q$<>"Y"ANDQ$<>"N" THEN GOTO 25 ELSE IF Q$="Y"THEN GOSUB600
0
35 CLS:INPUT"WANT TO RECALL SAVED INFORMATION (Y/N)";Q$
40 IF Q$<>"Y"ANDQ$<>"N"THENGOTO35 ELSE IF Q$="Y"THEN GOSUB6500
42 IF Q$="N" THEN GOTO 100
45 CLS:INPUT"WANT TO CHANGE ANY NAMES OR SCORES (Y/N)";Q$
50 IF Q$<>"Y"ANDQ$<>"N" THEN GOTO45
52 IF Q$="Y" THEN GOSUB 4000
55 CLS:IF Q$="N"THEN GOTO400 ELSE INPUT"ANY MORE CHANGES (Y/N)";
Q$
60 IF Q$<>"Y"AND Q$<>"N"THEN GOTO55 ELSE IF Q$="Y" THEN GOTO52
70 GOTO 400
100 PRINT"INPUT SKILL WHEN PROMPTED, WHEN DONE ENTER -LAST"
110 S=1
120 PRINT"INPUT SKILL";S;"  ";:INPUT S$(S)
130 IF S$(S)<>"LAST" THEN S=S+1:GOTO 120
140 S=S-1
150 CLS:PRINT"INPUT PERSON'S NAME AND SKILL SCORE AS PROMPTED"
160 PRINT"ENTER - LAST WHEN FINISHED"
170 N=1
180 PRINT"ENTER NAME NUMBER";N;"  ";:INPUT N$(N)
190 IF N$(N)="LAST" THEN N=N-1:GOTO 250
200 FOR I=1 TO S
210 PRINT"INPUT ";N$(N);"'S SCORE FOR ";S$(I);:INPUT PS(N,I)
220 NEXT I
230 N=N+1
240 GOTO 180
250 CLS:PRINT"INPUT TASK/POSITION AND SKILL VALUE AS PROMPTED"
260 PRINT"IT WILL BE CONVIENT TO INPUT TASKS IN ORDER"
270 PRINT"OF IMPORTANCE.  ENTER - LAST WHEN DONE."
280 T=1
290 PRINT"ENTER TASK/POSITION NUMBER";T;:INPUT T$(T)
300 IF T$(T)="LAST" THEN T=T-1:GOTO 355
310 FOR I=1 TO S
320 PRINT"ENTER VALUE OF ";S$(I);" TO ";T$(T);" ";:INPUTST(I,T)
```

Program continues

The Key Box

Basic Level II
Model I
32K RAM
TRSDOS
1 Disk Driver
Printer optional

Dennis Wangsness' formal training is in Engineering Physics. His computer application interests include sports and education.

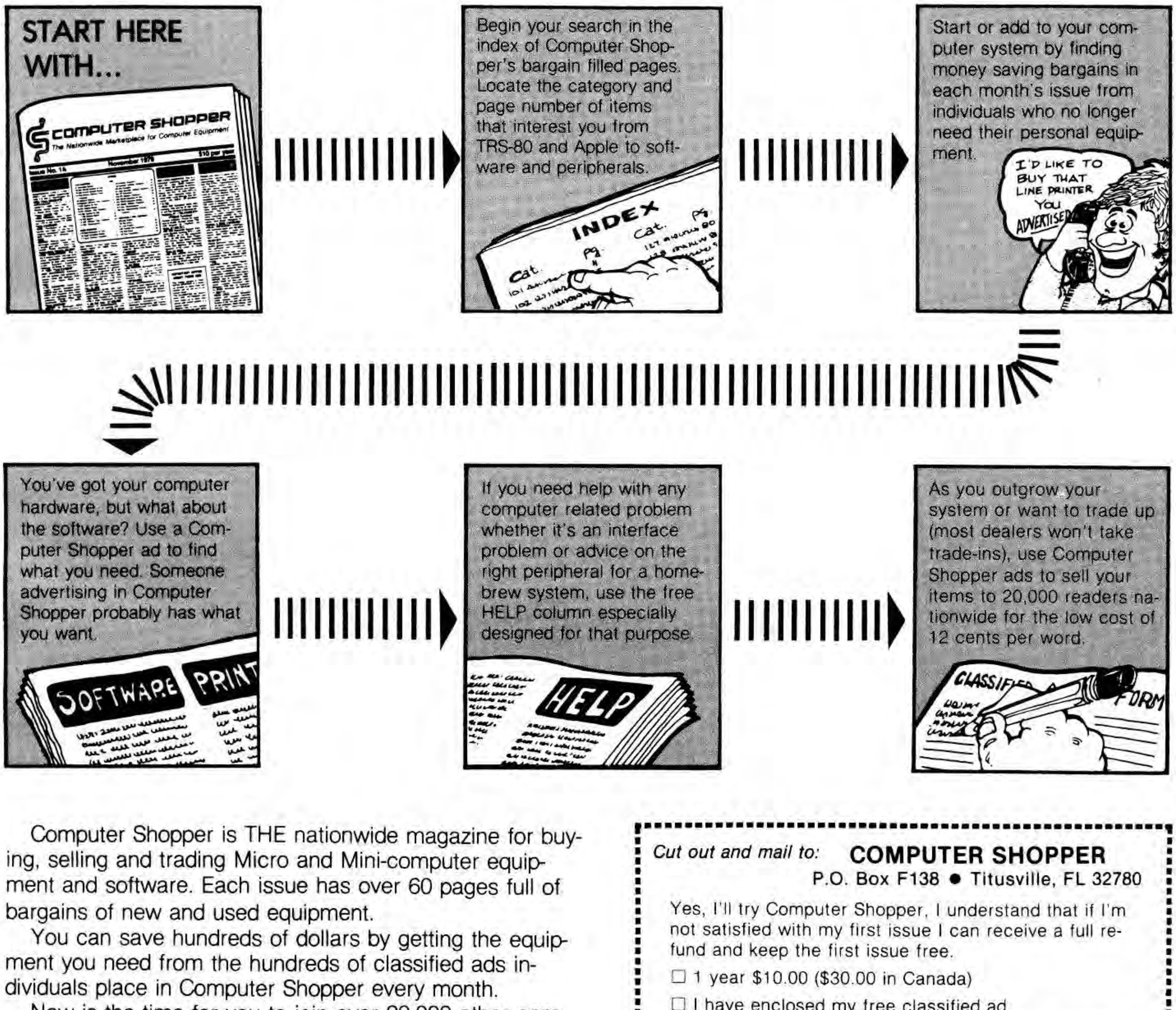
```
330 NEXT I
340 T=T+1
350 GOTO 290
355 FOR I=1 TO T:I(I)=I:NEXT I
360 CLS:PRINT"DO YOU WANT TO CHANGE ANY VALUES (Y/N)"
370 INPUT Q$:IF Q$<>"Y"AND Q$<>"N" THEN GOTO 360
380 IF Q$="Y" THEN GOSUB 4000
390 IF CH=0 THEN GOTO 470
400 CLS:PRINT@466,"COMPUTING PERSON-TASK MATRIX"
410 FOR I=1 TO N
420 FOR J=1 TO T
430 PT(I,J)=0
440 FOR K=1 TO S:PT(I,J)=PT(I,J)+PS(I,K)*ST(K,J):NEXT K
450 NEXT J
452 PRINT@543,I;
455 NEXT I
460 CH=0
470 CLS:INPUT"WANT THE COMPLETE STRATAGY MATRIX PRINTED (Y/N)";Q
$
480 IF Q$<>"Y"AND Q$<>"N" THEN GOTO 470
490 IF Q$="Y" THEN GOSUB 2500
510 PRINT"DO YOU WANT TO SPECIFY THE TASK ORDER OF IMPORTANCE (Y
/N)"
520 INPUT Q$:IF Q$<>"Y"AND Q$<>"N" THEN GOTO 510
530 IF Q$="Y" THEN GOSUB 3000
600 GOSUB2000: REM *******DETERMINE BEST TEAM*******
610 GOSUB3500: REM *******PRINT THE BEST TEAM*******
613 GOSUB2200: REM ***** REFINE TEAM ************
616 IF Q$="Y"THEN GOSUB3500: REM ****** PRINT REFINED TEAM ***
620 INPUT"WANT TO MAKE ANY CHANGES TO YOUR INPUT (Y/N)";Q$
630 IF Q$<>"Y"AND Q$<>"N" THEN GOTO 620 ELSE IF Q$="Y" THEN GOTO
 380
635 INPUT"CHANGE TASK ORDER OF IMPORTANCE (Y/N)";Q$
637 IF Q$="Y" THEN GOTO 530
640 INPUT"WANT YOUR INPUTS SAVED (Y/N)";Q$
650 IF Q$<>"Y"AND Q$<>"N"THEN GOTO 640 ELSE IF Q$="Y" THEN GOSUB
 5500
655 IF Q$="N" THEN GOTO 680
660 CLS: INPUT"WANT TO SAVE A BACKUP OF YOUR DATA (Y/N)";Q$
670 IF Q$<>"Y"AND Q$<>"N" THEN GOTO 660 ELSE IF Q$="Y" THEN GOSU
B5500
680 CLS:INPUT"WANT THE MENU AGAIN (Y/N)";Q$
690 IF Q$<>"Y"AND Q$<>"N" THEN GOTO 680 ELSE IF Q$="Y" THEN GOTO
380
700 END
2000 REM ************* FIND BEST TEAM *****************
2005 CLS:PRINT@472,"FINDING GOOD TEAM"
2010 FOR J=1 TO N:J(J)=0:NEXT J : REM-INIT. PLAYER INDEX VECT.
2020 FOR I=1 TO T: REM **** TASK LOOP ****
2030    M=0
2040      FOR J=1 TO N: REM **** PERSON LOOP ****
2050 IF J(J)=0 AND PT(J,I(I))>=PT(M,I(I)) THEN M=J
2060 NEXT J
2070 J(M)=I(I)
2075 PRINT@542,I;
2080 NEXT I
2100 RETURN
2200 REM ******* SUBROUTINE TO REFINE TEAM ************
2210 PRINT"WANT COMPUTER TO SEARCH FOR BETTER TEAM (Y/N)";
2220 INPUT Q$:IF Q$<>"Y"AND Q$<>"N" THEN GOTO2210
2230 IF Q$="N" THEN RETURN
2231 SS=0
2232 FOR I=1 TO N
2233 IF J(I)<>0 THEN SS=SS+PT(I,J(I))
2234 NEXT I
2240 CLS:PRINT@470,"REFINING THE TEAM"
2242 C=0
2245 Q=0
2250 FOR K=1 TO N-1
2260 IF J(K)=0 THEN GOTO 2320
2270 FOR L=K+1 TO N
2280 IF J(L)=0 THEN GOTO 2310
2285 SD=PT(K,J(L))+PT(L,J(K))-PT(K,J(K))-PT(L,J(L))
2290 IF SD>0 THEN TK=J(K):J(K)=J(L):J(L)=TK:Q=Q+1:SS=SS+SD
2310 NEXT L
2315 PRINT@542,K;
2320 NEXT K
2322 C=C+1:PRINT@58,C;:PRINT@122,Q;:PRINT@186,SS
2325 IF Q=0 OR C=20 THEN RETURN ELSE GOTO 2245
2330 RETURN
2500 REM *********** PRINT STRATAGY MATRIX *************
2510 CLS
2520 PRINT"INPUT THE NUMBER CORRESPONDING TO YOUR PRINT OPTION"
:PRINT:PRINT
2530 PRINT"      1 - PRINT ON SCREEN ONLY"
2540 PRINT"      2 - PRINT ON PRINTER ONLY"
2550 PRINT"      3 - PRINT ON SCREEN AND PRINTER"
2555 INPUT Q
2560 IF Q<1 OR Q>3 THEN GOTO 2510
2570 IF T/5=INT(T/5) THEN K1=1:K2=T/5 ELSE K1=1:K2=INT(T/5)+1
2580 FOR K=K1 TO K2
2590 I0=5*K:I1=I0-1:I2=I0-2:I3=I0-3:I4=I0-4
2600 IF Q=1 OR Q=3 THEN PRINT TAB(10)T$(I4);TAB(20)T$(I3);TAB(30
)T$(I2);TAB(40)T$(I1);TAB(50)T$(I0)
2610 IF Q=2 OR Q=3 THEN LPRINT TAB(10)T$(I4);TAB(20)T$(I3);TAB(3
0)T$(I2);TAB(40)T$(I1);TAB(50)T$(I0)
2620 FOR I=1 TO N
2630 IF Q=1 OR Q=3 THEN PRINTN$(I);TAB(10)PT(I,I4);TAB(20)PT(I,I
3);TAB(30)PT(I,I2);TAB(40)PT(I,I1);TAB(50)PT(I,I0)
2640 IF Q=2 OR Q=3 THEN LPRINT N$(I);TAB(10)PT(I,I4);TAB(20)PT(I
,I3);TAB(30)PT(I,I2);TAB(40)PT(I,I1);TAB(50)PT(I,I0)
2650 NEXT I
2660 PRINT:PRINT
2670 NEXT K
2675 IF Q=2 OR Q=3 THEN LPRINT" "
2680 RETURN
2990 IF SD>0 THEN TK=J(K):J(K)=J(L):J(L)=TK:Q=Q+1:SS=SS+SD
3000 REM *********** TASK RANKING **************
3010 CLS
3020 PRINT"EACH TASK/POSITION WILL BE LISTED AND YOU ARE TO PROV
IDE"
3030 PRINT"IT'S RANK IN IMPORTANCE.  NO TWO TASKS CAN HAVE"
3040 PRINT"THE SAME RANK.  IF THERE ARE N TASKS/POSITIONS, THEN
ALL"
3050 PRINT"OF THE NUMBERS FROM 1 THROUGH N MUST BE USED "
3060 PRINT"IN THE RANKING.  THE CURRENT RANK WILL BE IN PARENS"
3070 PRINT"AFTER EACH TASK/POSITION.
3071 FOR I=1 TO T:T$=STR$(I):L=LEN(T$)
3072 T$=STRING$(4-L," ")+T$:I$(I)=T$
3073 NEXT I :P$="":FOR I=1 TO T:P$=P$+I$(I):NEXT I
3074 PRINT@704,P$
3075 PRINT@640,"RANKS REMAINING";
3080 FOR I=1 TO T
3090 PRINT@512, T$(I);" (";I(I);") NEW RANK =";:INPUT I(I)
3092 I$(I(I))="    ":P$="":FOR J=1TOT:P$=P$+I$(J):NEXT J
3094 PRINT@704,P$
3095 PRINT@512,STRING$(60," ");
3100 NEXT I
3110 PRINT:INPUT"THAT IS ALL.  WANT TO CHANGE THEM AGAIN (Y/N)";
Q$
3120 IF Q$<>"Y"AND Q$<>"N" THEN GOTO 3110
3130 IF Q$="Y" THEN GOTO 3010
3140 RETURN
3500 REM ********** PRINT SELECTED TEAM ************
3510 CLS:PRINT"TO PRINT THE TEAM -"
3520 PRINT"INPUT THE NUMBER CORRESPONDING TO YOUR PRINT OPTION":
PRINT:PRINT
3530 PRINT"      1 - PRINT ON SCREEN ONLY"
3540 PRINT"      2 - PRINT ON PRINTER ONLY"
3550 PRINT"      3 - PRINT ON SCREEN AND PRINTER"
3560 INPUT Q
3570 IF Q<1 OR Q>3 THEN GOTO 3510
3575 FOR I=1 TO N
3580 IF J(I)=0 THEN GOTO 3610
3590 IF Q=1 OR Q=3 THEN PRINT T$(J(I));" - ";N$(I)
3600 IF Q=2 OR Q=3 THEN LPRINT CHR$(14);T$(J(I));" - ";N$(I)
3610 NEXT I
3615 IF Q=2 OR Q=3 THEN LPRINT" ":LPRINT" ":LPRINT" "
3620 RETURN
4000 REM ********* MODIFICATION OF INPUT ***********
4010 CLS:PRINT"ENTER NUMBER CORRESPONDING TO WHAT YOU WANT TO CH
ANGE."
4020 PRINT:PRINT
4030 PRINT"      1 - CHANGE SKILL NAME
4040 PRINT"      2 - CHANGE PERSON NAME"
4050 PRINT"      3 - CHANGE PLAYER SKILL SCORE"
4060 PRINT"      4 - CHANGE TASK NAME"
4070 PRINT"      5 - CHANGE SKILL VALUE TO TASK"
4080 PRINT"      6 - NO CHANGES"
4100 INPUT Q:IF Q<1 OR Q>6 THEN GOTO 4010
4105 IF Q=6 THEN RETURN
4110 CH=1:ON Q GOTO 4200,4400,4600,4800,5000
4200 CLS:PRINT"EACH SKILL WILL BE DISPLAYED."
4210 PRINT"PRESS THE Y KEY IF YOU WANT TO CHANGE THE NAME."
4220 PRINT"PRESS THE N KEY TO LEAVE THE NAME."
4230 FOR I=1 TO S
4240 PRINT@389,S$(I)
4250 Q$=INKEY$:IF Q$<>"Y"AND Q$<>"N" THEN GOTO 4250
4260 IF Q$="N" THEN GOTO 4290
4270 PRINT@453,"OLD NAME IS ";S$(I);" NEW NAME =";:INPUT S$(I)
4280 PRINT@453,STRING$(64," ")
4290 NEXT I
4300 INPUT"WANT THE MENU AGAIN (Y/N)";Q$
4310 IF Q$="Y" THEN GOTO 4010 ELSE IF Q$<>"N" THEN GOTO 4300
4320 RETURN
4400 CLS:PRINT"EACH NAME WILL BE DISPLAYED."
4410 PRINT"PRESS THE Y KEY TO CHANGE THE NAME."
4420 PRINT"PRESS THE N KEY TO LEAVE THE NAME"
4430 FOR I=1 TO N
4440 PRINT@389,N$(I)
4450 Q$=INKEY$:IF Q$<>"Y"ANDQ$<>"N" THEN GOTO 4450
4455 PRINT@389,STRING$(64," ")
4460 IF Q$="N" THEN GOTO 4490
4470 PRINT@453,"OLD NAME IS ";N$(I);" NEW NAME =";:INPUTN$(I)
4480 PRINT@453,STRING$(64," ")
4490 NEXT I
4500 INPUT"WANT THE MENU AGAIN (Y/N)";Q$
4510 IF Q$="Y" THEN GOTO 4010 ELSE IF Q$<>"N" THEN GOTO 4500
4520 RETURN
4600 CLS:INPUT"WHICH PERSONS SKILL SCORE NEEDS CHANGING";Q$
4610 FOR I=1 TO N
4620 IF Q$=N$(I) THEN J=I:GOTO 4650
4630 NEXT I
4640 INPUT"NAME NOT FOUND, INPUT NAME AGAIN.";Q$:GOTO 4610
4650 PRINT"INPUT ";N$(J);"'S SCORE FOR TASK INDICATED"
4655 B$=STRING$(64," ")
4660 FOR I=1 TO S
4670 PRINT@453,S$(I);"(";PS(J,I);")";:INPUT PS(J,I)
4675 PRINT@453,B$
4680 NEXT I
4690 INPUT"WANT THE MENU AGAIN (Y/N)";Q$
4700 IF Q$="Y" THEN GOTO4010 ELSE IF Q$<>"N" THEN GOTO 4690
4710 RETURN
4800 CLS:PRINT"EACH TASK WILL BE DISPLAYED."
4810 PRINT"PRESS Y IF YOU WANT TO CHANGE THE TASK NAME."
4820 PRINT"PRESS N TO LEAVE THE TASK NAME ALONE."
4825 B$=STRING$(64," ")
4830 FOR I=1 TO T
4840 PRINT@389,T$(I)
4850 Q$=INKEY$:IF Q$<>"Y" AND Q$<>"N" THEN GOTO 4850
4860 IF Q$="N" THEN GOTO 4890
4870 PRINT@453,"OLD TASK NAME IS ";T$(I);" NEW NAME =";:INPUT T$
(I)
4880 PRINT@453,B$
4890 NEXT I
4900 INPUT"WANT THE MENU AGAIN (Y/N)";Q$
4910 IF Q$="Y" THEN GOTO 4010 ELSE IF Q$<>"N" THEN GOTO 4900
4920 RETURN
5000 CLS:PRINT"EACH TASK WILL BE DISPLAYED."
5010 PRINT"PRESS THE Y KEY TO CHANGE THE SKILL VALUE FOR THE TAS
K"
5020 PRINT"PRESS THE N KEY FOR NO CHANGE"
5025 B$=STRING$(64," ")
5030 FOR I=1 TO T
5040 PRINT@389,T$(I)
5050 Q$=INKEY$:IF Q$<>"Y" AND Q$<>"N" THEN GOTO 5050
5060 IF Q$="N" THEN GOTO 5107
5070 FOR J=1 TO S
5080 PRINT@453,"SKILL VALUE FOR ";S$(J);" IS";ST(J,I);" NEW VALU
E =";:INPUT ST(J,I)
```

Program continues

Program continued

```
5100 PRINT@453,B$
5105 NEXT J
5107 PRINT@389,B$
5110 NEXT I
5120 INPUT"WANT THE MENU AGAIN (Y/N)";Q$
5130 IF Q$="Y"THEN GOTO 4010 ELSE IF Q$<>"N" THEN GOTO 5120
5140 RETURN
5500 REM ******** SAVE INFORMATION *********
5510 CLS:INPUT" WHAT SHOULD THE DATA FILE NAME BE";FILE$
5520 OPEN"O",1,FILE$
5530 PRINT#1,S;N;T;
5540 FOR I=1 TO N
5550 FOR J=1 TO S
5560 PRINT#1,PS(I,J);
5570 NEXT J,I
5580 FOR I= 1 TO T:PRINT#1,CHR$(34);T$(I);CHR$(34);:NEXT I
5585 FOR I=1 TO N:PRINT#1,CHR$(34);N$(I);CHR$(34);:NEXT I
5587 FOR I=1 TO S:PRINT#1,CHR$(34);S$(I);CHR$(34);:NEXT I
5590 FOR I=1 TO S
5600 FOR J=1 TO T
5610 PRINT#1,ST(I,J);
5620 NEXT J,I
5630 FOR I=1 TO T:PRINT#1,I(I);:NEXT I
5640 FOR I=1 TO N:PRINT#1,J(I);:NEXT I
5650 CLOSE 1
5660 RETURN
6000 REM ************** INSTRUCTIONS *****************
6010 CLS
6020 PRINT"YOU WILL BE PROMPTED FOR ALL THE INPUT REQUIRED."
6030 PRINT"YOU WILL BE ASKED FOR:"
6040 PRINT"    1 - THE SKILLS REQUIRED"
6050 PRINT"    2 - NAMES OF PEOPLE"
6060 PRINT"    3 - PERSONS SKILL VALUE/RANK (INTEGERS)"
6070 PRINT"    4 - NAMES OF TASKS/POSITIONS TO BE FILLED"
6080 PRINT"    5 - VALUE OF SKILLS TO TASKS (INTEGERS)"
6090 PRINT"    6 - ORDER OF IMPORTANCE OF TASKS"
6100 PRINT"YOU WILL BE ABLE TO MAKE CORRECTIONS.  THE"
6110 PRINT"PROGRAM WILL PROMPT YOU FOR CORRECTIONS AT THE "
6120 PRINT"APPROPRIATE TIME."
6130 PRINT"PRESS ANY KEY TO CONTINUE"
6140 Q$=INKEY$:IF Q$="" THEN GOTO 6140 ELSE RETURN
6500 REM ******** DATA FETCH ROUTINE ***********
6510 CLS:INPUT"INPUT DATA FILE NAME";FILE$
6520 OPEN"I",1,FILES$
6530 INPUT#1,S,N,T
6540 CLOSE:OPEN "I",1,FILES$
6550 INPUT#1,X,Y,Z
6560 FOR I=1 TO N
6570 FOR J=1 TO S
6580 INPUT#1,PS(I,J)
6590 NEXT J,I
6600 FOR I=1 TO T:INPUT#1,T$(I):NEXT I
6605 FOR I=1 TO N:INPUT#1,N$(I):NEXT I
6607 FOR I=1 TO S:INPUT#1,S$(I):NEXT I
6610 FOR I=1 TO S
6620 FOR J=1 TO T
6630 INPUT#1,ST(I,J)
6640 NEXT J,I
6650 FOR I=1 TO T:INPUT#1,I(I):NEXT I
6660 FOR I=1 TO N:INPUT#1,J(I):NEXT I
6670 CLOSE
6680 RETURN
```

How to Buy or Sell Computer Equipment and Software



Computer Shopper is THE nationwide magazine for buying, selling and trading Micro and Mini-computer equipment and software. Each issue has over 60 pages full of bargains of new and used equipment.

You can save hundreds of dollars by getting the equipment you need from the hundreds of classified ads individuals place in Computer Shopper every month.

Now is the time for you to join over 20,000 other computer users who save time and money with a subscription to Computer Shopper.

Subscribe today and get your first issue and a classified ad absolutely FREE. Type or print your ad on a plain piece of paper and send it along with your subscription.

Just fill in the coupon or MasterCard or VISA holders can phone **TOLL FREE 1-800-327-9920** and start making your computer dollar go further today.

Cut out and mail to: **COMPUTER SHOPPER**
P.O. Box F138 • Titusville, FL 32780

Yes, I'll try Computer Shopper, I understand that if I'm not satisfied with my first issue I can receive a full refund and keep the first issue free.

☐ 1 year $10.00 ($30.00 in Canada)

☐ I have enclosed my free classified ad.

☐ I want to use my free ad later, send me a coupon.

NAME: ______________________

ADDRESS: ______________________

CITY: ______________________

STATE: ____________ ZIP: ____________

COMPUTER SHOPPER
P.O. Box F138 • Titusville, FL 32780
Telephone: 305-269-3211

✓212

✓See List of Advertisers on page 290

On monitors and cassette I/O.

Mod III Notes

John Ratzlaff
Mount Pisgah Academy
Candler, NC 28715

After several years on a TRS-80 Model I, I bought a Model III and chose to use a cassette recorder other than the CTR-80 or CTR-41.

I used a Hitachi TRQ-276 cassette recorder on my Model I, but the high speed cassette load would not work on my Model III.

I experimented with different tapes and cleaned and adjusted the record head, but found no improvement.

Another tape recorder of the same type worked perfectly. I concluded that when reading and writing tapes at high speed where high frequencies are much more important, the condition of the record/play head in the recorder is crucial.

Cassette Modifications

I made two modifications, published several times in TRS-80 literature, to my cassette recorder. A switch controls the cassette motor and a resistor permits hearing the sound even when the earphone plug is in place.

My recorder has the Cue/Review capability. You can hear the sound while in fast-forward or rewind modes, facilitating program location. My current library consists of three tapes containing dozens of programs, rather than many tapes containing one or two programs each. I identify programs by recording their names by voice with the built-in microphone.

"My current library consists of three tapes containing dozens of programs. . ."

Monitors

A machine-language monitor is indispensable for any machine-level work. Few monitor programs are designed for the Model III; so I tested my old monitors for the Model I. I had three available: T-Bug (Radio Shack), RSM-2D (Small Systems Software), and MON-3 (Hubert Howe). T-Bug is rather primitive, good for elementary use but lacking a disassembler and printer output.

All three monitors work to some extent on the Model III. When creating a machine-language tape T-Bug and RSM use their own cassette routines rather than ROM routines. Their tape output is at 500 baud regardless of the Model III speed. (They are really slightly faster than 500 baud; the 2.0 MHz Model III clock runs approximately 18 percent faster than the 1.7 MHz Model I clock.) Also, since the Model III uses a different port for cassette motor control, these monitors do not turn the cassette on or off and cannot create standard machine language tapes.

MON-3 uses ROM routines for all cassette operations and is useable on the Model III. It will create machine-language tapes at high speed (1500 baud).

The monitor-created machine-language tapes loaded properly with the System command, but the MON-3 program would not read any tapes with the L command. The placement of a Return (C9) into the RST28 and RST38 jump addresses at 400CH and 4012H created the problem; this section at the beginning of MON-3 can be replaced with NOPs.

Monitor programs advertised as ROM-independent do not use any ROM routines, avoiding difficulty with the different Model I ROMs. These programs are not hardware-independent, because they use the same cassette port and clock speed as the Model I. They *cannot* work on the Model III. A program using ROM input/output routines *will* work, since Radio Shack kept Model I I/O routine addresses on the Model III.

All other MON-3 commands work properly on the Model III. Even the Model III's repeating keyboard is active under MON-3, convenient for scanning slowly through disassemblies, ASCII, or hex dumps. ■

MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of 80 Microcomputing.

Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

- ☐ Address change only
- ☐ Extend subscription
- ☐ Enter new subscription
- ☐ 1 year \$24.97
- ☐ Payment enclosed (1 extra BONUS issue)
- ☐ Bill me later

Canadian \$27.97 US funds, Foreign surface \$44.97 US funds, Foreign air mail please inquire

80 MICROCOMPUTING

If you have no label handy, print OLD address here.

AFFIX LABEL

Name ____________ Call ________

Address ____________

City ________ State ________ Zip ________

print NEW address here:

Name ____________ Call ________

Address ____________

City ________ State ________ Zip ________

P.O. Box 981 ● Farmingdale NY 11737 ●

Isolate a specific element in a chemical solution with the Color Computer.

Colorful Titrations

James W. Wood
424 North Missouri
Atwood, IL 61913

Microcomputers are found in more schools and classrooms across the country every year. They are easy to apply to spelling and math drills, but their potential to simulate science problems should also be tapped.

A common lab assignment in high school chemistry courses is titration. Program Listing 1 is a titration simulation for the Color Computer. It prepares students for their lab assignment and lets them practice titration problems.

Actual titration procedures mix measured quantities of an acid whose strength is unknown with a base of known strength. The acid and base are dispensed from burettes which are graduated to the tenth of a milliliter. The chemicals are collected in a flask containing an indicator. (The indicator is one color when it is in a base and another color when it is in an acid.) The base and acid are added to one another until they are neutralized—the point when one drop of acid or base changes the indicator's color.

When the chemicals are neutralized, a simple calculation using the formula in Fig. 1 determines the strength of the acid.

In practice, students usually repeat the procedure several times before they perform a titration for a grade. Sometimes they forget to take the initial reading, and the chemicals are wasted. Other times they forget to let the air out of the burette tips. This will cause the amount of chemicals used to be read incorrectly which adds error to their experimental determination of the acid's normality.

Time and the cost of chemicals make it impossible to do enough titrations for every student to receive maximum benefit from the lab experiment. This simulation gives students a chance to become familiar with the procedures before they perform the actual experiment. The actual lab session can go more smoothly.

The simulation randomly picks acid and base values, stating only the value for the base. Burettes are drawn on the screen. The level of the liquids in the drawings lowers slightly to indicate that air is eliminated from the tips of the valves (see Photo 1).

Pressing A on the computer keyboard dispenses acid, and pressing B dispenses base. When the solution is neutral, the student can test the value of the normality of the acid by pressing the space bar on the keyboard. The program gives the true value of solution concentration and the percent of the student's error. And without wiping off the table or washing the flasks, the simulation is ready to begin again.

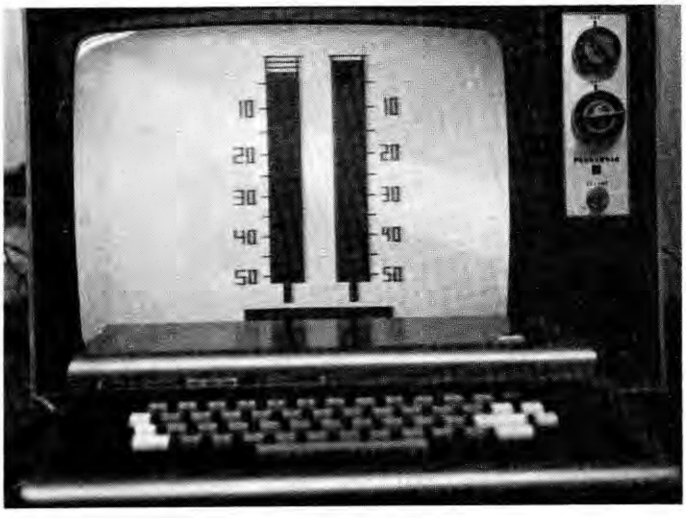


Photo 1

The Key Box
Color Computer

Unfortunately, the burettes in the simulation can only be read to the nearest milliliter. In the actual experiment students use burettes graduated to tenths of a milliliter. ■

$$N_{acid} = \frac{(V_{base\ final} - V_{base\ initial}) \cdot N_{base}}{(V_{acid\ final} - V_{acid\ initial})}$$

Fig. 1

0	Determines strength of acid and base
1-3	Instruction
10	Defines graphics screen
20	String to draw burette
30-60	Draws and colors burette
70	Draws container to collect acid and base
200-230	Calibrates burette
240-270	Makes extended calibrations on burette
291	Strings for numbers 2,3,4,5,0 on graphics screen
300-390	Draws numbers next to burette
450	Determines starting point for acid
460	Determines starting point for base
470	Records starting point
500-540	Loops to look for a request to dispense acid, dispense base, give answer or change the color of the indicator
10000	Graphics to lower acid
10010	Graphics to lower acid
20000	Graphics to lower base
20010	Determines if all the base is used
25000-26000	Changes the color of the indicator
30000-30030	Checks answer for acid's normality

Table 1. Line Functions

```
0 A=(4+RND(11))/10:B=(4+RND(11))/10
1 CLS:PRINT"WELCOME TO THE TITRATION LAB":PRINT:PRINT"THE ACID W
ILL BE ON THE LEFT":PRINT"THE BASE IS ON THE RIGHT":PRINT"PRESS
A TO RELEASE ACID":PRINT"PRESS B TO RELEASE BASE":PRINT"PRESS SP
ACE BAR WHEN READY TO    GIVE NORMALITY OF ACID"
2 PRINT"THE BASE'S NORMALITY IS";B:PRINT:PRINT"PRESS ANY KEY TO
CONTINUE"
3 K$=INKEY$:IFK$=""THEN3
10 PMODE3,1:PCLS:SCREEN1,1
20 X$="D155R10D15R5U15R10U155R2L29"
30 DRAW"BM100,0XX$;"
40 DRAW"BM150,0XX$;"
50 PAINT(102,2),6,8
60 PAINT(152,2),7,8
70 LINE(80,175)-(190,196),PSET,B
200 FORQ=3TO150STEP3
210 LINE(100,Q)-(125,Q),PSET
220 LINE(150,Q)-(175,Q),PSET
230 NEXTQ
240 FORQ=15TO150STEP15
250 LINE(95,Q)-(100,Q),PSET
260 LINE(175,Q)-(180,Q),PSET
270 NEXTQ
291 Y$="R5D10L5U10":M$="L5D5R5D5L5":N$="D5R5U5D10":O$="R5D5L5R5D
5L5":P$="R5D5L5D5R5"
300 DRAW"BM80,25D10":DRAW"BM84,25XY$;"
310 DRAW"BM190,25D10":DRAW"BM194,25XY$;"
320 DRAW"BM75,55XP$;":DRAW"BM84,55XY$;"
330 DRAW"BM186,55XP$;":DRAW"BM194,55XY$;"
340 DRAW"BM75,85XO$;":DRAW"BM84,85XY$;"
350 DRAW"BM186,85XO$;":DRAW"BM194,85XY$;"
360 DRAW"BM75,115XN$;":DRAW"BM84,115XY$;"
370 DRAW"BM186,115XN$;":DRAW"BM194,115XY$;"
380 DRAW"BM80,145XM$;":DRAW"BM84,145XY$;"
390 DRAW"BM191,145XM$;":DRAW"BM194,145XY$;"
450 C=RND(7):FORE=1TOC:PAINT(102,3*E-1),5,8:NEXTE
460 D=RND(7):FORE=1TOD:PAINT(152,3*E-1),5,8:NEXTE
470 F=C:G=D
500 A$=INKEY$
510 IFA$="A"THENGOSUB10000
520 IFA$="B"THENGOSUB20000
525 IFA-(C-F)>B*(D-G)THENGOSUB25000ELSEGOSUB26000
530 IFA$=" "THENGOTO30000
540 GOTO500
10000 C=C+1:PAINT(102,3*C-1),5,8
10010 IFC=51THEN10100
10050 RETURN
10100 CLS:PRINT"SORRY, YOU RAN OUT OF ACID":END
20000 D=D+1:PAINT(152,3*D-1),5,8
20010 IFD=51THEN20100
20050 RETURN
20100 CLS:PRINT"SORRY,YOU RAN OUT OF BASE":END
25000 PAINT(82,177),5,8:RETURN
26000 PAINT(82,177),7,8:RETURN
30000 CLS:INPUT"WHAT IS THE NORMALITY OF THE    ACID";AA
30010 PRINT"THE CORRECT ANSWER IS";A
30020 PRINT"YOUR ERROR IS";INT(ABS(AA-A)/A*100+.5);"%"
30030 INPUT"ANOTHER LAB";L$:IFLEFT$(L$,1)="Y"THENRUN
```

Program Listing 1

✓See List of Advertisers on page 290

Use this graphics technique in your game.

Polar Generator

Ken Webb
88 Presland Road
Ottawa, Ont.
Canada K1K 2C3

Six months ago I began planning a complex game, simulating the discovery of the northern polar regions. I designed a procedure to display on the screen a partial map of the arctic, and then, at the player's command, to give an expanded view of any desired section. This was all to be done in machine language for maximum speed.

During ship movement, water would be shown with ice conditions that would be consistent with the geographical location and time of year. A ship would be unable to get through any ice shown on the screen. The problem was how to generate up to 1024 (one for each screen location) random numbers in a fraction of a second.

```
                00810 ;*******************************************
*****
                00820 ;* CALLS ROM RANDOM NUMBER GENERATOR
    *
                00830 ;* AND DISPLAYS GRAPHIC EQUIVALENTS
    *
                00840 ;* TO ACCCESS ROUTINE:
    *
                00850 ;*       SYSTEM  /32620
    *
                00860 ;*(OR)  POKE 16526,108: POKE16527,127: X=USR
(0) *
                00870 ;*******************************************
*****
                00880 ;
7F6C DD21003C 00890           LD      IX,3C00H         ;SCREEN LOCA
TION
7F70 010004   00900           LD      BC,1024          ;SCREEN BYTE
 COUNTER
7F73 117F00   00910           LD      DE,127           ;GRAPHICS IN
CREMENT
              00915 ;MUMFORD'S ROM CALL
7F76 214000   00920 RND       LD      HL,64            ;UPPER LIMIT
 OF RANDOM #
7F79 CDCC14   00930           CALL    14CCH            ;ROM RANDOM
# ROUTINE
7F7C CD7F0A   00940           CALL    0A7FH            ;CONVERT TO
INTEGER
              00942 ;END OF MUMFORD'S ROM CALL
7F7F 19       00950           ADD     HL,DE            ;NOW 128 - 1
91
7F80 DD7500   00960           LD      (IX),L           ;DISPLAY GRA
PHICS CHAR.
7F83 DD23     00970           INC     IX               ;NEXT SCREEN
 LOCATION
7F85 0B       00980           DEC     BC               ;LOOP 1024 T
IMES
7F86 78       00990           LD      A,B
7F87 B1       01000           OR      C
7F88 20EC     01010           JR      NZ,RND
7F8A C9       01020           RET                      ;TO BASIC
0000          01030           END
00000 TOTAL ERRORS
```

Program Listing 1

Generating and Characters

I finally accomplished this by using RND(11000) in Basic to generate a random address in ROM which is then passed to a machine language routine. The routine takes the number at each sequential ROM address up to 1024, and adjusts it with Set and Reset based on the contents of a table. The table specifies ice conditions, and loads the resulting byte into screen memory. There are three types of ice, only two are randomly generated.

As a further example of the random method of generating numbers, use your editor/assembler, or a machine language monitor such as T-Bug to enter Program Listing 4. This program takes the contents of successive ROM locations, and converts them to X and Y coordinates which are then Set and Reset on the screen.

Listing 3 is a Basic driver program that defines the overall shape of the patterns created by the assembly/machine language routine. When you power-up, answer the memory size question with 32544.

I would like to describe one final machine language method of wringing random numbers out of your TRS-80. The R register is continuously cycling through every integer between 0 and 255 (just why it does this I can't say).

Program Listing 6 gets the contents of the R register 255 times, and prints a graphic equivalent of each of these numbers on the screen. You should get a different pattern each time you run this routine, but each pattern will repeat itself at a regular interval. On my TRS-80 Model I the interval is 64, so each line is the same as the one above it. The routine prints every ninth graphics character. By adding a delay loop, you should be able to get it to print every tenth character, or every twenty-third, etc. So Program Listing 6 does not generate very good random numbers because we easily pre-

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM

FOR TRS-80*

[Reprint of June 1980 Review, *80 Microcomputing*]

Now the widely acclaimed JPC Cassette System is available for your TRS-80* computer. The price is only $90.00

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 290

dict the interval between one number and the next. You may, however, be able to use something like it in a very restricted application. It is short and very fast.

Program Listing 7 adds an extra random element, an irregular delay. This routine rolls two dice, one at a time. First hit the 1 key, then the 2, and repeat each time you want to roll. The resulting numbers are unpredictable (although there are more fives and ones than I would expect).

You may want to combine two of the methods described in this article. In a machine language program, to create part of the surface of an alien planet, I use both the numbers from some arbitrary section of ROM and the R register. One simple device would be to And/Or the contents of a ROM location with the R register. The result may less repetitive than using either method alone.

If you need a small number of good random numbers, or you are not too concerned with speed, then use the built-in ROM routine. But if you want speed, try linking the random method with use of the R register. I doubt that you will find a faster way of churning out random numbers.■

```
25 'RANDOM NUMBER GENERATOR
30 D# = 2 [ 31
40 Y# = 568731   'ANY NUMBER BETWEEN 100000 AND 999999
50 INPUT"LOWER LIMIT, UPPER LIMIT";M,N   'LIMITS OF RANDOM NUMBE
RS
60 X# = (15625 * Y# + 22221) / D#
70 Y# = (X# - INT(X#)) * D#
80 X = M + INT(((N - M + 1) * Y#) / D#)
90 PRINT X;
100 GOTO 60
```

Program Listing 2

```
30 'CONVERTS CONTENTS OF ROM LOCATIONS TO DISPLAYABLE GRAPHICS C
HARACTERS
40 FOR X = 0 TO 12287
50      PRINT CHR$((PEEK(X) AND 63) + 128);
60 NEXT
```

Program Listing 3

Program Listing 4

```
               00100 ;********************************
               00110 ;* RANDOM PATTERN GENERATOR     *
               00115 ;* BY KEN WEBB                  *
               00120 ;* ALL RIGHTS RESERVED          *
               00130 ;********************************
7F21           00140 START   EQU     7F21H          ;CHANGE TO R
ELOCATE
               00150 ;X   HORIZONTAL POSITION
               00160 ;XLIMIT * XADDS + XCONST MUST BE >0 AND <127
7F21           00170 XLIMIT  EQU     START          ;UPPER LIMIT
 OF RANDOM #
7F22           00180 XADDS   EQU     START+1        ;# OF RANDOM
 #'S TO ADD
7F23           00190 XCONST  EQU     START+2        ;ADDED TO FI
NAL RANDOM #
               00200 ;Y   VERTICAL POSITION
               00210 ;YLIMIT * YADDS + YCONST MUST BE >0 AND <47
7F24           00220 YLIMIT  EQU     START+3
7F25           00230 YADDS   EQU     START+4
7F26           00240 YCONST  EQU     START+5
7F27           00250 ROMLOC  EQU     START+6        ;LOCATION IN
 ROM
7F29           00260 BITCNT  EQU     START+8        ;# OF BITS T
O SET/RESET
7F2B           00270 STRST   EQU     START+10       ;0=SET  1=RE
SET
7F2C           00280 XVALUE  EQU     START+11       ;SUM OF RAND
OM #'S
7F2D           00290 YVALUE  EQU     START+12
7F2E           00300         ORG     START+13
7F2E ED4B297F  00310         LD      BC,(BITCNT)    ;GET BIT COU
NTER
7F32 C5        00320         PUSH    BC             ;SAVE BIT CO
UNTER
7F33 3A237F    00330         LD      A,(XCONST)     ;GET CONSTAN
TS
7F36 67        00340         LD      H,A
7F37 3A267F    00350         LD      A,(YCONST)
7F3A 6F        00360         LD      L,A
7F3B E5        00370         PUSH    HL             ;SAVE CONSTA
NTS
7F3C FD2A277F  00380 STARTX  LD      IY,(ROMLOC)    ;GET ROM CON
TENTS
7F40 3A217F    00390         LD      A,(XLIMIT)     ;GET UPPER L
IMIT
7F43 CD987F    00400         CALL    LMT            ;CREATE MASK

7F46 3A227F    00410         LD      A,(XADDS)      ;GET # OF RA
NDOMS TO ADD
7F49 47        00420         LD      B,A
7F4A 3E00      00430         LD      A,0            ;INITIALIZE
SUM TO 0
7F4C 322C7F    00440         LD      (XVALUE),A
7F4F CDA27F    00450 LOOP1   CALL    ROMBYT         ;GET A SINGL
E RANDOM #
7F52 3A2C7F    00460         LD      A,(XVALUE)     ;GET OLD SUM

7F55 81        00470         ADD     A,C            ;ADD RANDOM
# TO OLD SUM
7F56 322C7F    00480         LD      (XVALUE),A     ;STORE NEW S
UM
7F59 10F4      00490         DJNZ    LOOP1          ;LOOP XADDS
TIMES
7F5B 3A247F    00500 STARTY  LD      A,(YLIMIT)     ;SAME AS STA
RTX
7F5E CD987F    00510         CALL    LMT
7F61 3A257F    00520         LD      A,(YADDS)
7F64 47        00530         LD      B,A
7F65 3E00      00540         LD      A,0
7F67 322D7F    00550         LD      (YVALUE),A
7F6A CDA27F    00560 LOOP2   CALL    ROMBYT         ;SAME AS LOO
P1
7F6D 3A2D7F    00570         LD      A,(YVALUE)
7F70 81        00580         ADD     A,C
7F71 322D7F    00590         LD      (YVALUE),A
7F74 10F4      00600         DJNZ    LOOP2
7F76 FD22277F  00610 SETT    LD      (ROMLOC),IY    ;SAVE NEW RO
M LOCATION
7F7A E1        00620         POP     HL             ;GET CONSTAN
TS
7F7B E5        00630         PUSH    HL
7F7C 3A2C7F    00640         LD      A,(XVALUE)     ;GET RANDOM
SUM
7F7F 84        00650         ADD     A,H            ;ADD CONSTAN
T TO SUM
7F80 67        00660         LD      H,A            ;FINAL X COO
RDINATE
7F81 3A2D7F    00670         LD      A,(YVALUE)
7F84 85        00680         ADD     A,L
7F85 6F        00690         LD      L,A            ;FINAL Y COO
RDINATE
7F86 3A2B7F    00700         LD      A,(STRST)      ;GET SET/RES
ET FLAG
7F89 CDAF7F    00710         CALL    PIXEL          ;SET/RESET A
 BIT
7F8C E1        00720         POP     HL
7F8D C1        00730         POP     BC             ;GET BIT COU
NTER
7F8E 0B        00740         DEC     BC             ;BIT COUNTER
 = 0 ?
7F8F 78        00750         LD      A,B
7F90 B1        00760         OR      C
7F91 C5        00770         PUSH    BC
7F92 E5        00780         PUSH    HL
7F93 20A7      00790         JR      NZ,STARTX      ;GO IF COUNT
ER > 0
7F95 E1        00800         POP     HL             ;ELSE FIX ST
ACK
7F96 C1        00810         POP     BC
7F97 C9        00820         RET                    ;RETURN TO B
ASIC
7F98 57        00830 LMT     LD      D,A            ;SAVE UPPER
LIMIT
7F99 1EFF      00840         LD      E,0FFH         ;MASK, BINAR
Y 11111111
7F9B CB27      00850 SHFT    SLA     A              ;TEST HIGH B
IT OF LIMIT
7F9D D8        00860         RET     C              ;GO IF IT'S
SET
7F9E CB3B      00870         SRL     E              ;ELSE ADJUST
 MASK
7FA0 18F9      00880         JR      SHFT
7FA2 FD23      00890 ROMBYT  INC     IY             ;NEXT ROM LO
CATION
7FA4 FD7E00    00900         LD      A,(IY)         ;GET ROM CON
TENTS
7FA7 A3        00910         AND     E              ;RESET HIGH
BITS
7FA8 BA        00920         CP      D              ;COMPARE WIT
H LIMIT
7FA9 F2A27F    00930         JP      P,ROMBYT       ;GO IF >= UP
PER LIMIT
7FAC 3C        00940         INC     A
7FAD 4F        00950        \LD      C,A
7FAE C9        00960         RET
               00970 ;******************************************
*
               00980 ;* SET/RESET ROUTINE BY WILLIAM BARDEN, JR.
*
               00990 ;* 80 MICROCOMPUTING, AUG 1980, P.25
*
               01000 ;******************************************
*
7FAF F5        01010 PIXEL   PUSH    AF             ;SAVE SET/RE
SET FLAG
7FB0 5C        01020         LD      E,H            ;X
7FB1 7D        01030         LD      A,L            ;Y
7FB2 CB3B      01040         SRL     E              ;GET CHAR PO
SITION (0-63) IN E
7FB4 1600      01050         LD      D,0            ;SET COL# TO
 0
7FB6 3001      01060         JR      NC,SET10       ;GO IF COL#=
0
7FB8 14        01070         INC     D              ;COL#=1
```

Program Listing 4 continues

Program Listing 4 continued

```
7FB9 06FF      01080 SET10    LD      B,0FFH          ;-1 TO B
7FBB 04        01090 SET20    INC     B               ;BUMP QUOTIE
NT IN B=LINE#
7FBC D603      01100          SUB     3               ;SUCCESSIVE
SUBT FOR /3
7FBE F2BB7F    01110          JP      P,SET20         ;GO IF NOT N
EGATIVE
7FC1 C603      01120 SET25    ADD     A,3             ;ADD BACK FO
R REMAINDER=ROW#
7FC3 07        01130          RLCA                    ;(ROW#)*2
7FC4 82        01140          ADD     A,D             ;(ROW#)*2+CO
L#=BIT POS
7FC5 4F        01150 SET27    LD      C,A             ;SAVE BIT PO
S IN C
7FC6 68        01160          LD      L,B             ;LINE #
7FC7 2600      01170          LD      H,0             ;NOW IN HL
7FC9 0606      01180          LD      B,6             ;SHIFT COUNT

7FCB 29        01190 SET30    ADD     HL,HL           ;MULTIPLY LI
NE#*64
7FCC 10FD      01200          DJNZ    SET30           ;LOOP TIL DO
NE
7FCE 1600      01210 SET32    LD      D,0             ;DE NOW HAS
CHAR POS
7FD0 19        01220          ADD     HL,DE           ;(LINE#)*64+
CHAR POS IN HL
7FD1 11003C    01230          LD      DE,3C00H        ;START OF VI
DEO
7FD4 19        01240 SET34    ADD     HL,DE           ;(LINE#)*64+
CHAR POS+3C00H
7FD5 0600      01250          LD      B,0             ;BC NOW HAS
BIT POS
7FD7 F1        01260          POP     AF              ;GET SET/RES
ET FLAG
7FD8 B7        01270          OR      A               ;TEST FLAG
7FD9 200C      01280          JR      NZ,RESET        ;GO IF RESET
7FDB DD21F37F  01290          LD      IX,MASK         ;START OF MA
SK TABLE
7FDF DD09      01300          ADD     IX,BC           ;POINT TO MA
SK
7FE1 7E        01310          LD      A,(HL)          ;LOAD PIXEL
7FE2 DDB600    01320          OR      (IX)            ;SET PIXEL
7FE5 77        01330 SET36    LD      (HL),A          ;STORE IN VI
DEO
7FE6 C9        01340          RET                     ;RETURN
7FE7 DD21F97F  01350 RESET    LD      IX,MASK1        ;RESET MASK
TABLE
7FEB DD09      01360          ADD     IX,BC           ;POINT TO MA
SK
7FED 7E        01370          LD      A,(HL)          ;LOAD PIXEL
7FEE DDA600    01380          AND     (IX)            ;RESET PIXEL

7FF1 18F2      01390          JR      SET36           ;GO TO STORE
, RETURN
7FF3 81        01400 MASK     DEFB    81H             ;MASK TABLE
7FF4 82        01410          DEFB    82H
7FF5 84        01420          DEFB    84H
7FF6 88        01430          DEFB    88H
7FF7 90        01440          DEFB    90H
7FF8 A0        01450          DEFB    0A0H
7FF9 FE        01460 MASK1    DEFB    0FEH
7FFA FD        01470          DEFB    0FDH
7FFB FB        01480          DEFB    0FBH
7FFC F7        01490          DEFB    0F7H
7FFD EF        01500          DEFB    0EFH
7FFE DF        01510          DEFB    0DFH
0000           01520          END
00000 TOTAL ERRORS
```

```
5 'DRIVER PROGRAM FOR ASSEMBLY LISTING 2
6 ' BY KEN WEBB
7 'ALL RIGHTS RESERVED
10 FORX=15360TO16383:POKEX,191:NEXT   'WHITE OUT SCREEN
20 POKE16526,46:POKE16527,127
     '=7F2EH
  MACHINE LANGUAGE ENT
RY POINT
25 READM
  '# OF DATA SETS
27 FORY=1TOM
30 P=32545:FORX=0TO10:READA:POKEP+X,RND(A):NEXT   '=7F21H TO 7F2
BH  PASSES DATA TO MACHINE LANGUAGE ROUTINE
40 X=USR(0)
45 NEXTY
50 RESTORE:GOTO25
200 DATA 2
202 '
205 '#-> 0   1   2   3   4   5   6   7   8   9   10
210 DATA 63,  2,  0, 23,  2,  0,  0,  0,  0,  8,  1
220 DATA 10,  3, 40,  4,  3, 17,  0,  0,  0,  4,  0
490 '
500 'DESCRIPTION OF DATA ELEMENTS (SEE ALSO FIRST LINES OF ASSEM
BLY LISTING 2)
590 '#: ASSEMBLY NAME:  RANGE:
600 '0  XLIMIT          >0      <127
610 '1  XADDS           >0      <=127/XLIMIT
620 '2  XCONST          >=0     <127-XLIMIT*XADDS
630 '3  YLIMIT          >0      <47
640 '4  YADDS           >0      <=47/YLIMIT
650 '5  YCONST          >=0     <47-YLIMIT*YADDS
659 'ROMLOC
660 '6  LOW BYTE        >=0     <=255
670 '7  HIGH BYTE       >=0     <=ABOUT 8
679 'BITCNT
680 '8  LOW BYTE        >=0     <=255
690 '9  HIGH BYTE       >=0     <=ABOUT 8
700 '10 STRST           0=SET   1=RESET
1000 '
1010 'TO CHANGE THE TYPE OF PATTERN:
1020 '   -CHANGE "RND(A)" TO "A" IN LINE 30
1030 '   -CHANGE THE NUMBER OF DATA SETS (MORE DATA STATEMENTS)
1040 '   -ALTER THE DATA, BUT KEEP IT WITHIN THE RANGES INDICATE
D IN LINES 600 TO 700
```

Program Listing 5

```
                ;*******************************************
*****
                ;* PRINTS GRAPHIC EQUIVALENTS OF THE R REGIS
TER *
                ;* TO ACCESS ROUUTINE:
    *
                ;*        SYSTEM  /32512
    *
                ;*(OR)  POKE 16526,0: POKE 16527,127: X=USR(
0)  *
                ;*******************************************
*****
                ;
7F00                        ORG     7F00H
7F00 21003C                 LD      HL,3C00H        ;CLEAR SCREE
N
7F03 11013C                 LD      DE,3C01H
7F06 01FF03                 LD      BC,1023
7F09 3680                   LD      (HL),128
7F0B EDB0                   LDIR
7F0D 21003C                 LD      HL,3C00H        ;SCREEN LOCA
TION
7F10 06FF                   LD      B,255           ;LOOP COUNTE
R
7F12 ED5F           RLOOP   LD      A,R             ;GET CONTENT
S OF R REG.
7F14 CBFF                   SET     7,A             ;CONVERT TO
GRAPHICS
7F16 CBB7                   RES     6,A
7F18 77                     LD      (HL),A          ;DISPLAY GRA
PHICS CHAR.
7F19 23                     INC     HL              ;NEXT SCREEN
 LOCATION
7F1A 10F6                   DJNZ    RLOOP           ;LOOP 512 TI
MES
7F1C 21403E                 LD      HL,3E40H        ;MOVE CURSOR
 LOCATION
7F1F 222040                 LD      (4020H),HL      ;TO LOWER SC
REEN
7F22 C3191A                 JP      1A19H           ;RETURN TO "
READY"
                ;
```

Program Listing 6

```
                ;*******************************************
*****
                ;* SIMULATES DICE ROLL USING R REGISTER
    *
                ;* TO ACCCESS ROUTINE:
    *
                ;*        SYSTEM  /32549
    *
                ;*(OR)  POKE 16526,37: POKE 16527,127: X=USR
(0) *
                ;*******************************************
*****
                ;
7F25 21003C                 LD      HL,3C00H        ;CLEAR SCREE
N
7F28 11013C                 LD      DE,3C01H
7F2B 01FF03                 LD      BC,1023
7F2E 3680                   LD      (HL),128
7F30 EDB0                   LDIR
7F32 DD21DE3D               LD      IX,3C00H+478    ;CENTER OF S
CREEN
7F36 0E30                   LD      C,30H           ;ASCII NUMBE
R CODE
7F38 3A1038         DIE1    LD      A,(3810H)       ;TEST KEYBOA
RD
7F3B FE02                   CP      02              ;IS IT "1" ?

7F3D 20F9                   JR      NZ,DIE1         ;LOOP TIL "1
" HIT
7F3F DD360480               LD      (IX+4),128      ;BLANK 2ND D
IE
7F43 CD5A7F                 CALL    RANDOM          ;GET A RANDO
M #
7F46 81                     ADD     A,C             ;MAKE IT AN
ASCII #
7F47 DD7700                 LD      (IX),A          ;DISPLAY ON
SCREEN
7F4A 3A1038         DIE2    LD      A,(3810H)       ;TEST KEYBOA
RD
7F4D FE04                   CP      04              ;IS IT "2" ?

7F4F 20F9                   JR      NZ,DIE2         ;LOOP TIL "2
" HIT
7F51 CD5A7F                 CALL    RANDOM          ;GET A RANDO
M #
7F54 81                     ADD     A,C             ;MAKE IT AN
ASCII #
7F55 DD7704                 LD      (IX+4),A        ;DISPLAY ON
SCREEN
7F58 18DE                   JR      DIE1
7F5A ED5F           RANDOM  LD      A,R             ;GET CONTENT
S OF R REG.
7F5C 47                     LD      B,A             ;RANDOM DELA
Y COUNTER
7F5D E607                   AND     7               ;MAKE IT 7 O
R LESS
7F5F FE07                   CP      7               ;TRY AGAIN I
F IT'S 7
7F61 2805                   JR      Z,TRYAGN
7F63 FE00                   CP      0               ;TRY AGAIN I
F IT'S 0
7F65 2801                   JR      Z,TRYAGN
7F67 C9                     RET
7F68 10FE           TRYAGN  DJNZ    $               ;LOOP RANDOM
 # OF TIMES
7F6A 18EE                   JR      RANDOM          ;TRY AGAIN
```

Program Listing 7

The Key Box

Basic Level II
Model I or III
4K RAM

Use your TRS-80 to implement the trapezoidal rule.

The Bemusing Triangle

C. Brian Honess
22 Shaftesbury Lane
Columbia, SC 29209

Assume we have a simple linear figure (Fig. 1), and we want to find the area under the line. In other words, we just want to find the area of a 6 by 3 right triangle. This is easily done by multiplying the base by the height, and dividing by 2:

$$A = \frac{3 \cdot 6}{2} = \frac{18}{2} = 9$$

But there is another way to find the area. We can find the height at point A, the height at point B, average these two values, and multiply by the width (Fig. 2):

$$A = \frac{0 + 3}{2} \cdot 6 = 1.5 \cdot 6 = 9$$

What we're really doing is finding the area under the dotted line (Fig. 3). We're sort of "folding" area A over into area B and thereby changing the problem from one where we want to find a triangular area to one where we calculate the area of a rectangle.

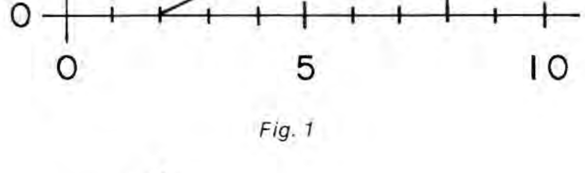


Fig. 1

We can use this idea to find the areas of some shapes other than triangles. In each case, the method will involve finding the average height of a figure (by taking half of the sum of the height on the left side and the height on the right side), and then multiplying by its width. Examples should make this clear (see Fig. 4).

$$\text{Area} = \frac{1 + 1.5}{2} \cdot 1 = \frac{2.5}{2} \cdot 1 = 1.25$$

$$\text{Area} = \frac{4 + 6}{2} \cdot 2 = \frac{10}{2} \cdot 2 = 5 \cdot 2 = 10$$

Suppose we have a function, plotted as in Fig. 5, and want to know the area between points A and B. Y could be almost any function of X, so that the curve has as many twists and turns as you like.

To find the area under the curve between points A and B, section the distance between A and B into many little rectangles and find the area of each, then add all the areas together (see Fig. 6). Of course the areas won't be exact if the curve isn't straight at any particular point, but by making the rectangles

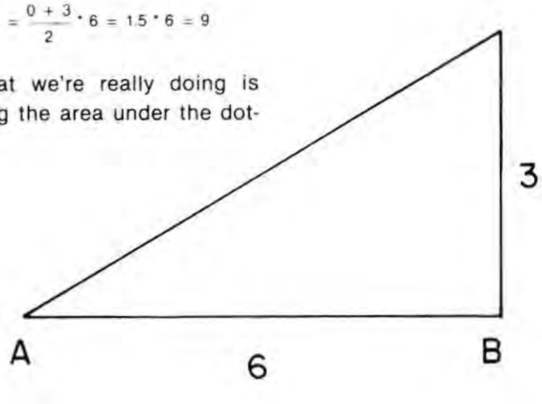


Fig. 2

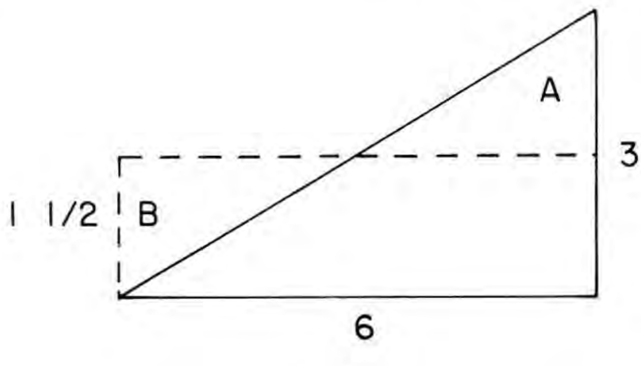


Fig. 3

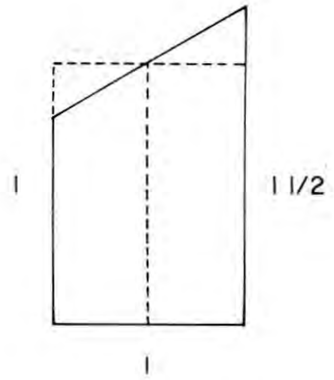


Fig. 4a

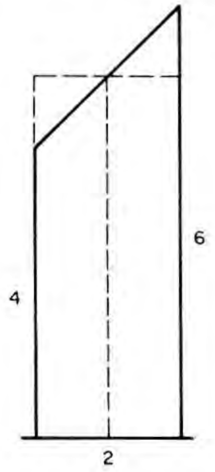


Fig. 4b

narrow enough, we can cut down the error. We could reduce the error in the difference in area between the two shaded areas by reducing the width of the interval.

Now consider a problem involving the Power Function, where: $y = x^2$.

Let's look at all values of y for values of x from 0 to 10, inclusive. We could plot the curve and thereby visualize the area we'd like to find (Fig. 7).

We'll begin with some rather large width increments, say, one unit, giving us 10 rectangles. What we'll be doing is shown on an enlarged section of just the

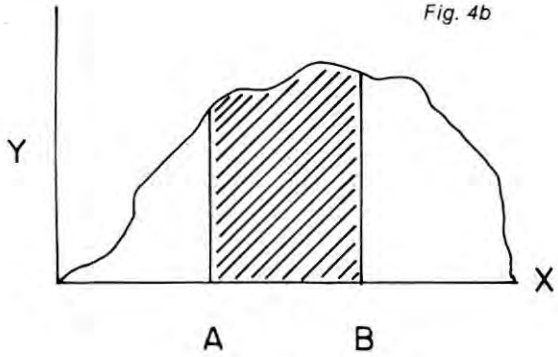


Fig. 5

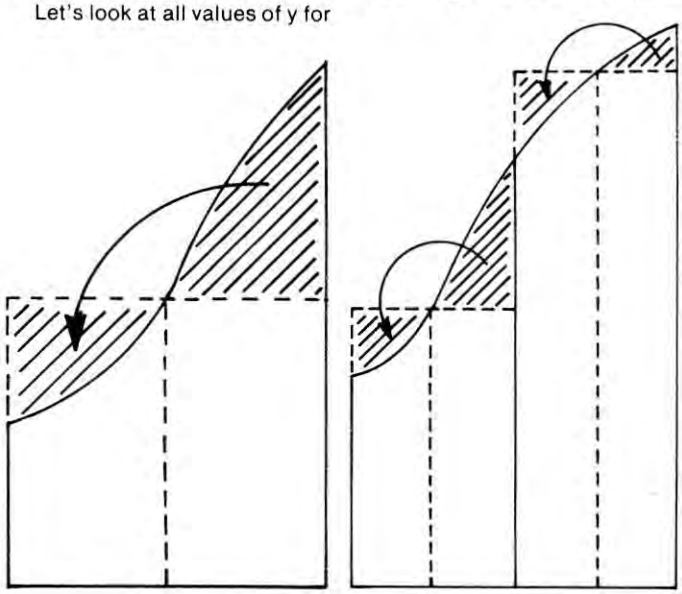


Fig. 6

See List of Advertisers on page 290

left side of the curve (Fig. 8).

I've labeled the distance from the X axis to the curve where X = 1, "h_1" (which stands for "height one"), and the distance from the X axis to the curve where X = 2, "h_2", etc. This is a special case situation, since we want the area all the way down to the origin, so $h_0 = 0$, and $h_{10} = 100$. Each of the rectangles has a width of 1: w = 1.

The Trapezoidal Rule

The preceding information has been preparation for the presentation of what is known as the Trapezoidal Rule, where n is the number of rectangles, or trapezoids, you want to use (Example 3).

The first height (h_0) and the last height (h_n) are used only once, while the others are each used twice. This allows us to get that big formula down to a more manageable size (Example 4).

We could simplify it further by putting a summation symbol inside the inner pair of parentheses, but that might be more confusing than helpful.

Soon we'll write a Basic program to find the area under the curve, but first let's work out the answer so we have something with which to compare the computer results (see Example 5).

Note: This is an approximation of the area under the curve using numerical integration and the Trapezoidal Rule. If you know some calculus you can easily verify that the actual area is 333.333, since:

$$A = \int_0^{10} x^2\,dx = \left.\frac{x^3}{3}\right]_0^{10} = \frac{1000}{3} = 333.333$$

If you don't know calculus, you'll have to take my word for it, because I'd rather not get into that any deeper!

Knowing that the answer should be 333.333, let's code a simple program and see if we get it (see Example 6).

Lines 40 and 50 calculate the limits of the sums of the values of Y for the second through the next to last values of X. If my lower limit of X (L) is zero, and the upper limit of X (U) is 10, then A and B will be 1 and 9 respectively. Line 60 initializes the sum (SM) to 0, and then the 70 to 100 loop forms the sum. I had to put the function (in terms of Y and X) in line 80; if you use the program for other functions, you'll need to change line 80.

Line 110 then calculates the area. We had to find the Y value for the lower limit of X (L↑2) and for the upper limit of X (U↑2) and these two terms would also have to be changed if you wanted to find the area for some other function. The 2 * SM term would not have to be changed, nor would the W / 2 term, since they are a part of the Trapezoidal Rule formula.

Run the program with the values of L, U and W, 0, 10 and 1 respectively, and you'll get an area of 335, which is exactly what we suspected. Intuition tells us that if we make the interval smaller, we could expect an answer closer to the answer found by calculus, 333.333. Let's cut the interval width to ½, ¼, etc. I ran it four times using the data shown, and got these results:

L	U	W	Area
0	10	1	335.
0	10	.5	333.75
0	10	.25	333.438
0	10	.2	333.4

Enjoy playing with this simple program. See what you can do with the power function for values of X from 0 to 25, etc., then try some situations where the lower limit isn't zero—say a case of X going from 3 to 16.

Simpson's Rule

A second method for finding the approximate area bounded

$$\text{Area} = \left(\frac{h_0 + h_1}{2} \cdot w\right) + \left(\frac{h_1 + h_2}{2} \cdot w\right) + \left(\frac{h_2 + h_3}{2} \cdot w\right) + \ldots\ldots$$

$$\ldots\ldots + \left(\frac{h_{n-1} + h_n}{2} \cdot w\right)$$

Example 3

$$\text{Area} = \frac{h}{2}\,(h_0 + 2h_1 + 2h_2 + 2h_3 + \ldots + 2h_{n-1} + h_n)$$

$$= \frac{h}{2}\,(h_0 + 2\,(h_1 + h_2 + h_3 + \ldots + h_{n-1}) + h_n)$$

Example 4

$$\text{Area} = \frac{h}{2}\,(h_0 + 2\,(h_1 + h_2 + \ldots + h_{n-1}) + h_n)$$

$$= \frac{1}{2}\,(0 + 2\,(1^2 + 2^2 + 3^2 + \ldots + 9^2) + 10^2)$$

$$= \frac{1}{2}\,(0 + 2\,(285) + 100) = 670\,/\,2 = \underline{\underline{335}}$$

Example 5

```
10 INPUT "WHAT IS LOWER LIMIT OF X"; L
20 INPUT "WHAT IS UPPER LIMIT OF X"; U
30 INPUT "WHAT INTERVAL WIDTH TO USE"; W
40 A = L + W
50 B = U - W
60 SM = 0
70 FOR X = A TO B STEP W
80 Y = X↑2
90 SM = SM + Y
100 NEXT X
110 AR = W/2 * (L↑2 + 2 * SM + U↑2)
120 PRINT AR
130 END
```

Example 6

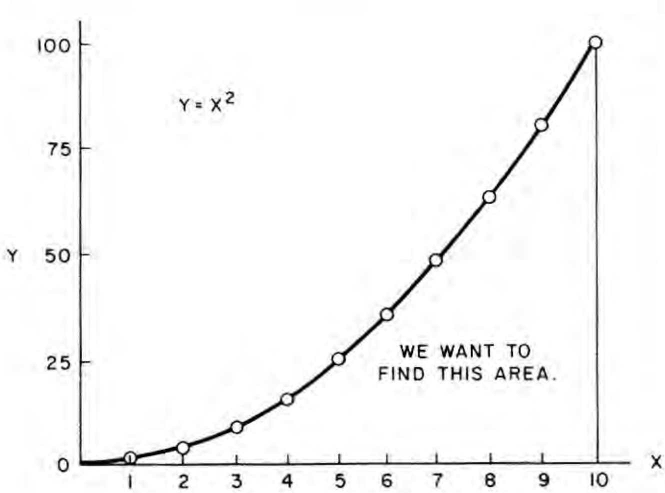


Fig. 7

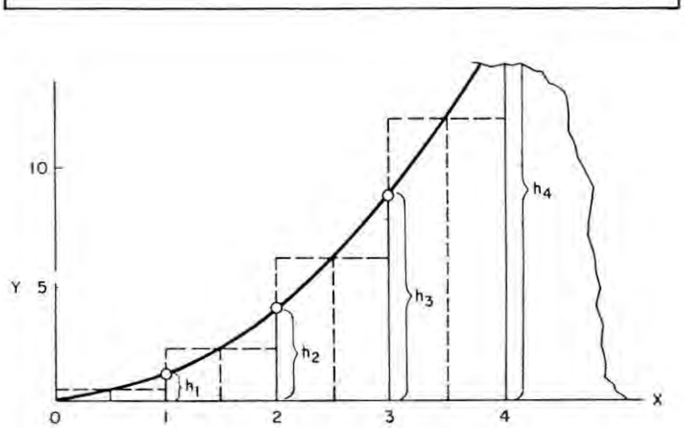


Fig. 8

See List of Advertisers on page 290

by a curve is called Simpson's Rule. It provides a little more accuracy in many cases. I'll present the formula without derivation, although you may want to play with it and see if you can discover how it was derived (Example 7).

We smooth things out a little by finding three heights for each term, and therefore dividing the interval width by three. We count the first and last height just once, and alternately multiplying the other terms by 2 and 4 (Example 8).

The function is in lines 90 and two places in line 160. Again we're finding the sum (SM) in the 80 to 150 loop, and then calculating and printing the area in lines 160 and 170. This time there is more going on inside the loop. Look at the formula: The coefficient for h_1 is 4. This is the first value needed inside the loop. After that we alternate between 2 and 4. Line 70 initializes this coefficient to 4, and lines 110 through 140 make it alternate between 4 and 2. Manually go through a few iterations of the loop until you get the idea.

The Trapezoidal Rule gave a value of 335 for the area between the X values of 0 and 10. If you try the same thing with Simpson's Rule, you get 333.333 which is also the answer we derived using calculus. The actual area under the curve for values of X between 0 and 25 is 5208.333. I tried the Trapezoidal Rule program with an interval of 1 and got 5212.50. Not bad! The Simpson's Rule program doesn't get too close with an interval width of 1, but if we reduce width to .25, it comes up with the exact answer.

Normal Distribution

Let's turn our attention to a much more interesting curve, the Normal Distribution (Fig. 9). This normal, or bell shaped curve, is used to approximate the distribution of a large number of things—from I.Q. to the life of lightbulbs. Basically it shows a clustering of values around the mean; as you get farther away from the mean, there is less and less likelihood of those values appearing.

If we set the area under the curve equal to 1, we can make probability statements about anything we assume is normally distributed. Tables are available showing the area under the curve for various x values. The area between a point one standard deviation below the mean and one standard deviation above the mean is about 68.27 percent of the area (Fig. 10). Virtually all the area lies between −3 standard deviations and +3 standard deviations from the mean.

To illustrate this concept, suppose the average life of a lightbulb is 2,000 hours, with a standard deviation of 200 hours. Consider the distribution of lightbulb life to be normal. We can then make the following (and many other) statements:

- 68.27 percent of the bulbs will last between 1,800 and 2,200 hours before burning out.
- 34.13 percent of the bulbs will last between 1,800 and 2,000 hours before burning out.
- Half the bulbs will last more than the average life of 2,000 hours.
- 13.59 percent of the bulbs will last between 2,200 and 2,400 hours.

This is all very nice, but we will rarely find a problem that involves just those intervals. We need to be able to find the area

$$\text{Area} = \frac{W}{3}(h_0 + 4h_1 + 2h_2 + 4h_3 + 2h_4 + \ldots + h_n)$$

Example 7

```
10 INPUT "WHAT IS LOWER LIMIT OF X"; L
20 INPUT "WHAT IS UPPER LIMIT OF X"; U
30 INPUT "WHAT INTERVAL WIDTH TO USE"; W
40 A = L + W
50 B = U - W
60 SM = 0
70 I = 4
80 FOR X = A TO B STEP W
90 Y = X ↑ 2
100 SM = SM + I * Y
110 IF I = 4 THEN 140
120 I = 4
130 GOTO 150
140 I = 2
150 NEXT X
160 AR = W/3 * (L ↑ 2 + SM + U ↑ 2)
170 PRINT AR
180 END
```

Example 8

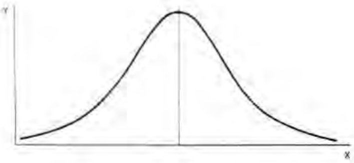


Fig. 9

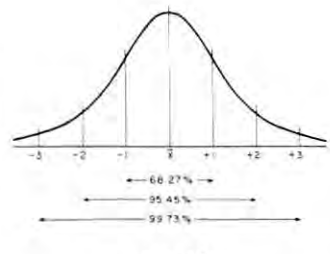


Fig. 10

under the curve between any two points on the X axis.

We worked with a simple power function in the last two example programs ($y = x^2$), but now the equation for the normal distribution is more formidable (see Example 9).

Assume a mean of 0 and a standard deviation of 1: this produces the Standard Normal Distribution, which is what the tables are produced for. With tables for the standard normal distribution, you can adjust the entries for any values of mean and standard deviation.

$$y = \frac{1}{\sqrt{2\pi}}\, e^{-X^2/2}$$

This short program will check what we've done thus far, using the values from the tables in most statistics books.

```
10 PI = 3.14159
20 E = 2.71828
30 T = 1 / SQR(2 * PI)
40 INPUT "WHAT VALUE OF X"; U
50 Y = T * E↑(- U↑2 / 2)
60 PRINT Y
```

We're reading in an X value, and calculating Y (the height of a line drawn from the X axis to the curve) for that particular value of X. We're not finding the area yet! (See Fig. 11.)

The following are answers the computer calculated, compared with answers from the statistical table:

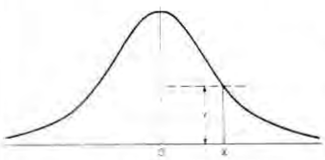


Fig. 11

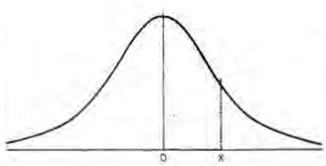


Fig. 12

X	Actual Y	Computed Y
0	.3989	.398942
0.5	.3521	.352065
1	.2420	.241971
1.5	.1295	.129518
2	.0540	.0539911
3	.0044	.00443186

You'll notice that in line 30 I've calculated T (which stands for Term) which is the reciprocal of the square root of 2π.

Keep the first four lines of our testing program and add code to find the area between the mean (0) and the value of X read in by line 40 (Fig. 12).

Since the curve is symmetrical, we don't have to bother with negative values of X—we can just fold over the area for a corresponding positive X.

Trapezoidal Rule

We'll use the Trapezoidal Rule to find the area under the curve. Assume that the lower limit of X is always zero; we already have input the upper limit value. We'll have to decide on the interval width to use, set up limits for the summation process, and then initialize our summation variable (SM) to zero. This is the program:

```
10 PI = 3.14159
20 E = 2.71828
30 T = 1 / SQR(2*PI)
40 INPUT "WHAT VALUE OF X"; U
50 INPUT "WHAT INTERVAL WIDTH"; W
60 A = 0 + W
70 B = U - W
80 SM = 0
90
```

If you later want to change the program to find areas where the lower limit isn't zero, add an input to enter the lower limit (L), and then change line 60 to read 60 A = L + W. Continuing the program to incorporate the Trapezoidal Rule from our first example:

```
90 FOR X = A TO B STEP W
100 Y = T * E↑(-X↑2 / 2)
110 SM = SM + Y
120 NEXT X
130 AL = T * E↑(-L↑2 / 2)
```

$$y = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}(X - \overline{X})^2/\sigma^2}$$

where: σ = the standard deviation
π = 3.14159
e = 2.71828
$\overline{X}$ = the arithmetic mean
X = the value of X for which you want to find the corresponding value of Y

Example 9

ALCOR PASCAL

AT LAST .....
A COMPLETE PROFESSIONAL QUALITY PASCAL FOR THE TRS80

Pascal Features

1. A complete Jensen and Wirth Standard Pascal
2. Produces compact efficient code that executes 10-20 times faster than interpreted BASIC
3. Generates reentrant Romable code
4. Can compile large programs (4000 lines +)
5. Fast one pass compiler
6. Simple commands for compiling and running programs
7. Supports separate compilation of procedures and functions
8. Compiler switch options, including conditional compilation
9. Full heap support including NEW and DISPOSE procedures that perform true heap allocation
10. Complete implementation of sets with up to 256 members. All standard set operators
11. Variant records are fully supported
12. Supports single and double precision REAL
13. Files are compatible with TRSDOS

Extensions

14. OTHERWISE clause on case statements
15. Identifiers may contain '$' and '—' characters
16. Automatic type conversion in arithmetic expressions and assignment statements
17. Constants may be expressed in decimal or hexadecimal
18. Characters within strings may be specified by ascii code. Allows non-printable characters in strings.
19. Type transfer operator to override type matching requirements
20. ESCAPE allows exit from anywhere in a procedure
21. LOCATION function returns the address of a variable
22. SIZE function returns the amount of memory required for a variable

TRS-80 Library

23. Graphics routines (setpoint, cleargraphics ....)
24. Interface to assembly language routines with parameter passing
    - •Can call operating system and ROM routines
    - •Memory may be protected from Pascal for use by assembly language routines.
25. Read keyboard (scan or wait for character)
26. Write to CRT screen with cursor addressing
27. Direct access to memory with PEEK and POKE
28. Input and output to IO ports from pascal
29. Programs may perform their own recovery from file and device errors
30. File or device names for Pascal files are determined from the keyboard when a program is executed. Alternatively a program may internally specify file names.

Full Screen Text Editor

31. Included with Pascal or available separately
32. No limit on file size (except disk capacity)
33. Full cursor movement and scrolling
34. Insert and delete characters
35. Insert, delete, duplicate, split, merge lines
36. Find string, replace string
37. Typewriter style tabs and autoindent
38. Show file, Insert file
39. Horizontal scrolling allows editing of files containing lines wider than the screen
40. Key and command mode access to commands
41. On-line documentation with HELP command
42. Files are compatible with TRSDOS
43. Can also edit text and BASIC programs
44. Many additional features

The Best of Both Worlds

45. Pseudocode (Pcode) for compactness
    - •Allows large programs in small memory space (8500 line + programs can execute in 48k)
46. Native code for speed
    - •Optional code generator produces Z80 instructions
    - •Z80 code can be mixed with Pcode

TRS-80 is a trademark of Tandy Corporation
CP/M is a trademark of Digital Research, Inc.
Z80 is a trademark of Zilog, Inc.
IBM is a trademark of International Business Machines
Z89 is a trademark of Zenith data systems

Linking Loader

47. Links separately compiled routines
48. Supports procedure and function libraries
49. Can create command files that are callable as commands from TRSDOS top level

250 Page Documentation Package

50. Beginner's guide
51. Pascal Tutorial with 500 line Data Base program. (source supplied on diskette)
52. Pascal Reference Manual
53. System Implementation Manual
54. Text Editor Manual
55. Handy System Reference Card
56. Cross reference index for documentation package

Optional Advanced Development Package

57. Pcode optimizer
    - •Reduces the size of a program by 25-30%
    - •Increases execution speed
58. Z80 native code generator
    - •Produces relocatable, reentrant native code for the Z80
    - •Native code executes 3-5 times faster than Pcode.
    - •Native code can be mixed with Pcode to provide speed where required and still benefit from the compactness of Pcode

Hardware Required

TRS-80 model I; TRS-80 model III; or CP/M system (with Z80)
48k of memory
One disk drive (two recommended)

TRS-80 version compatible with TRSDOS and most other operating systems.

Special discounts for purchase of Model I & III versions together. Available directly or through authorized dealers.

COMING SOON
Alcor Pascal for:
Zenith Z89, IBM Personal Computer, 8086 and 68000 based systems

Yes, I'm interested in Pascal for:

master charge VISA

- ☐ TRS-80 Model I
- ☐ TRS-80 Model III
- ☐ CP/M (available March 1, 1982)

☐ Send More Information	free
☐ Alcor Pascal System	$199
☐ Text Editor only	$ 60
☐ Advanced Development Package	$125
☐ Manuals only	$ 25

Name ______________________

Address ______________________

____

City ____________ State ____________ Zip ____________

Send to: ALCOR SYSTEMS
13534 Preston Road
Suite 365
Dallas, Texas 75240
For immediate service call:
(214) 226-4476

ALCOR

✓31

```
140 AU = T * E↑(-U↑2/2)
150 AR = W/2 * (AL + 2 * SM + AU)
160 PRINT AR
170 END
```

Consult a statistics table: The area between the mean and an X value of +1 should be 0.3413. We must decide what interval width to use so that our program accurately approximates this area.

Let's try 10, 20 and 100 trapezoids and compare the results. Run the program three times, using a value of X each time of 1, and a value of W of 0.1, 0.05 and 0.01.

X	Width	Actual Area	Computed Area
1	.1	.3413	.314535
1	.05	.3413	.328592
1	.01	.3413	.341343

You can see that we'll have to use an interval width of 1/100th of a standard deviation (0.01) in order to get the desired accuracy. We'll settle on an interval width of 0.01. Run the program and see what the area is corresponding to other values of X, and how things compare with the book values:

X	Width	Actual Area	Computed Area
1	.01	.3413	.341343
1.5	.01	.4332	.433192
2	.01	.4772	.477249
2.5	.01	.4938	.493791
3	.01	.4987	.498651
3.5	.01	.4998	.499768

We don't have to modify the program to find the area between two X values, both of which are something other than zero. We can simply run the program for each value of X, and then subtract (Fig. 13).

The area between values of X of 1 and 2 =

$$\begin{array}{r} .477249 \\ -\ .341343 \\ \hline .135706 \end{array}$$

The only problem you'll have is applying it to situations where the mean of the distribution is other than zero and/or the standard deviation is other than one. For these situations transform the given mean and standard deviation into a standardized mean and standard deviation of zero and one, respectively. This is done using the formula:

$$z = \frac{X - \bar{X}}{SD}$$

and then using the z value where we used the X value previously.

Suppose the grades on an examination are normally distributed with a mean of 70 and a standard deviation of 5, and we decide to award a grade of B to those scoring between 73 and 81. There are 129 in the class. How many get a B? (see Example 10).

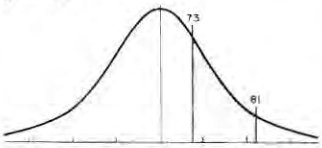


Fig. 14

Compute z values for the two limit values:

$$z_1 = \frac{73 - 70}{5} = 0.6 \text{ and } z_2 = \frac{81 - 70}{5} = 2.2$$

A score of 81 is 2.2 standard deviations to the right of the mean, and a score of 73 is 0.6 standard deviations from the mean (Fig. 14).

If you run the program and enter a value of 2.2 for X, and a width of 0.01, you should get an area (from 0 to 2.2) of 0.486097. If you again run the program and enter a value of 0.6 for X, and a width of 0.01, you should get an area (from 0 to 0.6) of 0.225745. Subtracting the second value from the first, we get an area between 0.6 and 2.2 of 0.260352. In other words, a little over 26 percent of the area under the curve is between those limits, or 26 percent of the class earned a B. To determine how many earned a B, just multiply the class size of 129 by 0.260352 and you get 33.5854.

To end this article on a happy note, we'll go ahead and give that 5854/10000 of a student a B too, so that makes 34.■

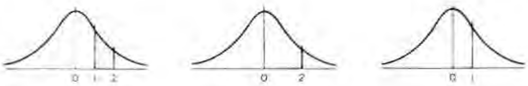


Fig. 13

-3	-2	-1	0	+1	+2	+3	standardized
55	60	65	70	75	80	85	actual

Example 10

✓See List of Advertisers on page 290

Ask your computer to help you keep out of a daze.

Can You Get Me A Date?

John T. Phillipp
104 Scout Rallos Street
Diliman, Quezon City
Metro Manila, Philippines

This article presents Dateread, a subroutine which may be added to any program and which will allow you to enter a date in just about any format you like—even 31 Aug 47 for those with a military inclination!

The Dateread subroutine (Program Listing 1) starts at line 65000 so that these numbers will not conflict with those of the calling program. It internally uses the A variables (A to A9, and A$ to A9$) so use of its variables can be easily avoided. Regardless of the format in which the date is entered, Dateread returns six values:

MO—The numeric value of the month (AUGUST returns MO = 8)
DA—The numeric value of the day (DA = 31)
YR—The numeric value of the year (1947 returns YR = 47)
MO$—The name of the month (AUGUST)
DA$—The number of the day as a string (DA$ = 31)
YR$—The numeric value of the year as a string (YR$ = 47)

These values may be used by the calling program in any way the programmer likes. The important thing is that the user is not restricted to any one format when he is asked to type in the date.

This program is divided into five sections:

- Input date string (lines 65000-65035) uses the INKEY$ function to accept any date, even those including commas, as a string. Input will not accept commas, and the line input command is not available in Level II.
- Remove spacers (lines 65040-65120) steps through the date-string as it was typed in, and separates it into a month string, day string, and year string ignoring all non-numeric, non-alpha spacer characters (/ : * - etc.).
- Alpha month to numeric (lines 65135-65155) converts a month string typed in (AUG) to a numeric value (8).
- Numeric month to alpha (lines 65160 to 65165) converts a month value typed in (8) to a month string (August).
- Error message (lines 65170-65175) repeats the date string that was entered, and suggests standard formats if the program can't make sense of what was typed in.

Dateread can help prevent the comment, "If that computer of yours is so smart, why do I have to enter the date like that?" referring to the MM/DD/YY format. Now all you have to do is write the program to use it.■

Program Listing 1

```
1000 ' ****************************************************
1010 ' *                                                  *
1020 ' *                     DATEREAD                     *
1030 ' *          A UNIVERSAL DATE INPUT SUBROUTINE       *
1040 ' *             by  John T. Phillipp, MD             *
1050 ' *                 Manila, Phillippines             *
1060 ' *                                                  *
1070 ' ****************************************************
1080 '
2000 GOSUB65000
2100 CLS:PRINT@448,"YOUR BIRTHDAY IS ";MO$;" ";DA$;", 19";YR$;".
"
2150 PRINT:PRINT"MO =";MO:PRINT"DA =";DA:PRINT"YR =";YR
3000 END
65000 ' * DATE READER *
65005 CLS:PRINT@448,"WHAT IS YOUR BIRTHDAY? ";
65010 A3$=CHR$(24)+" "+CHR$(24)+CHR$(95)
65015 A1$="":A4$="":A5$="":A6$="":PRINTCHR$(95);
65020 A$=INKEY$
65025 IFA$=""THEN65020ELSEIFA$=CHR$(13)THEN65035ELSEIFASC(A$)=8T
HENPRINTA3$;:GOTO65030:ELSEPRINTCHR$(24);A$;CHR$(95);:A1$=A1$+A$
:GOTO65020
65030 IFLEN(A1$)>0THENA1$=LEFT$(A1$,LEN(A1$)-1):GOTO65020
65035 IFLEN(A1$)=0THEN65000
65040 PRINT@448,CHR$(30);"THINKING. . .";
```

Program continues

The Key Box

Basic Level II
Model I

Program continued

```
65045 FORA=1TOLEN(A1$)
65050 A1=A:A2$=MID$(A1$,A1,1)
65055 IFASC(A2$)<48ORASC(A2$)>57ANDASC(A2$)<65THENNEXTA
65060 FORA=A1TOLEN(A1$):A2=A:A2$=MID$(A1$,A2,1)
65065 IFASC(A2$)>47ANDASC(A2$)<58ORASC(A2$)>64THENA4$=A4$+A2$:NE
XTA
65070 FORA=A2TOLEN(A1$):A3=A:A2$=MID$(A1$,A3,1)
65075 IFASC(A2$)<48ORASC(A2$)>57ANDASC(A2$)<65THENNEXTA
65080 FORA=A3TOLEN(A1$):A4=A:A2$=MID$(A1$,A4,1)
65085 IFASC(A2$)>47ANDASC(A2$)<58ORASC(A2$)>64THENA5$=A5$+A2$:NE
XTA
65090 FORA=A4TOLEN(A1$):A5=A:A2$=MID$(A1$,A5,1)
65095 IFASC(A2$)<48ORASC(A2$)>57ANDASC(A2$)<65THENNEXTA
65100 FORA=A5TOLEN(A1$):A2$=MID$(A1$,A,1)
65105 IFASC(A2$)>47ANDASC(A2$)<58ORASC(A2$)>64THENA6$=A6$+A2$:NE
XTA
65110 IFASC(LEFT$(A4$,1))<65ANDASC(LEFT$(A5$,1))>=65THENMO$=A5$:
DA$=A4$ELSEMO$=A4$:DA$=A5$
65115 YR$=RIGHT$(A6$,2):YR=VAL(YR$)
65120 DA=VAL(DA$)
65125 IFVAL(MO$)=0THENGOSUB65135ELSEMO=VAL(MO$):GOSUB65160
65130 RETURN
65135 MO=0:A7$="JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDECAAA"
65140 A8$=LEFT$(MO$,3)
65145 FORA=1TO39STEP3
65150 A9$=MID$(A7$,A,3):MO=MO+1
65155 IFA8$<>A9$THENNEXTA
65160 IFMO=1THENMO$="JANUARY"ELSEIFMO=2THENMO$="FEBRUARY"ELSEIFM
O=3THENMO$="MARCH"ELSEIFMO=4THENMO$="APRIL"ELSEIFMO=5THENMO$="MA
Y"ELSEIFMO=6THENMO$="JUNE"ELSEIFMO=7THENMO$="JULY"
65165 IFMO=8THENMO$="AUGUST"ELSEIFMO=9THENMO$="SEPTEMBER"ELSE IF
MO=10THENMO$="OCTOBER"ELSEIFMO=11THENMO$="NOVEMBER"ELSEIFMO=12TH
ENMO$="DECEMBER"
65170 IFMO<1ORMO>12ORDA<1ORDA>31ORYR<1THENPRINT@448,CHR$(30);"SOR
RY. I DON'T UNDERSTAND WHAT YOU MEAN BY ";CHR$(34);A1$;CHR$(34);
".
TRY SOMETHING LIKE  AUGUST 31, 1947 OR 8/31/47.";:ELSE RETURN
65175 FORA=1TO1500:NEXTA:GOTO65000
```

Good features + Good price = Success.

Model II Scripsit

Model II Scripsit
Tandy/Radio Shack
Fort Worth, TX
$300

Richard Harkness
1224 King Henry Drive
Ocean Springs, MS 39564

Scripsit for the Model II is Tandy's high-level entry into the word processing arena for its top-of-the-line microcomputer system. Scripsit's features compare favorably with other state-of-the-art, sophisticated, screen-oriented machine language wordmashers like Electric Pencil, WordStar and Magic Wand.

Word processing comprises two basic functions: editing and formatting. Use the edit functions to create and revise text. The formatting functions make the printout look like you want.

I need flexible formatting features, but my primary emphasis as a professional writer is text editing. The commands should be simple and require a minimum of keystrokes. An integrated system from one manufacturer mates the software specifically with the hardware. It is thus with Scripsit and the Model II.

Moving the Cursor

The arrow keys scroll and position the cursor. Some other programs require the control key and four different character keys. You can move the cursor one character at a time in any of the four directions indicated by the arrows. Pressing the Repeat key and an arrow simultaneously moves the cursor continuously. Pressing the Hold key and an arrow transports the cursor to the left or right margins or to the top or bottom of the page. Press Hold and then U or D with a number to move the cursor a specific number of lines. Press Hold and then L with a number to move the cursor to a specific line.

Basic Editing

The two special function keys F1 and F2 serve as your single-stroke edit keys. F1 opens the text at any place you position the cursor to allow insertions. F2 deletes text.

Scripsit uses Model II's reverse video feature to highlight selected text portions for bulk deleting, moving and duplicating. You do not have to use special text boundary markers. For example, to move a paragraph of text from one place to another within a document, press Control and D together (to set the define function); press P (for paragraph) to highlight the text in reverse video; and press M (for move). Position the cursor where you want the text and press Control and R together. Presto!

The Define command, used to highlight text for manipulation, has several variations. Press W to define a word, S to define a sentence, P to define a paragraph, A to define all text above the cursor, and B to define all text below the cursor.

Basic Formatting

A special format line showing the format settings appears as a line of dashes along the bottom of the video display. Set the left and right margins on this line with { and }. Typing O sets the standard outline tab to indent the first line of each paragraph.

A status line directly below the format line indicates the margin settings. It also shows the document name; current page number; cursor position on the page (which changes as you move the cursor); the left edge of the screen window area (if you scroll the text horizontally); line spacing; mode (the regular mode is called "outline"); and locked block (shown by an asterisk). Locked block protects text blocks you do not want split over two pages.

Printing

You can print text at any time while you are working in a document. You can print a document from beginning to end continuously, or pause after each page to allow insertion of another sheet of paper. Other options are: paper size in lines (1–99 with 66 as default); maximum number of text lines per page (default is 50); justification by character, word or space insertion; pitch (10 for pica or 12 for elite); number of copies of document needed (from 1 to 254); request printing of line numbers; parallel or serial-driven printer (default is parallel for all Radio Shack printers); and line feed after each carriage return (not for Radio Shack printers).

Scripsit commands use all the Daisy Wheel Printer II's features. There is bold printing; underlining; bold and underlining together; double underlining; overstrike (to create accented or other special characters e.g. £µ°ƒ); superscripts and subscripts (characters printed a half line above or below the text line); printing of vertical lines; and headers and footers in multiple variations. There is a procedure for printing multi-column text, especially useful for newsletters.

The Key Box

Model II
64K RAM
TRSDOS
1 or more disk drives
Daisy Wheel Printer II or other Radio Shack printer
Direct Memory Access

"There is a trade-off with a page-at-a-time system like Scripsit. . ."

General Comparisons

Scripsit uses only a portion of available RAM (about 8K) for text storage. RAM-oriented programs like Electric Pencil, Magic Wand, WordStar, and Auto Scribe use all available RAM (between 31K and 50K depending on the program—and assuming 64K of RAM) for text storage. With Scripsit, you cannot fill computer memory. Each time you begin on a new page, Scripsit saves the former page to disk. WordStar and Auto Scribe also have automatic save-to-disk transfers when RAM is filled. With Scripsit your document length is not limited by the amount of RAM you have. With RAM-oriented systems like Electric Pencil and Magic Wand, you must break a document into portions if it is too long for available RAM. There is a trade-off with a page-at-a-time system like Scripsit, however. Documents longer than one page produce the delays inherent in disk access. Scripsit is fairly fast in its disk procedures, but transferring a page between disk and RAM requires a few seconds. The other programs do not require disk access with documents that fit into RAM. Making changes within them is as fast as doing this within a single page in Scripsit. Scripsit, like WordStar and Auto Scribe, formats the pages on the video display exactly as they will be printed.

Scripsit is a completely integrated program, as are WordStar, Electric Pencil, and Auto Scribe. This means the editor and formatter are in the same program; you can print a portion in a specified format while you are editing (WordStar requires that text be saved to disk before printing, however). In Magic Wand, on the other hand, the editor and formatter are separate programs. With this arrangement, when you finish editing the text, you must run the formatter program. In separate programs the formatter does not take up any RAM during editing, leaving the largest possible memory space for text.

Scripsit in Detail

Once you become familiar with Scripsit, it will be easy to use. When you first sit down with this program, you could be a mite overwhelmed, especially if this is your first exposure to word processing. Because it is a rather complex package, Tandy includes an eight lesson audio cassette training course. Additionally there's an onscreen Help menu to call up when you get bogged down.

You start each Scripsit session with the directory, which lists the current date, the amount of disk storage space in use (expressed as a percent) and the number of documents on it, plus the base information for each document shown in Fig. 1.

Press the F1 key to create a new document. A prompt asks for a password; press Enter to bypass. The Create New Document menu appears, in which you name the document, set the format line, give the author or operator's name, type comments or a description, set the number of lines per page, set the activity level (H, M, L or N), and set the format (V or H). You can choose a default setting by pressing Enter. Press Enter at the end of the menu and a blank page appears with the cursor at the top left corner ready for text. Scripsit allows up to 84 lines on a page. If you try to input more than that, an error message appears: "There are too many lines of text on this page." Press Break and get a new page by pressing Control and N together. Continue inputting text when a new page appears. You can scroll by page in the document with Control and N (next) to go forward or Control and P (previous) to go backward. You can also go directly to a page by pressing Control and G (get), the page number, and Enter.

To open an existing document, position the cursor next to the document name and press O. The open document menu appears with the base information you typed in when you first created the document. If you do not want to change any information, press the Escape key to bypass the menu prompts and display the first page of the document. Add new pages between existing pages by typing a decimal page number after pressing Control and G. For example, to insert a page between pages 7 and 8, type the number 7.1 (up to 9 decimal pages can be inserted this way). There is a procedure to renumber pages and start over if you need to insert more than 9 new pages.

The repagination utility reorganizes a document with a uniform number of lines per page. You will be using it a *lot*. Be sure to repaginate a revised document *before* printing it. Otherwise there may not be a uniform number of lines per page or the last line of a page may not be complete. This procedure lets you reformat the document and hyphenate if desired.

Scripsit provides almost unlimited formatting variations within a document. You can reset the format line with new margins and change line spacing at any time for the complete document or any portion of it. You can set three kinds of tabs for columnar work: regular tabs, align tabs (for typing columns of numbers easily), and combination tabs. You can even save and recall up to eleven frequently used margin and tab settings.

Around the Keyboard

There are five special keys (besides F1 and F2) you will use often with Scripsit: Use Enter to end a paragraph, or move the cursor back to text area from a format line or status line. Use Control together with another key to give an instruction. You can use this key to bypass the prompt lists. Escape begins a utility, such as printing, copying, repagination; bypasses a document menu; goes to screen prompts on status line; and goes to disk directory. Break cancels an instruction and clears an error message. Press Hold before an arrow key to move the cursor quickly; Hold also cancels entries within a document menu and returns to default settings.

You can program five user-defined keys (J, K, Q, Y and Z) to hold up to 32 characters of repetitive information (such as a company or project name) or frequently used instructions (print instructions for underlining, bold printing, superscripts and subscripts). To call a user-defined instruction sequence, press Control and the particular key.

Other Features

The un-edit command cancels any edit changes and redisplays the page as it was last recorded.

Error messages appear and flash on the status line when you enter an instruction incorrectly. Press Break to cancel.

The merge utility constructs form letters by combining a base document with a merge (variable) file. This requires a two-drive system. You can, however, with a one-drive

Document name
Format (vertical for standard-sized paper; horizontal for some legal forms)
Date created
Date revised
Author/Operator initials (optional)
Number of pages
Size (percent of disk used)
Efficiency (machine speed/performance in percent form)
Activity level (projected amount of revision—H for high, M for moderate, L for low, N for none)
Comments/description
Selection prompts (one keystroke commands to Open a document; Copy a document; Create a document; Delete a document; Next screen to see next page of directory; Disk to see the directory for disk in drives 1, 2, or 3; Disk Utilities to format, backup, swap disks or end a session, list directory to printer, change disk defaults, enable printing from disk, disable printing from disk, print from disk).

Fig. 1. Directory information for each document.

"If you have a one-drive system, be prepared for some hair-pulling."

system, use a simple form of this utility for printing addresses on envelopes or labels.

The video mode (press Control with V) shows paragraph ends, soft returns and tab settings. It's a good practice to be in this mode when you insert and delete text, so you can see exactly what is happening. To center a line, press Control with L.

The global find/delete/replace utility finds a word, phrase or unique string of characters throughout a document and replaces or deletes it. You can choose to ignore upper/lowercase; you can stop at each occurrence or repeat continuously. You can begin the search at the beginning of the document, on any page, or from the cursor position.

Drawbacks

Scripsit's page-at-a-time orientation can be frustrating. As such, Scripsit works within the relatively small RAM space of 8K (8064 characters, about 1150 words). It does this quite well for same-disk operations.

Suppose you want to make a backup copy on another disk. Common sense and Murphy's Law dictate that a safety backup copy should be made after each major revision.

The copy utility copies the entire document one page at a time. Page one is transferred from disk to RAM, displayed on the screen, and copied; page two is transferred from disk to RAM, displayed, copied and so on. Now, if you have a two-drive system, this is done automatically. All you do is sit and wait.

If you have a one-drive system (like me), be prepared for some hair-pulling. You have to manually swap disks after each page is copied. A 20 page document requires 20 exchanges of disks! Believe me, you will be cursing vehemently after the tenth swap.

Why not use the backup utility? That takes about 20 minutes and copies user and program material, overkill when you need a copy of one document. This, too, requires manual swapping of disks several times on a one-drive system.

Tandy programmers should make a special provision to utilize all available RAM so as many pages as possible are loaded into computer memory, copied, then RAM completely filled again and copied. Copying only one page at a time is needlessly inconvenient and wasteful of a 64K system's capabilities.

I have found two ways to ease this burden. I repaginate the document before copying it to ensure that each page completely fills the 8K slice. This means that each page is set to the full margin width (96 characters) and the maximum number of lines (84) with single spacing. A 15 page document with standard settings (margin width of 60 and 50 lines per page), when repaginated, becomes a six page document requiring only six disk exchanges. After the copy process, repaginate the original document back to the standard settings.

Second, break a long document into manageable chunks or chapters and treat each as a separate document. Revise and copy each as a smaller, independent entity. Combining this technique with the first one, you can usually keep each chapter to five or six pages (5,000–7,000 words).

If you need to combine some of the chapters into a longer unbroken document, use Scripsit's Assembly utility.

Scripsit would be more efficient if its developers had taken a tip from Scripsit for the Model I and the other micro-based programs and provided a RAM buffer large enough to hold and manipulate 15 to 20 pages of text. The only advantage of the program in its present form is that, in the event of a power failure, you lose a maximum of a single page. However, with most other word processing programs, you can also save to disk after each page.

The program does not print a page directly. Suppose you are working on page 12 of a document and decide you want a printout. You call up the print utility and give the instructions to print page 12. The video screen jumps back to page 1 and displays each page in order, one after the other, until page 12 appears, and then prints it. It would be useful to be able to print a selected portion of a page, but Scripsit prints the entire page or nothing at all.

Scripsit also lacks nonprinting text comments.

Conclusions

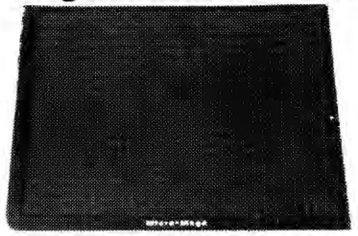
Scripsit is a satisfying and sophisticated word processing program with advanced, easy-to-use editing features and wide flexibility in formatting text. I like to print a copy of a rough draft. Word, logic and syntax errors seem easier to catch on a sheet of paper. Then I make the corrections on the screen. Having the text pages appear on the screen exactly as they do on the printout makes such revising much easier.

Scripsit will no doubt be the choice of more Model II owners than any other word processing program. Scripsit costs $300; Electric Pencil II (TRSDOS version) costs $350; Magic Wand ($400) and WordStar ($500) both require CP/M ($175). But these other programs do not support Radio Shack's Daisy Wheel Printer II; Model II owners who desire letter quality print are apt to choose this printer over the comparable but more expensive Diablo or NEC.

In the future look for an updated version of Scripsit with several enhanced features, including a dictionary to check for misspelled words. ■

for the TRS-80 from Micro-Mega

The Original GREEN-SCREEN



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 $1.50 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. ✓ 29

Micro-Mega · P.O. Box 6265 · Arlington, Va 22206

Micro-Grip™

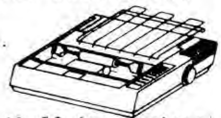
FRICTION FEED FOR YOUR EPSON

MX-70* or MX-80*

*MX-70 and MX-80 are Trade Marks of EPSON, Inc.

****

+ Converts your printer for friction feed of SINGLE SHEETS or ROLL PAPER.
+ SIMPLE Installation (all you need is a screwdriver, no soldering)
+ Tractor feed remains undisturbed.
+ Proven reliability.
+ Available NOW!
+ Only $39.95 (add $1.50 for postage)



****

SATISFACTION GUARANTEED OR YOUR MONEY BACK

VISA

VISA & Master Card Accepted
(804) 826-8808

MICRO-GRIP, Ltd.
P.O. BOX 873
LAFB, VA 23665

✓ 470

Powerful Utilities To Get More Out Of Programming your TRS-80*...

BOSS 2.2 Back to work everybody the BOSS is BACK! A BASIC Operated Single Stepper -monitor your basic programs, line by line, while they are executing! ....c $ 24.95
....d 29.95

VFU Visual File Utility - Allows you to: EXECUTE or PURGE a basic or machine language file; COPY a file from one disk drive to another or HARD copy your directory, ALL with single key entries ....c $ 19.95
....d 24.95

ULTRAZAP Disk modification utility - Read, Write, Copy, Purge or ZAP! any byte or sector on your diskette. The same utility that comes with ULTRADOS ....c $ 19.95
....d 24.95

SPOOL Parallel port, print spooler - Define spooler buffer size, lines per page, and characters per line, allows continued use of CPU during the printing process. A True RAM Spooler! ....c $ 19.95
....d 24.95

GEDIT Global Editor - Allows the user to search and replace any keyword command, variable or string with incredible speed, even pack strings for high speed graphics ....c $ 19.95
....d 24.95

RENUM-80 Program Renumberer - Tape or Disk operation, ultra fast, accurate operation ....c $ 5.95

Lower Case Driver - Will work with all Model I lower case conversions. Shortest available, only 68 bytes ....c $ 5.95

Epson Driver - For the Epson MX-80 printer, allows the use of all the functions available on today's most popular printer - Works with the Model I or Model III, cassette or disk ....c $ 5.95

OR,

Get most of these utilities and much, much more with...

ULTRADOS!
Now only $ 94.95!

LEVEL IV PRODUCTS

For Orders Only 1(800) 521-3305
For Information (313) 525-6200
32461 Schoolcraft Rd.,
Livonia, Michigan 48150

VISA MasterCard

* **A trademark of Tandy Corporation**

✓See List of Advertisers on page 290

Stuff 4K of program into 1K of memory by bank selection.

Four Into One Will Go—Part I

Jim Hawkes
25 Bainbridge Drive
Charleston, SC 29407

Grady R. Reese
500 Tulip Street
Summerville, SC 29483

The Z80 CPU used in the TRS-80 is capable of addressing 64K of memory, but even a fully expanded system does not utilize the full 64K. The area in the memory map in Appendix D of the Level II user manual labeled "reserved" (3000H to 37DDH) is just empty.

The 1K area of memory space Radio Shack left unassigned may be used to store up to 4K of utility Assembly programs in a combination of two 1K EPROM's and two 1K sections of static RAM. How do you store 4K of programs in 1K of memory space? The answer is bank selection of the four 1K segments of memory into the 1K memory space in the TRS-80.

Two Micros in One

Besides the features described above, you will get another microcomputer in the process: This project uses a commercially available single-board microcomputer called the M-80 as the basic building block. No permanent changes are necessary, although a few minor additions will be required. These additions have no effect on its use as a microcomputer.

Part II will discuss the use of the M-80 as a single board computer and its interface to the TRS-80.

Looking at memory allocation in the TRS-80 shows an unas-

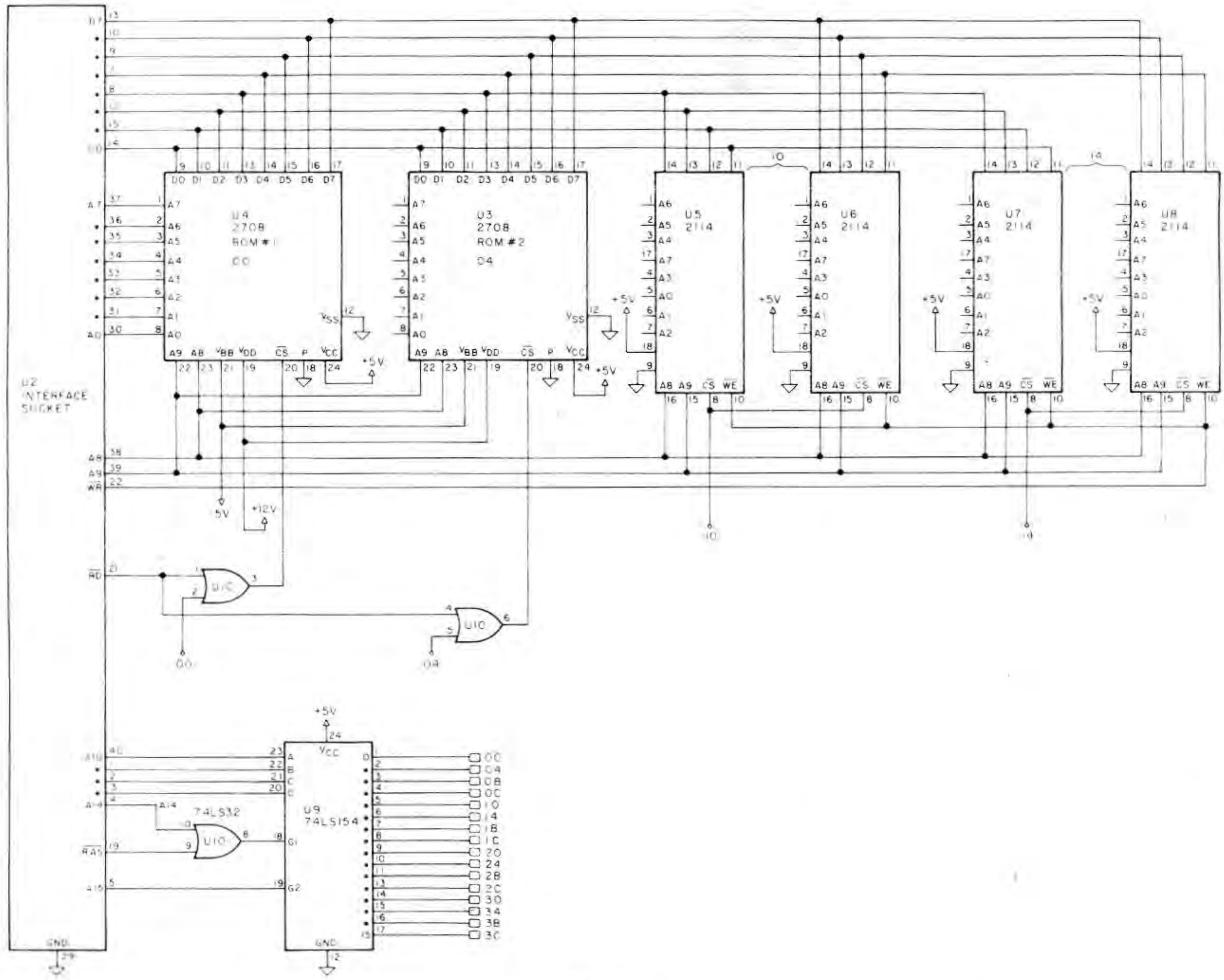


Fig. 1

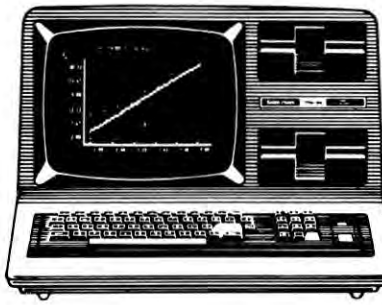
AMERICAN Small Business COMPUTERS

118 SO. MILL ST • PRYOR, OK 74361
PHONE (918) 825-4844

ALL LISTED PRICES ARE CASH DISCOUNTED. WE ACCEPT MC, VISA, AND COD (REQUIRES CERTIFIED CHECK, CASHIER'S CHECK OR CASH). FOR CHARGE ORDERS, PLEASE ADD 3%. FOR COD SERVICE, PLEASE ADD 3%. OKLAHOMA RESIDENTS PLEASE ADD 4% SALES TAX.

TRS-80 MODEL 3 MINI DISK DRIVES

Our Model 3 Mini Disk Drive is based on the famous Tandon mini disk unit. Because of it's low cost and high reliability, we thought this Tandon unit the perfect choice for the Model 3. Much testing confirmed our opinion, and we think you'll agree that it's the best mini disk for the money (and the Model 3)!!!



INTERNAL DISK DRIVE KITS FOR THE MODEL III COMPUTER

The kit includes one disk drive of your choice, controller, power supply, cables, mounting hardware, and complete instructions for easy installation.

DRIVE TYPE	DRIVE 0 External	DRIVE 1 External	DRIVE 2	DRIVE 3
Single Side 40 Track Stores about 180,000 Bytes/Disk Drive	$595.00	219.95	259.95	259.95
Dual Head 40 Track Stores about 360,000 Bytes/Disk Drive	$679.95	349.95	415.00	415.00
Single Side 80 Track Stores about 360,000 Bytes/Disk Drive	$689.95	359.95	415.00	415.00
Dual Head 80 Track Stores about 720,000 Bytes/Disk Drive	$829.95	499.95	564.00	564.00

*Drives 0 and 1 may be installed externally (with controller and power supply mounted internally). Please add $40.00 for cost of cabinet and cover.

* DEALER INQUIRIES INVITED

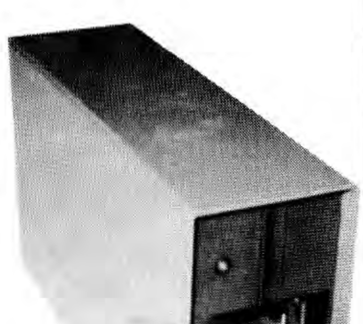
✓520

TRS-80* MODEL 1 MINI DISK DRIVES

Our Mini Disk Drives are based on the world famous TEAC Mini Disk Unit. The TEAC Unit features an estimated 8000 power-on-hours mean time between failure (that's 8 hours a day, 5 days a week, for 3.85 years!!). Our drives also feature a lead screw head positioner. This type of positioner insures accurate head positioning over the read-write media while giving a very quiet, noise-free read-write operation. If you have any doubts about our drives, our **one year** warranty should convince you that **we** know this unit is the **best** drive at the **best** price on the American market. This is one example where you can pay less and get more!!

MODEL 1 40 TRACK FULLY TRS-80 COMPATIBLE DISK DRIVE....................$265
MODEL 1 80 TRACK DRIVE (Requires NEWDOS 80 or equivalent software to work)....................$369.95
NEWDOS 80 (Version 2.0)....................$139.95
BOX OF 10 VERBATIM DATALIFE DISKETTES....................$29.95

$265



COMPLETE DRIVE WITH POWER SUPPLY

CALL: 918-825-4844

✓521
* TM Tandy Corp.

signed area of memory starting at 3000H and ending at 37CFH. The 1K section starting at 3000H and ending at 33FFH is an excellent place to store short Assembly language programs for utility functions: line printer drivers (for those who do not use Radio Shack printers); lowercase drivers; and keyboard routines for keybounce and repeat keys. These routines are usually small and require protection when stored in the user memory space, but when placed at location 3000H to 33FFH, which is inaccessible to Basic, no protection is necessary.

Getting Started

The heart of the project is the M-80 single board microcomputer manufactured by Miller Technology and available for about $30 from a number of companies including Quant Systems and Quest Electronics. In addition to the printed circuit board, you will need the parts listed in Table 1. You will also need to build or purchase a tri-voltage power supply that will provide regulated +5 volts at 280 mA, −5 volts at 35 mA, and +12 volts at 60 mA. We used the power supply described in Don Walters' article "The Mighty Five-Way Power Supply," *Kilobaud Microcomputing*, April 1980. As an alternative to the Walters power supply, the design shown in Fig. 4 represents an adequate substitute at substantial savings.

Board Design

A schematic diagram of the M-80 microcomputer board after its conversion to the TRS-80 memory board is presented in Fig. 1. Note that U9, a 74LS154 four line to 16 line decoder, decodes the address from 0000H to 3C00H in 1K increments.

In the original M-80, this 16K address space was not uniquely assigned since the address line A15 was not used in the decoding scheme. It is necessary to provide unique decoding to use the project in TRS-80s 16K and larger without memory assignment problems. To accomplish this, we connected A15 to the G2 input at pin 19 of U9 (see Fig. 1). The original scheme used a section of U10, a 74LS32 quad 2 input OR gate, to OR A14 and the MREQ (active low) signals from the Z80 CPU. In this design, both signals must be low to get a high from the OR gate, satisfying one of the inputs of the decoder (G1).

The TRS-80 bus does not provide MREQ, so we used the RAS (active low), which is generated from the Z80 MREQ (active low) in the TRS-80. A section of U11, a CD4049 inverter, was used in the M-80 to invert the CPU refresh (low) signal to satisfy another decoder input (G2), preventing static RAMs and EPROMs in the refresh cycle. The RAS (active low) signal at U10 resolves this problem on our memory board: Two gates of U10 are used to select one of the two EPROMs.

The remaining components on the M-80 board are of no interest in the memory project.

Building the Project

Construction is simple and straightforward. Begin by installing the IC sockets in locations called for in Fig. 3; be sure there are no bent pins as you insert each socket. Install the bypass capacitors and solder them in place. Next, build the bank select circuit shown in Fig. 2 following the parts placement suggested in Fig. 3. Lastly, install the wire wrap jumpers called for in the following text detailing the bank select circuit options you chose.

You are now ready to connect the power supply and verify the correct voltages at all IC locations as specified in Fig. 1.

M-80 Board to TRS-80 Interface

Because all of the buswork and address decoding has been done, interfacing the M-80 board will be very simple. Since the board was designed to use the Z80 CPU, the connection of the data, address and control lines will also be quite easy. We will use the Z80 CPU socket (U2) for our interface connections since all of the needed signals are at this location.

Table 3 shows the interface connections made through a suitable 8-inch length of 40 conductor cable terminating in an edge connector plug compatible with the TRS-80. The connections to the "raw" end of the cable are made via a 40 pin header which will plug into the M-80 CPU socket (U2). See Fig. 3 for the cable wiring table.

Some changes must be made to the address decoder to prevent the memory chips from being decoded at their intended address as used in the M-80. Lift U9 pins 1, 2, 5, 6, and 19 out of their IC socket; thus, when the 74LS154 is inserted, these pins will not be engaged. The locations labeled 00, 04, 10, and 14 now identify our 1K sections of memory rather than refer to their decoded location in the M-80 memory map. The bank select circuitry shown in Fig. 2 allows you to determine which of these sections of memory (ID) will be used at the decoded address 3000H (see Table 2).

You may build the bank select circuit shown in Fig. 2 or you may decide to enable only one section of the memory, eliminating the need for this added work. If you do not wish to use the bank select approach, connect a wire wrap jumper from the location labeled 30 at U9 to the position labeled according to the ID of the memory you wish to use (00, 04, 10, 14). The corresponding memory chips are labeled 0000, 0400, 1000, and 1400 near their sockets. Since the board does not provide pull-up to +5 volts for the unused chip selects, you must install only the memory for the section you wish to use. Installation of additional memory chips may cause erratic operation without the bank select circuitry.

If you use the bank select circuitry in Fig. 2, build its decoder in the breadboard area of the M-80 board as instructed in the previous section. Connect a wire wrap jumper from the 74LS139 decoder pin 1 to the location marked 30 at U9. Connect wire wrap jumpers from each of the locations marked 00, 04, 10 and 14 at U9 to the 74LS139 decoder pins 4, 5, 6 and 7 as shown in Fig. 3.

Next, we will provide the unique decoding mentioned earlier. The cable provides A15

Memory	ID	SW-1	SW-2
EPROM #1	00	On	On
EPROM #2	04	Off	On
RAM #1	10	On	Off
RAM #2	14	Off	Off

Table 2. Memory Selection Chart

Qty.	Designation	Description
1	PCB	M-80 printed circuit board
2	U3-U4	2708 EPROM
1	U9	74LS154 decoder
4	U5-U8	2114 RAM
1	U10	74LS32 OR gate
1	U-12	74LS139 decoder
5	C1-C5	.1 UF disc capacitor
2	R4-R5	2.2K 1/4 W resistors
1	SKT(U-2)	40 pin IC socket
3	SKT(U-3,4,9)	24 pin IC socket
4	SKT(U5-U8)	18 pin IC socket
1	SKT(U-10)	14 pin IC socket
1	SKT(U-12)	16 pin IC socket
1	SKT(SW1-2)	8 pin IC socket
1	SW1-SW2	4 position DIP switch
1	Header	40 pin DIP header plug
1	Cable	40 conductor cable with connector
1	PS	Tri voltage regulated power supply

Table 1. Parts List

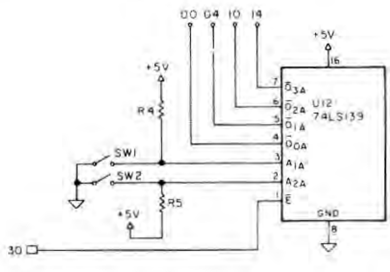


Fig. 2

✓See List of Advertisers on page 290

at pin 5 of U2; this location is labeled A15 at IC socket U2. Connect a wire wrap jumper from this pad (A15) to the bent-out pin 19 (G2) of U9. This connection ensures that the address decoded is unique and will not enable any address above 3C00H.

Testing the Memory Board

Test your interface cable by plugging it into the M-80 memory board. Use an ohmmeter to test each line from the TRS-80 card edge connector to the location on the memory board corresponding to the TRS-80 signal name (see Table 3). The power supply may be tested by applying power and verifying the correct voltages at each IC socket as detailed in Fig. 1. After you are satisfied all is correct, remove power and install all ICs. Remember to observe anti-static precautions for CMOS devices when installing the memory chips.

Plug the interface cable into the TRS-80 bus connector on the rear of your keyboard or into the screen printer port on the expansion interface. Power up in the normal manner with the memory board receiving power last. You

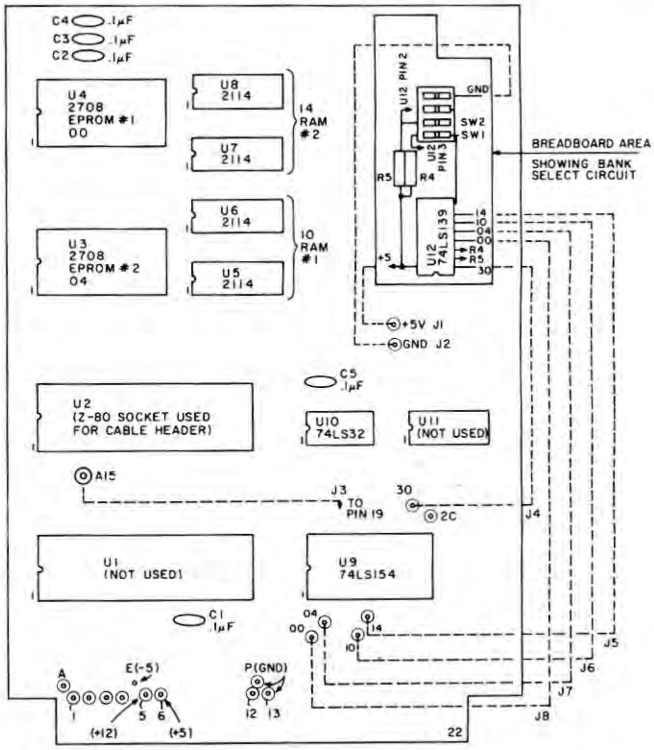


Fig. 3

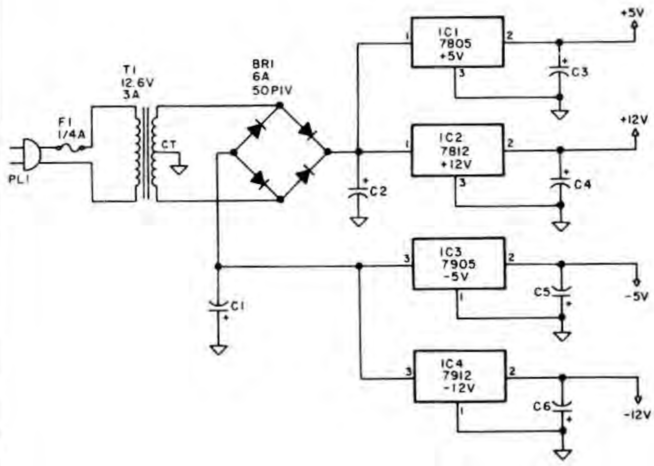


Fig. 4

should notice no difference in the operation of your machine; the new memory does not concern Basic.

Set the DIP switch on the bank select circuit for RAM number one (see Table 2). Now, load and run your favorite monitor program. We recommend the RSM monitors from Small System Software because of their wide range of features, flexibility, and ease of use. Using the monitor, write a known value into memory location 3000H to 33FFH and if everything works correctly you will see this value in memory on a subsequent read. If you are using the RSM monitors, you may run the memory test as a final check.

In order to test the EPROM locations, it will be necessary to obtain a 2708 EPROM and observe its contents. You will see FF in all locations if the device is erased. Place a programmed 2708 into the memory board and, again, look at its contents. You should see the program contained in the device.

Using the Memory Board

There are only two rules to remember in using the M-80 memory board. First, be sure the software does not exceed 1024 bytes in size. Second, be sure the software will load and run in memory location 3000H. Any Assembly language routines using relative addressing and relative jumps may be assembled to operate at the new location.

Once the software is in memory enter System. Now enter the starting address of your program in the form /aaaaa, which represents the starting address. You should see your program operating as usual except that it does not require memory protection and is not taking up valuable user memory space.■

M-80 CPU Socket		TRS-80 Cable		
Signal Name	Pin No.	Pin No.	Wire No.	Wire Color
A0	30	25	26	Green
A1	31	27	28	Orange
A2	32	40	39	Red
A3	33	34	33	Gray
A4	34	31	32	White
A5	35	35	36	Green
A6	36	38	37	Yellow
A7	37	36	35	Blue
A8	38	11	12	White
A9	39	17	18	Orange
A10	40	4	3	Gray
A11	1	9	10	Brown
A12	2	5	6	Green
A13	3	6	5	Blue
A14	4	10	9	Red
A15	5	7	8	Orange
D0	14	30	29	Red
D1	15	22	21	Black
D2	12	32	31	Black
D3	8	26	25	Blue
D4	7	18	17	Yellow
D5	9	28	27	Yellow
D6	10	24	23	Gray
D7	13	20	19	Red
WR	22	13	14	Violet
RD	21	15	16	Green
RAS	19	1	2	White
GND	29	7	8	Yellow

Table 3. Cable Assembly Table

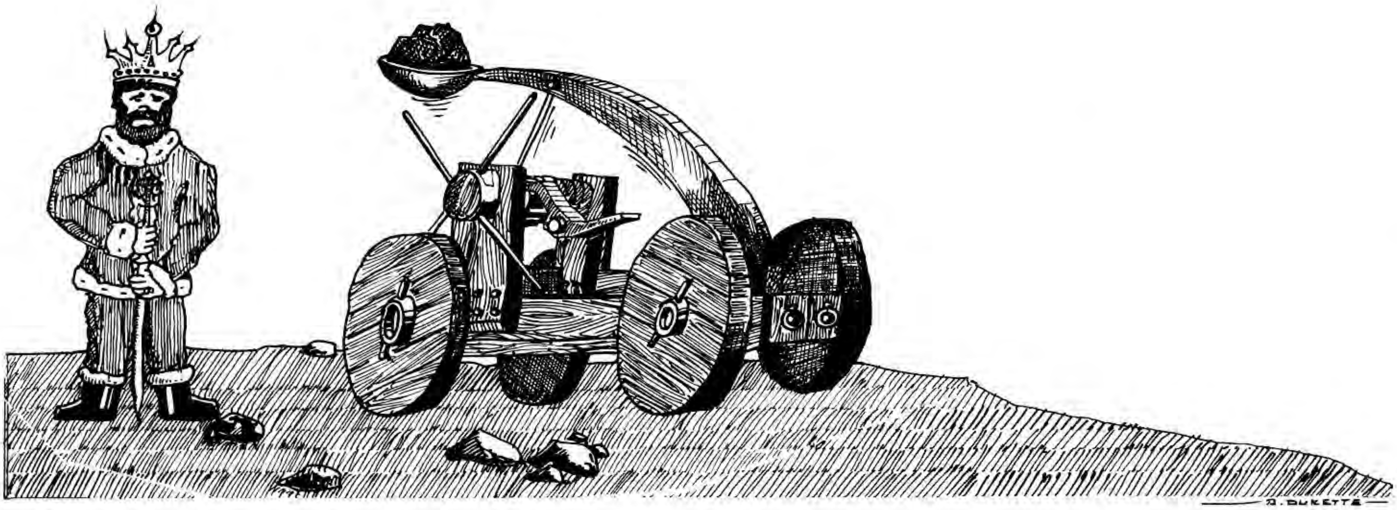
Qty.	Designation	Description	Part Number
1	F1	1/4 A fast blow fuse	RS# 270-1270
1	T1	12.6V 3A transformer	RS# 237-1511
1	BR1	6A 50PIV bridge rect.	RS# 276-1180
2	C1,C2	4700Uf capacitor	RS# 272-1022
4	C3-C6	22Uf capacitor	RS# 272-1026
1	IC1	7805 voltage regulator	RS# 276-1770
1	IC2	7812 voltage regulator	RS# 276-1771
1	IC3	7905 voltage regulator	Jameco Elect.
1	IC4	7912 voltage regulator	Jameco Elect.
1	PCB	Suitable circuit board	RS# 276-1394
4	Heat Sink	Heat sink for IC1-4	RS# 276-1363

Table 4. Power Supply Parts List

See List of Advertisers on page 290

Everyone out of the moat!

Kings and Catapults



William C. Adams
Rt. 4, Box 156D
Hattiesburg, MS 29401

Kings and Catapults is a game for one or two players in which each starts with a castle, a budget, 500 cannon, an army and one to five assassins. If you are lucky, you may also get an experimental catapult which, if it works right, does a great deal of damage.

The object of the game is to defeat your opponent, either by leveling his castle with artillery fire, destroying his army and taking his castle with your forces, or successfully infiltrating his castle with one of your assassins and killing him.

However, all this costs money—wages for your troops, $10,000 each to hire assassins, money for more cannons if those you start cut with are destroyed. Woe to him who runs out of cash—his army may fight on out of love, but it may also starve to death or switch sides.

You can always hope your father will turn up. At random moments he will come by with aid of one kind or another—possibly money, cannon, a new catapult, troops or civilians. If you run low on troops you can recruit from your civilian population, but remember that once you do, you have to meet the new payroll when the computer decides it's pay time.

When I first wrote the program it was nothing more than

The Key Box

Basic Level II
Model I or III
16K RAM

1. A$ INKEY$
2. CC$ Captain
3. MM$ Major
4. L$ Lt. Colonel
5. C1$ Colonel
6. BG$ Brigadier General
7. MG$ Major General
8. LG$ Lt. General
9. G1$ General
10. E$ Emperor
11. C$ Cannon
12. W$ Warriors
13. D$ Catapult
14. H$ Hitpoints
15. M$ Money
16. E$ Civilians

Table 2. Strings

Program Listing. Kings and Catapults.

```
0 RANDOM:CLS:PRINTCHR$(23):PRINTTAB(9)"K I N G S":PRINT:PRINT:PR
INTTAB(11)"A N D":PRINT:PRINT:PRINTTAB(5)"C A T A P U L T S":PRI
NT:PRINT:PRINTTAB(9)"(C) 1981":PRINT:PRINTTAB(5)"WILLIAM C. ADAM
S":PRINT:PRINT"    HATTIESBURG, MISS."
2 FORX=1TO3000:NEXT:'MEMBER OF THE RAMERS COMP. CLUB
3 CLS:INPUT"DO YOU WANT AN INTRODUCTION";A$:IFA$="YES"THEN5ELSE3
5
5 GOTO10
7 FORX=1TO2000:NEXT:RETURN
10 PRINT"THIS WAR TAKES PLACE FAR IN THE PAST, FAR AWAY":GOSUB7:
PRINT"THIS WAR IS FAUGHT BETWEEN MINOR KINGS IN TWO KINGDOMS.":P
RINT"EACH RULES A VERY SMALL PART OF THIER GREAT FATHER'S KINGDO
M!!":GOSUB7:PRINT"BOTH KINGS ARE BATTLING FOR THE SAME THING..."

20 GOSUB7:PRINT"TO TAKE THE OTHER'S LAND, OWN MORE PROPERTY, BEC
OME STRONGER,":PRINT"RISE IN EMPERIAL RANK AND FINALLY":GOSUB7:P
RINT"TO OWN AND RULE THEIR WHOLE COUNTRY.":GOSUB7:PRINT"BOTH KIN
G'S FATHERS WOULD LOVE TO SEE THEIR SONS REACH THIS"
30 PRINT"GOAL AND WILL OFTEN HELP HIS SON OUT!!":GOSUB7:PRINT"TO
 BEGIN TYPE RUN AND TELL THE COMPUTER WHICH KINGDOM YOU WISH  TO
 RULE":GOSUB7:PRINT"THE STATEMENTS ABOVE ARE NOW BEING DELETED"
35 DELETE2-35
40 CLEAR(100):U$=STRING$(64,191):CLS:INPUT"ENTER THE NAME OF WES
TERN RULER";N$:INPUT"EASTERN RULER";S$
```

Program continues

Program continued

```
45 RANDOM
46 ONERRORGOTO200
48 GOSUB7000
50 M=RND(30000):M=M+150000:NM=M:SM=M:PRINT"RULERS, YOU EACH HAVE
 $";M
60 NW=RND(10000):SW=RND(10000):NW=NW+20000:SW=SW+20000:PRINT"THE
 WEST HAS";NW;W$:PRINT"THE EAST HAS";SW;W$:N=RND(30000):N=N+1000
0:S=RND(30000):S=S+10000:CN=500:CS=500:DI=RND(100)
62 PRINT"AT THE BEGINNING OF THIS WAR EACH KING HAS A CERTAIN NU
MBER":PRINT"OF ALLIED SPIES OR INFILTRATORS WHICH CAN BE HIRED F
OR $10000":PRINT"EACH. IF YOU WISH TO FIGHT AGAINST THE   COMPUT
ER THEN ENTER "
63 PRINTCHR$(34)"COMPUTER"CHR$(34)" FOR THE EASTERN RULER!1"
70 PRINT:PRINT"EACH"W$"REQUIRES $2 PER HOUR FOR FIGHTING":NC=RND
(10000):SC=RND(10000):PRINT"IF YOU FIND THAT YOU NEED MORE SOLDI
ERS, HIRE"E$"FOR THE SAME PRICE":PRINT"WEST'S"E$"POPULATION";NC:
PRINT"THE EAST HAS";SC
72 NS=RND(100):IFNS<25THENNS=1ELSENS=0
73 SS=RND(100):IFSS<25THENSS=1ELSESS=0
75 IFSS=1THENPRINT"THE EAST HAS A SECRET WEAPON"
79 IFNS=1THENPRINT"THE WEST HAS A SECRET WEAPON"
80 INPUT"PRESS ENTER TO CONTINUE";A:I=RND(5):I1=RND(5)
200 CLS:PRINT@0,N$;"'S  KINGDOM":PRINT@32,S$;"'S  KINGDOM"
210 PRINT@64,"YOU HAVE $";:IFNM<0THENNM=0:PRINTNMELSEPRINTNM;:PR
INT@96,"YOU HAVE $";:IFSM<0THENSM=0:PRINTSMELSEPRINTSM;
220 PRINT@127,W$"LEFT";:IFNW<0THEN4100ELSEPRINTNW;:PRINT@159,W$"
LEFT";:IFSW<0THEN4110ELSEPRINTSW;:PRINT@192,"CASTLE'S"H$"LEFT";:
IFN<=0THEN7010ELSEPRINTN;:PRINT@224,"CASTLE'S"H$"LEFT";:IFS<=0TH
EN7020ELSEPRINTS;
223 IFCN<=0THENCN=0
224 IFCS<=0THENCS=0ELSE225
225 PRINT@256,"YOU HAVE"CN;C$:PRINT@288,"YOU HAVE"CS;C$:PRINT@31
9,E$"POPULATION IS";:IFNC<=0THENNC=0:PRINTNC;:GOTO230ELSEPRINTNC

230 PRINT@351,E$"POPULATION IS";:IFSC<=0THENSC=0:PRINTSC;:ELSEPR
INTSC;:IFNS>=1THENPRINT@383,D$;NS
235 IFSS>=1THENPRINT@415,D$;SS
238 PRINT@448,"INFILTRATOR";I:PRINT@480,"INFILTRATOR";I1
240 FORX=15872TO15934:POKEX,131:NEXT:FORX=15390TO15838STEP64:POK
EX,170:NEXT
244 'WHOSE TURN?
245 IFT=0THENT=1:PRINT@596,N$;"'S TURN":GOTO250ELSEIFT=1THENT=0:
PRINT@596,S$;"'S TURN":GOTO600
250 PRINT@640," (F)IRE CANNON   (A)DVANCE TROOPS   (C)ATAPULT
 (H)IRE CIV'S   (B)UY CANNON    (I)NFILTRATOR"
299 'NORTH'S INKEY
300 A$=INKEY$:IFA$=""THEN300ELSEPRINT@640,CHR$(31):GOSUB1000:GOS
UB5000:GOSUB20095:IFA$="F"THEN310ELSEIFA$="A"THEN400ELSEIFA$="C"
THEN450ELSEIFA$="H"THEN500ELSEIFA$="B"THEN550ELSEIFA$="I"THEN702
9ELSE300
310 IFCN=<0PRINT"YOU HAVE NO CANNON LEFT":CN=0:GOSUB4000:GOTO200
ELSEPRINT@704,"WILL YOU FIRE AT ";S$"'S (C)ASTLE OR HIS (M)EN"
320 A$=INKEY$:IFA$=""THEN320ELSEIFA$="C"THEN330ELSEIFA$="M"THEN3
50
329 'TO HIT
330 TH=RND(3):IFTH=1THENH=2:GOSUB9000:GOTO6051ELSEH=1:GOSUB9000:
D=RND(5000):S=S-D:IFS=<0THENPRINT@320,S$;"'S CASTLE IS FINISHED"
;:Y=1:GOTO20000:ELSEPRINT@320,"BOOM!!! ";S$;", YOU LOST ";D"POIN
TS":D=RND(100):CS=CS-D:PRINT" YOU LOST ";D" CANNON":D=RND(10)
340 SC=SC-D:PRINT"YOU LOST ";D" CIVILIANS":IFSS<>1THENGOSUB4000:
GOTO200ELSED=RND(10):IFD<4THENPRINT"OH NO!!! ";N$" HIT YOUR CATA
PULT":SS=SS-1:IFSS<1THENSS=0:GOSUB4000:GOTO200ELSEGOSUB4000:GOTO
200
341 GOTO6051
345 TH=RND(3):IFTH=1PRINT"YOU MISSED":
350 D=RND(1000):SW=SW-D:PRINT@768,S$;" LOST ";D" MEN":IFDI<=0THE
ND=RND(1000):NW=NW-D:PRINT@832,N$;"'S TROOPS WERE TOO CLOSE TO "
;S$"'S TROOPS. ";N$" LOST ";D" MEN":GOSUB4000:GOTO200ELSEGOSUB40
00:GOTO200
400 RANDOM:DI=0:IFSW=0GOTO4110ELSEH=11:GOSUB9000:D=RND(4500):SW=
SW-D:PRINT@128,S$" LOST ";D" MEN AND":D=RND(4500):NW=NW-D:PRINT@
192,N$;" LOST ";D" MEN IN THIS BATTLE.":PRINT"BOTH SIDES HAVE SL
ACKED OFF!":GOSUB4000:GOTO200
450 IFNS<1PRINT"YOU DON'T HAVE A CATAPULT":GOSUB4000:GOTO200ELSE
X=RND(10):IFX=5THEND=RND(1000):N=N-D:PRINT@896,"OH NO!! IT BACKF
IRED ";N$" LOST ";D" HIT POINTS!!":GOTO6051
460 X=RND(2):IFX=2THENH=6:GOSUB9000:GOTO6051ELSEH=5:GOSUB9000:X=
RND(2):IFX=2THEND=RND(11000):S=S-D:PRINT@320,S$;"'S CASTLE LOST
";D" HIT POINTS!":GOSUB4000:GOTO6051ELSED=RND(11000):SW=SW-D:PRI
NT@320,S$;" LOST ";D" OF HIS WARRIORS":GOSUB4000:GOTO200
500 IFNC=0THENPRINT@704,"THERE IS NO ONE LEFT TO RECRUIT":GOSUB4
000:GOTO200
510 PRINT@640,CHR$(31):PRINT@704,"HOW MANY CIVILIANS DO YOU WISH
 TO RECRUIT"
520 INPUTA:IFA>NCTHENPRINT@768,"THEIR ARE ONLY ";NC" CIVILIANS!"
:GOSUB4000:GOTO200ELSENC=NC-A:NW=NW+A:PRINT@768,"YOU HAVE RECRUI
TED ";A"CIVILIANS.":GOSUB4000:GOTO200
550 PRINT@640,CHR$(31):PRINT@704,"A LOUD HORN IS BLOWN. THE MERC
HANTS ARE SIGNALED.":PRINT"ONE OF THEM SAYS, 'HOW MANY??'"
560 X=RND(100):PRINT@832,"CANNON COST $";X" EACH"
570 INPUTA:A1=A:A1=A1*X:IFA1>NMTHENPRINT"YOU DON'T HAVE ENOUGH M
ONEY":GOSUB4000:GOTO200ELSEPRINT@896,N$;" JUST BOUGHT ";A" CANNO
N FOR $";A1:NM=NM-A1:CN=CN+A:GOSUB4000:GOTO200
600 'SOUTH'S INKEY
610 PRINT@640," (F)IRE CANNON   (A)DVANCE TROOPS   (C)ATAPULT
 (H)IRE CIV'S   (B)UY CANNON    (I)NFILTRATOR"
620 IFS$="COMPUTER"THENPRINT@640,CHR$(31):GOSUB1000:GOSUB6000:GO
SUB20095:X=RND(5):GOTO623ELSEA$=INKEY$:IFA$=""THEN620ELSEPRINT@6
40,CHR$(31):GOSUB1000:GOSUB6000:GOSUB20095:IFA$="F"THEN630ELSEIF
A$="A"THEN700ELSEIFA$="C"THEN750ELSEIFA$="H"THEN800
621 IFA$="B"THEN850ELSEIFA$="I"THEN7039ELSE620
623 K=K+1:IFK=1THENSS=1ELSEK=K-1
625 IFX=1THEN630ELSEIFX=2THEN700ELSEIFX=3THEN750ELSEIFX=4THEN704
0
630 IFCS<=0PRINT"YOU HAVE NO CANNON LEFT":CS=0:GOTO6051ELSEPRINT
@704,"WILL YOU FIRE AT ";N$"'S (C)ASTLE OR HIS (M)EN"
640 IFS$="COMPUTER"THEN645ELSEA$=INKEY$:IFA$=""THEN640ELSEIFA$="
C"THEN650ELSEIFA$="M"THEN670
645 X=RND(2):IFX=2THEN650ELSE670
650 TH=RND(3):IFTH=1THENH=4:GOSUB9000:GOTO6051ELSEH=3:GOSUB9000:
D=RND(5000):N=N-D:IFS<=0THENPRINT@320,N$;"'S CASTLE IS FINISHED"
;:Y=2:GOTO20000ELSEPRINT@384,"BOOM!!! ";N$;", YOU LOST ";D"POINT
S":D=RND(100):CN=CN-D:PRINT"AND LOST ";D" CANNON":D=RND(10)
660 NC=NC-D:PRINT"YOU LOST ";D" CIVILIANS":IFNS<>1THENGOSUB4000:
GOTO200ELSED=RND(10):IFD<4THENPRINT"OH NO!!! ";S$" HIT YOUR CATA
PULT":NS=NS-1:IFNS<0THENNS=0:GOSUB4000:GOTO200ELSEGOSUB4000:GOTO
200
661 GOTO6051
```

Program continues

✓See List of Advertisers on page 290

words and description, and even then my friends loved it. Since I added graphics I can't keep them out of the house—they love it as much as I, and I hope as much as you will, also. I wrote this game for all to enjoy.

The program is for a 16K Model I Level II TRS-80. It features a display at the start of each turn which tells the players how many "hit points" their castles have, how large their army is, how many cannons they have and how much money they have. It also lists all the single-key commands you need to make your move.

Once you enter a move the computer displays its graphics—two castles and the activity between them. You can watch your army charge the enemy, your infiltrator sneak into the other castle, or your shot fly towards his castle.

I hope you like the graphics; I think they're as much fun as the game itself.■

1. DI Tests distance between armies
2. D Determines the amount of damage done to the enemy
3. N$ Stores western ruler's name
4. S$ Stores eastern ruler's name
5. M Determines amount of money given to both rulers
6. CN Number of cannon owned by western king
7. CS Number of cannon owned by eastern king
8. NW Western number of warriors
9. SW Eastern number of warriors
10. N West hit points for castle
11. S Eastern hit points for castle
12. NM West's assets
13. SM East's assets
14. P West's pay amounts
15. Pl East's pay amounts
16. T Determines whose turn
17. T1 West's turn routines
18. T2 East's turn routines
19. M5 Determines when payday comes along
20. NC West's civilian population
21. SC East's civilian population
22. NS West's secret weapon
23. SS East's secret weapon
24. X For...Next loops

Lines	Program Function
0-30	Title and Introduction
35	Deletes Introduction
40-80	Initializes system and gives preliminary status of both kingdoms
200-240	Updates status reports
244-245	Determines whose turn it is
250-300	West's options and inkey routine
310-570	Carries out all western options
600-625	Test S$ to see if the west is playing against the computer. If so, then the computer takes a choice of the given options. If not, then the computer lets the Eastern player choose an option
630-870	Carries out all eastern options
Subroutine 1000	Checks for payday. If so, $2.00 per warrior is deducted
1020-1070	Subroutines which check to see how warriors react when a king has no money to pay them
4000	For...Next loop
4100-4110	Informs the defeated king that he lost for lack of warriors
5000-5030	Subroutine/checks to see if West's retired father brings reinforcements
6050-6051	For...Next delays
7000	Sets strings for status reports
7010-7020	Informs defeated king that his castle is leveled
7029-7035	Western infiltrator subroutine
7039-7050	Eastern infiltrator subroutine
8000-8010	Informs defeated king that he has been killed by an enemy infiltrator
9000-9021	Graphics for castles and sets up graphic routines
9050-9121	Sets up graphics for cannon fire, hits and misses
9200-10000	Catapult graphics
19999-20060	Determines rank and checks rank to see if a king has risen to his greatest strength
20069-20090	Graphic infiltrator subroutines
20095-End	Checks for floods and plagues

Table 1. Variables

Program continued

```
670 D=RND(1000):NW=NW-D:PRINT@768,N$;" LOST ";D" MEN":IFDI<=0THE
ND=RND(1000):NW=NW-D:PRINT@832,S$;"'S TROOPS WERE TOO CLOSE TO "
;N$"'S TROOPS. ";S$" LOST ";D" MEN":GOSUB4000:GOTO200ELSEGOSUB40
00:GOTO200
700 DI=0:IFNW=0THEN4100ELSEH=11:GOSUB9000:D=RND(2500):NW=NW-D:PR
INT@128,N$;" LOST ";D" MEN AND":D=RND(2500):SW=SW-D:PRINT@192,S$
;" LOST ";D" MEN IN THIS BATTLE.":PRINT"BOTH SIDES HAVE SLACKED
OFF!":GOSUB4000:GOTO200
750 IFS$="COMPUTER"ANDSS<1THEN620ELSEIFSS<1PRINT"YOU DON'T HAVE
A CATAPULT":GOSUB4000:GOTO200ELSEX=RND(10):IFX=5THEND=RND(1000):
S=S-D:PRINT@896,"OH NO!! IT BACKFIRED ";S$" LOST ";D" HIT POINTS
!!":GOTO6051
760 X=RND(2):IFX=2THENH=8:GOSUB9000:GOTO6051ELSEH=7:GOSUB9000:X=
RND(2):IFX=2THEND=RND(15000):N=N-D:PRINT@320,N$;"'S CASTLE LOST
";D" HIT POINTS!":GOSUB4000:GOTO200ELSED=RND(15000):NW=NW-D:PRIN
T@320,N$;" LOST ";D" OF HIS WARRIORS":GOSUB4000
761 GOTO200
800 IFSC<=0THENPRINT@704,"THERE IS NO ONE LEFT TO RECRUIT":GOSUB
4000:GOTO200
810 PRINT@704,"HOW MANY CIVILIANS DO YOU WISH TO RECRUIT"
820 INPUTA:IFA>SCTHENPRINT@768,"THEIR ARE ONLY ";SC" CIVILIANS!"
:GOSUB4000:GOTO200ELSESC=SC-A:SW=SW+A:PRINT@768,"YOU HAVE RECRUI
TED ";A" CIVILIANS.":GOSUB4000:GOTO200
850 PRINT@704,"A LOUD HORN IS BLOWN. THE MERCHANTS ARE SIGNALED.
":PRINT"ONE OF THEM SAYS, 'HOW MANY?? '":
860 X=RND(100):PRINT@832,"CANNON COST $";X" EACH"
870 INPUTA:A1=A:A1=A1*X:IFA1>SMTHENPRINT"YOU DON'T HAVE ENOUGH M
ONEY":GOSUB4000:GOTO200ELSEPRINT@896,S$;" JUST BOUGHT ";A" CANNO
N FOR $";A1:SM=SM-A1:CS=CS+A:GOSUB4000:GOTO200
1000 M5=M5+1:IFM5=11THENPRINT"PAY TIME. PROPER AMOUNT IS BEING D
EDUCTED":M5=0:P=NW*2:P1=SW*2:NM=NM-P:SM=SM-P1:IFNM<=0ANDSM<=0THE
N1030ELSEIFNM<=0THEN1025ELSEIFSM<=0THEN1020ELSEFORM4=1TO2000:NEX
T:GOTO200ELSERETURN
1020 X=RND(2):IFX=1THEND=RND(32767):SW=SW-D:PRINTD;" OF ";S$"'S
MEN JUST DIED OF STARVATION":GOTO6050ELSED=RND(32767):SW=SW-D:NW
=NW+D:PRINTD"OF"S$"'S MEN JOINED"N$"'S ARMY. YOU CAN'T PAY THEM!
":GOTO6050
1025 X=RND(2):IFX=1THEND=RND(32767):NW=NW-D:PRINTD"OF"N$"'S MEN
JUST DIED OF STARVATION":GOTO6050ELSED=RND(32767):NW=NW-D:SW=SW+
D:PRINTD"OF"N$"'S MEN JUST JOINED"S$"'S ARMY. YOU CAN'T PAY THEM
!":GOTO6050
1030 X=RND(10):IFX>6THENX=RND(2)ELSE1040:IFX=1THEND=RND(5000)ELS
E1040:IFD>NWTHEN1030ELSENW=NW-D:PRINT@832,N$;" CAN'T PAY HIS MEN
! HE HAS NO MONEY";D" OF HIS WARRIORS DIED OF STARVATION":GOTO60
50
1040 X=RND(10):IFX>6THENX=RND(2)ELSE1050:IFX=1THEND=RND(5000)ELS
E1070:IFD>NWTHEN1040ELSENW=NW-D:SW=SW+D:PRINT@832,N$;" CAN'T PAY
 HIS MEN! HE HAS NO MONEY";D" OF HIS WARRIORS JOINED ";S$;"'S AR
MY":GOTO6050
1050 X=RND(10):IFX>6THENX=RND(2)ELSE1060:IFX=1THEND=RND(5000)ELS
E1060:IFD>SWTHEN1050ELSESW=SW-D:PRINT@832,S$" CAN'T PAY HIS MEN!
 HE HAS NO MONEY!";D" OF HIS WARRIORS DIED OF STARVATION":GOTO60
50
1060 X=RND(10):IFX>6THENX=RND(2)ELSE1070:IFX=1THEND=RND(5000)ELS
E1070:IFD>SWTHEN1060ELSESW=SW-D:NW=NW+D0PRINT@832,S$;" CAN'T PAY
 HIS MEN! HE HAS NO MONEY!";D" OF HIS WARRIORS JOINED ";N$"'S AR
MY":GOTO6050
1070 PRINT@896,"YOU CAN'T PAY YOUR MEN, BUT THEY REMAIN LOYAL":G
OTO6050
4000 FORX=1TO2500:NEXT:RETURN
4100 CLS:PRINT"SORRY ";N$", NO WARRIORS LEFT. YOUR CASTLE IS TAK
EN OVER":Y=2:GOTO20000
4110 CLS:PRINT"SORRY ";S$", NO WARRIORS LEFT. YOUR CASTLE IS TAK
EN OVER!!":Y=1:GOTO20000
5000 X=RND(100):IFX>20THENRETURNELSEPRINT@832,"AHOY!!!    ";N$"'
S GREAT FATHER JUST BROUGHT HIM --":Y=RND(7)
5010 IFY=1THENX=RND(100):CN=CN+X:PRINTX;C$:GOTO6050ELSEIFY=2THEN
X=RND(10000):NW=NW+X:PRINTX;W$:GOTO6050ELSEIFY=3THENNS=NS+1:PRIN
T"ONE";D$:GOTO6050ELSEIFY=4THENX=RND(20000):N=N+X:PRINT@896,"A R
EPAIR CREW FOR YOUR CASTLE. IT NOW HAS ";N;H$:GOTO6050
5020 IFY=5THENX=RND(30000):NM=NM+X:PRINT@896,M$,"$";X:GOTO6050EL
SEIFY=6THENX=RND(10000):NC=NC+X:PRINT@896,X;E$:GOTO6050:ELSEIFY=
7THEN5030
5030 C5=RND(500):W5=RND(30000):NS=NS+1:H5=RND(30000):M8=RND(3000
0):E5=RND(30000):CN=CN+C5:NW=NW+W5:N=N+H5:NM=NM+M8:NC=NC+E5:PRIN
T@896,C5;C$,W5;W$,"ONE";D$," A REPAIR CREW WHICH RESTORES";H5;H$
,"$";M8,E5;E$:I=RND(5):GOTO6050
6000 X=RND(100):IFX>20THENRETURNELSEPRINT@832,"AHOY!!!   ";S$"'S
 GREAT FATHER JUST BROUGHT HIM --":Y=RND(7)
6010 IFY=1THENX=RND(100):CS=CS+X:PRINTX;C$:GOTO6050ELSEIFY=2THEN
X=RND(10000):NW=NW+X:PRINTX;W$:GOTO6050ELSEIFY=3THENSS=SS+1:PRIN
T"ONE";D$:GOTO6050ELSEIFY=4THENX=RND(20000):S=S+X:PRINT@896,"A R
EPAIR CREW FOR YOUR CASTLE. IT NOW HAS ";S;H$:GOTO6050
6020 IFY=5THENX=RND(30000):SM=SM+X:PRINT@896,M$,"$";X:GOTO6050EL
SEIFY=6THENX=RND(10000):SC=SC+X:PRINT@896,X;E$:GOTO6050:ELSEIFY=
7THEN6030
6030 C5=RND(500):W5=RND(30000):SS=SS+1:H5=RND(30000):M8=RND(3000
0):E5=RND(30000):CS=CS+C5:SW=SW+W5:S=S+H5:SM=SM+M8:SC=SC+E5:PRIN
T@896,C5,C$,W5;W$,"ONE";D$," A REPAIR CREW WHICH RESTORES";H5;H$
,"$";M8,E5;E$:I1=RND(5):GOTO6050
6050 FORX=1TO2000:NEXT:RETURN
6051 FORX=1TO2500:NEXT:GOTO200
7000 C$=" CANNON ":W$=" WARRIORS ":D$=" CATAPULT ":H$=" HIT POIN
TS ":M$=" MONEY ":E$=" CIVILIANS ":RETURN
7010 CLS:PRINT"SORRY, ";N$" YOUR CASTLE IS LEVELED! YOU LOSE!!":
Y=2:GOTO20000
7020 CLS:PRINT"SORRY, ";S$" YOUR CASTLE IS LEVELED! YOU LOSE!!":
Y=1:GOTO20000
7029 IFNM<10000PRINT"YOU DON'T HAVE THE MONEY TO HIRE HIM!!":GOT
O6051
7030 IFI=0THENPRINT@896,"THEY'RE DEAD":GOTO6051ELSEI=I-1:I=INT(I
):X4=RND(5):H=9:GOSUB9000:FORX=1TO500:NEXTX:PRINT@128,"HE HAS TO
 FIGHT"X4;" GUARDS.":NM=NM-10000:FORX2=1TOX4:X3=RND(2)
7031 IFX3<>1THENFORX=1TO250:NEXTX:PRINT" HE DIDN'T MAKE IT":GOTO
6051:ELSEFORX=1TO250:NEXTX:PRINT@192,"ONE DOWN":NEXTX2:FORX=1TO2
50:NEXTX:FORX=1TO250:NEXTX:PRINT"MADE IT"
7035 FORX=1TO500:NEXT:GOTO8000
7039 IFSM<10000PRINT"YOU DON'T HAVE THE MONEY TO HIRE ONE!!":GOT
O6051
7040 IFI1=0THENPRINT@896,"THEY'RE DEAD!":GOTO6051ELSEI1=I1-1:I1=
INT(I1):H=10:GOSUB9000:FORX=1TO250:NEXTX:X4=RND(5):PRINT@128,"HA
S TO FIGHT"X4" GUARDS":SM=SM-10000:FORX2=1TOX4:X3=RND(2)
7041 IFX3<>1THENFORX=1TO250:NEXTX:PRINT" HE DIDN'T MAKE IT!":GOT
O6051:ELSEFORX=1TO250:NEXTX:PRINT"ONE DOWN !":NEXTX2:FORX=1TO250
:NEXT:PRINT"HE MADE IT!!"
7050 FORX=1TO500:NEXT:GOTO8010
```

Program continues

✓ *See List of Advertisers on page 290*

Program continued

```
8000 PRINTN$;"'S INFILTRATOR KILLED ";S$:ER=0:Y=1:GOTO20000
8010 PRINTS$;"'S INFILTRATOR KILLED ";N$:WR=0:Y=2:GOTO20000
9000 CLS:FORX=16195TO16206:POKEX,191:NEXT:POKE16130,191:POKE1613
1,191:POKE16132,191:POKE16142,191:POKE16143,191:POKE16141,191:PO
KE16065,176:POKE16067,176:POKE16069,176:POKE16076,176:POKE16078,
176:POKE16080,176:POKE16135,170:POKE16071,170:POKE16007,171
9010 POKE16174,191:POKE16176,191:POKE16109,176:POKE16111,176:POK
E16113,176:POKE16239,191:POKE16249,191:POKE16250,191:POKE16185,1
91:POKE16187,191:POKE16120,176:POKE16122,176:POKE16124,176:POKE1
6179,170:POKE16115,170:POKE16051,171:POKE16081,156
9020 POKE16108,172:POKE16082,140:FORX=16240TO16248:POKEX,191:NEX
T:POKE16186,191:POKE16175,191:POKE16107,140:PRINT@896,U$;:POKE16
301,131:POKE16302,131:POKE16271,131:POKE16272,131:IFH=1THEN9060E
LSEIFH=2THEN9080ELSEIFH=3THEN9090ELSEIFH=4THEN9120ELSEIFH=6THEN1
."
9021 IFH=5THEN9200ELSEIFH=7THEN9220ELSEIFH=8THEN9230ELSEIFH=9THE
N20070ELSEIFH=10THEN20080ELSEIFH=11THEN20090ELSEH=0:RETURN:GOTO9
080
9050 ' NORTH HIT
9060 Y=34:PRINT@723,"<*";:FORX=1TO5:NEXT:PRINT@723,"  ";:FORX=39
TO125:MM=RND(7)*.02:Y=Y+MM:SET(X,Y):SET(X+1,Y):IFPOINT(X+2,Y)=-1
THEN9061ELSEFORQQ=1TO2:NEXTQQ:RESET(X,Y):RESET(X+1,Y):NEXTX
9061 PRINT@812,CHR$(128);:FORYY=1TO10:NEXTYY:PRINT@812,"**";:FOR
X=1TO10:NEXTX:PRINT@812,"  ";:FORY=191TO128STEP-1:PRINT@814,CHR$
(Y);:NEXTY:PRINT@814,CHR$(131);:RETURN
9070 ' NORTH MISS
9080 Y=34:PRINT@723,"<*";:FORX=1TO5:NEXTX:PRINT@723,"  ";:FORX=3
9TO100:MM=RND(9)*.035:Y=Y+MM:SET(X,Y):SET(X+1,Y):IFPOINT(X,Y+1)=
-1THEN9081ELSEFORQQ=1TO2:NEXTQQ:RESET(X,Y):RESET(X+1,Y):NEXTX
9081 PRINT@540,"MISSED":RETURN
9089 'SOUTH HIT
9090 Y=34:PRINT@744,"*>";:FORX=1TO50:NEXT:PRINT@744,"  ";:FORX=8
5TO33STEP-1:MM=RND(7)*.02:Y=Y+MM:SET(X,Y):SET(X-1,Y):IFPOINT(X-2
,Y)=-1THEN9091ELSEFORQQ=1TO2:NEXTQQ:RESET(X,Y):RESET(X-1,Y):NEXT
X
9091 PRINT@786,CHR$(128);:FORYY=1TO10:NEXTYY:PRINT@784,"**";:FOR
X=1TO10:NEXTX:PRINT@784,"  ";:FORX=1TO2:FORY=191TO128STEP-1:PRIN
T@783,CHR$(Y);:NEXTY,X:PRINT@783,CHR$(131);:RETURN
9110 'SOUTH MISS
9120 Y=34:PRINT@744,">*";:FORX=1TO50:NEXT:PRINT@744,"  ";:FORX=8
5TO37STEP-1:MM=RND(9)*.035:Y=Y+MM:SET(X,Y):SET(X-1,Y):IFPOINT(X,
Y+1)=-1THEN9121ELSEFORQQ=1TO2:NEXTQQ:RESET(X,Y):RESET(X-1,Y):NEX
TX
9121 PRINT@540,"MISSED":RETURN
9199 'W CATAPULT HIT
9200 LL$=CHR$(176):A=706:FORX=1TO11:A=A-62:PRINT@A,LL$;:FORU=1TO
20:PRINT@A+62," ";:NEXTU:NEXTX:FORX=1TO12:A=A+66.5:PRINT@A,LL$;:
FORU=1TO20:PRINT@A-66.5," ";:NEXTU,X
9205 PRINT@A,"*";@A-65,"**";@A-129,"***";@A-192,"**";:FORX=1TO10
:NEXT:PRINT@A," ";@A-65,"  ";@A-129,"     ";@A-192,"     ";:RETU
RN
9210 'E CATAPULT HIT
9220 LL$=CHR$(176):A=763:FORX=1TO11:A=A-66:PRINT@A,LL$;:FORU=1TO
20:PRINT@A+66," ";:NEXTU,X:FORX=1TO12:A=A+61.5:PRINT@A,LL$;:FORU
=1TO20:PRINT@A-61.5," ";:NEXTU,X
9225 PRINT@A,"*";@A-65,"**";@A-129,"***";@A-192,"**";:FORX=1TO10
:NEXT:PRINT@A," ";@A-65,"  ";@A-129,"     ";@A-192,"     ";:RETU
RN
9229 'EAST CATAPULT MISS
9230 LL$=CHR$(176):A=763:FORX=1TO11:A=A-66:PRINT@A,LL$;:FORU=1TO
20:PRINT@A+66," ";:NEXTU,X:FORX=1TO13:A=A+62.5:PRINT@A,LL$;:FORU
=1TO20:PRINT@A-62.5," ";:NEXTU,X:PRINT@A,"*";@A-65,"**";@A-126,"
*";:FORX=1TO9:PRINT@A," ";@A-65,"  ";@A-126," ";@538,"MISSED":RE
TURN
9999 'WEST CATAPULT MISS
10000 LL$=CHR$(176):A=706:FORX=1TO11:A=A-62:PRINT@A,LL$;:FORU=1T
O20:PRINT@A+62," ";:NEXTU:NEXTX:FORX=1TO13:A=A+67:PRINT@A,LL$;:F
ORU=1TO20:PRINT@A-67," ";:NEXTU,X:PRINT@A,"*";:PRINT@A-65,"**";:
FORX=1TO10:PRINT@A," ";@A-65,"  ";@538,"MISSED":RETURN
19999 'WR=WEST RANK
20000 FORX=1TO2000:NEXT:CC$="CAPTAIN":MM$="MAJOR":L$="LT. COLONE
L":C1$="COLONEL":BG$="BRIGADIER GENERAL":MG$="MAJOR GENERAL":LG$
="LT. GENERAL":G1$="GENERAL":E1$="EMPEROR":GE$="GRAND EMPEROR":I
FY=1THEN20010ELSE20030
20010 WR=WR+1:CLS:PRINTN$;"'S RANK IS NOW ";:IFWR=1THENPRINTCC$E
LSEIFWR=2PRINTMM$ELSEIFWR=3PRINTL$ELSEIFWR=4PRINTC1$ELSEIFWR=5TH
ENPRINTBG$ELSEIFWR=6PRINTMG$ELSEIFWR=7PRINTLG$ELSEIFWR=8PRINTG1$
ELSEIFWR=9PRINTE1$ELSEIFWR=10PRINTGE$:GOTO20050
20025 GOTO20045
20030 ER=ER+1:CLS:PRINTS$;"'S RANK IS NOW ";:IFER=1THENPRINTCC$E
LSEIFER=2PRINTMM$ELSEIFER=3PRINTL$ELSEIFER=4PRINTC1$ELSEIFER=5PR
INTBG$ELSEIFER=6PRINTMG$ELSEIFER=7PRINTLG$ELSEIFER=8PRINTG1$ELSE
IFER=9PRINTE1$ELSEPRINTGE$:GOTO20060
20045 PRINT"SETTING UP NEXT GAME. RANKS WILL NOT BE CHANGED!!":F
ORX=1TO1000:NEXT:INPUT"TO CHANGE PLAYERS TYPE (YES) OTHERWISE HI
T (ENTER)";A$:IFA$="YES"THEN40ELSE45
20050 PRINTN$" HAS RISEN FAR ABOVE KINGSMANSHIP AND HAS BEEN GIV
EN COMMAND":PRINT"OF THE WHOLE COUNTRY!!!!":END
20060 PRINTS$" HAS RISEN FAR ABOVE KINGSMANSHIP AND HAS BEEN GIV
EN COMMAND":PRINT"OF THE WHOLE COUNTRY!!!!":END
20069 'WESTERN INFILTRATER
20070 Y=41:FORX=30TO93:SET(X,Y):FORYY=1TO8:NEXTYY:RESET(X,Y):NEX
TX:RETURN
20079 'EASTERN INFILTRATER
20080 Y=41:FORX=93TO30STEP-1:SET(X,Y):FORYY=1TO8:NEXTYY:RESET(X,
Y):NEXTX:RETURN
20090 PRINT@540,"CHARGE!!":Y=41:X=32:X1=89:FORXS=1TO27:X=X+1:X1=
X1-1:SET(X,Y):SET(X1,Y):FORXT=1TO10:RESET(X-3,Y):RESET(X1+3,Y):N
EXTXS:FORXX=191TO128STEP-1:PRINT@862,CHR$(XX);:NEXTXX:RETURN
20095 X=RND(1000):IFX>=500ANDX<=525THEN20100ELSEX=RND(1000):IFX>
=500ANDX<=525THEN20300ELSERETURN
20100 CLS:PRINT" OH NO!! FLOODS":FORX=1TO500:NEXT:D=RND(10000):N
W=NW-D:PRINTN$" LOST "D" MEN":D=RND(10000):N=N-D:PRINT"AND LOST
"D"HIT POINTS":D=RND(1000):NC=NC-D:PRINTD" CIVILIANS"
20200 FORX=1TO2000:NEXT:PRINTS$" LOST:":D=RND(10000):PRINTD"MEN"
:SW=SW-D:D=RND(10000):PRINTD"HIT POINTS":D=RND(1000):PRINTD"CIVI
LIANS":FORX=1TO2000:NEXT:RETURN
20300 CLS:PRINTTAB(10)"IT'S A PLAGUE. I DON'T FEEL TOO WELL!":PR
INTN$" LOST--":D=RND(10000):NW=NW-D:PRINTD"MEN":D=RND(10000):N=N
-D:PRINTD"HIT POINTS":D=RND(1000):NC=NC-D:PRINTD"CIVILIANS":FORX
=1TO2000:NEXT:PRINTS$" LOST-- ":D=RND(10000):SW=SW-D:PRINTD"MEN"

20310 D=RND(10000):S=S-D:PRINTD"HIT POINTS":D=RND(1000):SC=SC-D:
PRINTD"CIVILIANS":FORX=1TO2000:NEXT:RETURN
60000 'WARNING: DELETE ALL REM STATEMENTS
60001 CLS:X=3:Y=118:XN=44:FORXS=1TO56:X=X+1:Y=Y-1:SET(X,XN):SET(
Y,XN):FORXT=1TO5:RESET(X-3,XN):RESET(Y+3,XN):NEXTXS:PRINT@926,CH
R$(191);:FORX=1TO100:NEXT
60002 GOTO60001
```

The Key Box

Basic Level II
Model I or II
16K RAM

A stock market simulator where you win by buying low and selling high.

The Trade Wins

Adam and Mark Finkelstein
Math Department
UC Irvine
Irvine, CA 92717

Have you ever thought of yourself as the "Wizard of Wall Street," complete with fantasies of pyramiding your financial bundle into a small (or large) fortune? With your TRS-80 Level II and this stock market simulator you can, at least on paper, make yourself rich. In this simulation, as in the real stock market, buying low and selling high is the key to success.

Day Trader is a Basic program which will allow you to trade (i.e., buy or sell) a single common stock, much as you would if you sat in the stock broker's office and watched the ticker tape. You are initially given $30,000 as your stake, half of it in cash, and half in the stock of some company. A random starting price is chosen by the program and displayed on the screen. The stock is then traded 100 times per day, for 30 days, with each day's stock prices displayed graphically on the screen. Each succeeding stock price is randomly a function of the previous price, varying up or down by an amount roughly between −1 percent and +1 percent of the previous price.

To Buy or Sell?

You may stop the action at any time by depressing any key; the computer will ask you if you wish to buy additional stock, sell some of the stock you have, or if you wish to review the history of your trades so far. Each time you buy or sell, you are charged a two percent commission; this realistically reflects current brokerage of trades in the $5,000 to $10,000 range. The commission is the constant C in the program, which is set in line 30. You may change this if you desire.

In addition to the random fluctuations of the stock from trade to trade, a drift factor (which is set at approximately ± $2 per day) is selected by the program once at the beginning of the game. The drift is then apportioned over the 100 trades each day. Thus your stock is, in addition to the random fluctuations from trade to trade, drifting either up or down, by an amount set by the program but unknown to you as the player. This also realistically reflects the behavior of a stock in the market place: the stock is on the way up or on the way down—you just don't happen to know which.

The program also keeps records of your trades, available for your review at any time by depressing any key to stop the display, then typing R (for records).

When you end the game the program will calculate: the amount of commission you paid; your profit, if you were a skillful (and lucky) trader; or your losses, if you were not. ■

Program Listing

```
0 REM ----- "DAY TRADER": A STOCK MARKET GAME -----
10 CLEAR 1000:DIM R$(150), R2(150), R3(150):RANDOM
20 ND=30  :REM - NUMBER OF DAYS IS 30
30 C=.02  :REM - BROKER COMMISSION IS 2%
40 P=RND(0)*70+70:S=INT(15000/P):A=30000-S*P*(1-C)
50 RC=1:R$(1)="BOUGHT IN WITH":R2(1)=S:R3(1)=P:BC=P*S*C
60 T=(RND(0)-.5)*SQR(P/5)
100 CLS:REM --- SETTING UP THE GRAPH ---
110 X=INT(P-9.5):Z=X+20
120 PRINT @960, "ACCOUNT= $";
130 PRINT @982, "SHARES=";
140 PRINT @999, "NET VAL= $";
150 PRINT @47, "PRICE= $";
160 FOR I=0 TO 43:SET (10,I):NEXT I
170 FOR I=11 TO 127:SET (I,43):NEXT I
180 PRINT @0, Z;:PRINT @448, INT(P+.5);:PRINT @896, X;
190 FOR I=1 TO 100:D=D+.01:IF D>ND THEN 510
200 PRINT @970, A;"   ";
210 PRINT @989, S;"   ";
220 PRINT @55, P;"   ";
230 PRINT @1009, INT(P*S*(1-C)+A);
240 IF P>Z OR P<X THEN 270
250 Y=43-INT((P-X)*42/(Z-X))
260 SET (I+12,Y)
270 P=P+T/100+2*(RND(0)-.5)
280 A$=INKEY$:IF A$="" THEN 500
290 B$="":PRINT @11, "RECORDS, BUY, OR SELL (R/B/S)";:INPUT B$
300 GOSUB 650
310 IF B$="B" THEN 430
320 IF B$="S" THEN 350
330 IF B$="R" THEN 700
340 GOTO 500
350 N=0:PRINT @11, "HOW MANY SHARES";:INPUT N
360 GOSUB 650:IF N<0 THEN 350
370 IF N<=S THEN 400
380 PRINT @11, "YOU CAN'T--SHARES HELD =";S
390 FOR LL=1 TO 1000:NEXT LL:GOSUB 650:GOTO 350
400 S=S-N:A=A+N*P*(1-C):BC=BC+N*P*C
410 RC=RC+1:R$(RC)="SOLD":R2(RC)=N:R3(RC)=P
420 GOTO 500
430 N=0:PRINT @11, "HOW MANY SHARES";:INPUT N
440 GOSUB 650:IF N<0 THEN 430
450 IF A>=N*P*(1+C) THEN 480
460 PRINT @11, "YOU CAN'T--MAX BUY =";INT((1-C)*A/P);
470 FOR LL=1 TO 1000:NEXT LL:GOSUB 650:GOTO 430
480 S=S+N:A=A-(1+C)*P*N:BC=BC+N*P*C
490 RC=RC+1:R$(RC)="BOUGHT":R2(RC)=N:R3(RC)=P
500 NEXTI:GOTO 100
510 A=INT(A+S*P*(1-C)+.5):BC=INT(BC+S*P*C+.5)
520 CLS:PRINT "YOUR";INT(ND);"DAYS ARE UP . . .":PRINT
530 PRINT "YOU STARTED WITH $ 30,000 IN YOUR ACCOUNT AND"
540 PRINT "BOUGHT IN WITH";R2(1);"SHARES AT $";R3(1):PRINT
550 PRINT "YOU ENDED BY SELLING YOUR";S;"SHARES AT $";P
560 PRINT "LEAVING YOU $";A;"IN YOUR ACCOUNT":PRINT
570 IF A<30000 THEN PRINT "YOU LOST $";30000-A;". . . TOO BAD!"
580 IF A>=30000 THEN PRINT "YOU MADE $";A-30000;". . . GOOD JOB!
"
590 PRINT "THAT'S ";(A-30000)*.012/ND;"% PROFIT PER YEAR!"
600 PRINT:PRINT "YOUR STOCK BROKER MADE $";BC:PRINT:PRINT
610 INPUT "PLAY AGAIN (Y/N)";P$:IF LEFT$(P$,1)="Y" THEN RUN
620 CLS:END
650 PRINT@11, STRING$(35,32):FOR LL=3 TO 5:SET(10,LL):NEXT LL:RE
TURN
700 REM --- RECORDS ----
710 CLS:PRINT TAB(15)"RECORDS:":PRINT TAB(15)"--------"
720 PRINT "DAY";INT(D)+1;TAB(35)"CURRENT PRICE= $";P:PRINT
730 PRINT "HOLDINGS:";S;"SHARES";TAB(35)"ACCOUNT =$";A:PRINT
740 FOR LL=1 TO RC
750 PRINT LL;"-- YOU ";R$(LL);R2(LL);"SHARES AT $";R3(LL)
760 IF (LL+7)/15=INT((LL+7)/15) THEN INPUT"PRESS <ENTER> TO CONT
INUE";EE$:CLS
770 NEXT LL:INPUT"PRESS <ENTER> TO RETURN TO MARKET OR TYPE 'END
' TO END GAME";EE$
780 IF EE$="END" THEN ND=D:GOTO 510 ELSE GOTO 100
```

✓496

Detail the execution path of your programs.

Performance Analysis

C. H. Ballard
P.O. Box 824
Monticello, MN 55362

In very broad terms, the performance analysis technique could be described as the measurement and evaluation of data that details the program's path during execution and the resources used to perform the various program tasks at different points along that path. Systematic performance analysis allows the professional programmer to concentrate his programming efforts on the areas that give him the greatest results per man-hour applied, and enables him to locate problems and correct them more quickly.

Directions and Definitions

The term performance analysis means the measurement of the computer's accomplishments and its programming, in terms of its objectives; providing a base of experience and data that identifies corrective actions and recommends procedures for improving the systems performance or program being measured. The basic tools used in performance analysis are hardware and software monitors.

Hardware monitors are physical electronic devices which attach directly to the circuitry of the central processor and/or its peripherals, and count or accumulate the number of occurrences of particular events, or the amount of time the system is in a particular state. For most personal computers, hardware monitors can range from simple digital logic probes to oscilloscopes to logic analyzers.

Software monitors are programs or routines that act as either a part of the operating system, or as a part of the program undergoing analysis. The software monitor may either sample event and timing data directly from internal tables maintained by the operating system, or may gather data itself at specific checkpoints inserted into the program or routine to be measured. Software monitors are generally written for particular operating systems and machines, and the types of measurements are usually restricted by the number and variety of checkpoints that are available or can be inserted into a particular system or program. Often, major modifications of the software monitor program are necessary to change the types of measurements being made, so this type of monitor can actually prove less flexible than the hardware variety.

Checkpoints, hooks, or interrupt points are selected points chosen in the sequence of a program. They break normal processing or sequencing in order to record the status or timing of certain sampled quantities important to the performance of the program or system.

It is also important to understand that, like a map, programs may be thought of as a directed graph or network. Many structures, like programs, may be visualized as graphs or networks and studied using mathematical techniques. A directed graph requires that there be a distinct mapping or pathway set from each intersection to some other intersection. A typical flow chart represents a highly condensed directed graph expressed in programming terms.

What approaches are available for performance analysis of programs on personal computers? The concentration is on high-level language programs because as the distribution of computers increases, the level of computer sophistication of the average user decreases.

The definition of performance is not the same for every program or programmer. In some cases, performance is measured by how well the program utilizes available resources, i.e., core storage. In another case, the speed with which the program processes the input data may be the most important measurement. In a real-time data acquisition environment, neither speed nor efficiency of the hardware may be needed, but adequate sampling distribution among numerous input data channels may be of primary importance. Performance is the manner in which the program fulfills the requirements of the program user and performance measurement is examining how closely the characteristics of the program meet the user's needs.

With this in mind, let's investigate the tools available to the personal computer for performance analysis and the applicability of these tools in high-level language programs.

Hardware Monitors

Basically, a hardware monitor is a set of counters that can

The Key Box

Basic Level II
Model I or III
16K RAM

↙515

be connected to various points on a computer's logic circuits to record events, such as the accessing of a certain location in memory or outputting to a certain peripheral. Monitors range from simple logic probes, to simple counters, to more sophisticated units such as logic analyzers. Hardware monitors operate on the idea that the performance of a program can be measured by counting discrete events.

I used a Paratronics Model 150 Buss Grabber S-100 system logic analyzer as a hardware monitor to improve the performance of Assembly data-acquisition systems in a commercial application. This is the simplest application involved in locating an infinite loop. By setting up the board to monitor the address bus, it was determined that the program was constantly addressing only certain areas of memory. By inspecting the coding lodged in this particular area of memory, the search was isolated to about 100 bytes instead of the approximate 10,000 the entire program occupied.

This particular board can also be easily modified to measure the relative amount of time spent by a program in a specific subroutine. It can be set to trigger its "snapshot" on the occurrence of a specified address on the address bus. By simply tying the trigger pulse to a simple counter, the number of times the program enters a given subroutine during execution can be counted. By moving the trigger address around, various portions of the program can be timed.

Simpler probes, like counters and oscilloscopes, can be used in similar ways. For example, an output instruction can be inserted into a program and the addressed output port can be monitored using a counter or a logic probe to determine if and when the portion of code containing the specified output instruction is entered.

Software Monitors

Software monitors fall into two general types: internal and external. Internal monitors operate within the coding of the program to be analyzed and are generally accompanied by physical modifications of the program to be monitored. These modifications usually take the form of special calls to a timing routine that are inserted at key points in the program. Sophisticated software monitoring systems often include an editing program that automatically finds key program points and inserts the required calls.

An external performance monitor's analysis software is kept separate from the program being monitored. One extreme class of external software monitor does no execution of the program to be analyzed at all. Utilizing topological methods, this type of monitor measures the complexity of the graph of the system. One way you can visualize this technique is to think in terms of counting up and tabulating the number of statements in each loop in the program, and then making an estimate of how many times each loop will be executed during the execution of the program. This data could be used to form both an estimate of how long the program would take to execute, and establish the areas of the program that will take the most time during that execution.

The major class of external software performance analysis techniques uses monitors operating from within the operating system of the computer. These programs become part of the operating system and utilize the privileges and interrupt structure of the operating system to gather performance data by sampling techniques. As an example, consider the possibilities of the TRON (*Trace On*) command present in some personal computer languages. When that command is used and a program subsequently executed, as each line of the program is reached and interpreted, the line number is printed. If it came printed out, the user could tabulate the number of occurrences of each line number to get a list of the most executed portions of the program. If the TRON command was a true performance analyzer, it could do this chore automatically.

```
<10><20><30><40><70><20><30><40><70><20><30><60><70><20><30><60>
<70><20><30><60><70><20><30><60><70><20><30><60><70><20><30><40>
<70><20><30><40><70><20><30><40><70><20><30><40><70><20><30><60>
<70><20><30><40><70><20><30><60><70><20><30><50><70><20><30><60>
<70><20><30><60><70><20><30><60><70><20><30><40><70><20><30><40>
<70><20><30><40><70><20><30><50><70><20><30><40><70><20><30><60>
<70><20><30><60><70><20><30><60><70><20><30><50><70><20><30><60>
<70><20><30><40><70><20><30><50><70><20><30><60><70><20><30><40>
<70><20><30><60><70><20><30><40><70><20><30><50><70><20><30><60>
<70><20><30><40><70><20><30><50><70><20><30><40><70><20><30><60>
<70><20><30><60><70><20><30><60><70><20><30><40><70><20><30><40>
<70><20><30><40><70>
```

Fig. 1

Another simple form of performance analysis by the external monitor can be found in several cross-reference programs. These programs usually generate lists of references to line numbers or variable names. A line number that has a lot of references may indicate an area of coding that can be rewritten to maximize efficient program operation. Examination of variables that are referenced frequently may show an opportunity to substitute an integer for a floating point variable, thus saving execution time and memory.

Utilizing Software Performance Analysis

A programmer uses a performance analysis with almost any reasonably complicated program because there is not usually a uniform distribution of execution time throughout the code structure. One unique rule of thumb is that 10 per cent of the coding will be responsible for 90 per cent of the execution time. It makes sense to work on first optimizing those parts of a program that are executed most frequently. The usefulness of a program performance monitor to identify those frequently used portions of a program is very evident.

A simple first step is shown in Fig. 1 which is the result of running Program Listing 1 with the TRON option. A simple count shows that statement line 60 was entered more times than any other. If that line were to be a complex calculation, it would be the first place to start optimizing.

Since getting the needed access to perform external program performance analysis, especially with ROM-based operating systems and interpreters, might be difficult even for experienced Assembly programmers, an internal software monitor will probably be more effective for the average user. The subroutines shown in Program Listing 2 are one example of how the techniques of performance analysis could be applied in the TRS-80. This example needs a disk operating system, but you don't need a disk to do the analysis.

Lines 10–340 are the sample program we will analyze. It has four main calculated sections, contained in loops starting at lines 60, 120, 180 and 240.

The hooks that get us the time and frequency information appear in lines 50, 90, 110, 150, 170, 210, 230 and 310. There are two different types of hooks, an entrance (k 1) and an exit (k – 1) type for each separate area of coding to be analyzed. The partition identifier (I) allows the timing routine to independently account for the time of each program segment when we GOSUB to it at line 61000.

Lines 60000 and above in Program Listing 2 are the performance analysis subroutines. The subroutine in lines 60000–60070 is for housekeeping. It sets up the accounting array A at the beginning of the run. The array A is here dimensioned (4,4). In general terms it could be considered an (x,4) array, where x is the number of areas of the program we need to analyze. The array elements are:

- A(x,1)—The time increment for each pass through area x of the program is stored here as it occurs. It is reinitialized to zero at the end of a pass in case

another pass is made in the course of program execution.

• A(x,2)—A simple counter, this element counts the physical number of times area x is entered during program execution.

• A(x,3)—This element is a totaler for the amount of time the program spent in execution in area x of the coding.

• A(x,4)—The increment of time used in the last pass through area x of the coding before execution terminated is stored here. This is of primary interest, of course, if only one pass was made through area x.

The timing accounting routine starting at line 61000 is the hardest working portion of the performance monitor. Line 61000 itself extracts the current time in seconds from the operating system's real-time clock the first time the subroutine is called, because A(I,1) is always set to zero initially. The second time the routine is called the result of line 61000 is the elapsed time between subroutine calls. (Note: Because of the nature of the output of the operating system's real-time clock, timing is incorrect if made over any period where the hour changes.)

If the routine is called from the entrance to a portion of coding to be timed, the subroutine returns from line 61010; otherwise, the coding to be timed has been passed, and accounting is performed in lines 61020-61036. Even if no real-time clock or timing routine is available, the frequency count made here would be useful. A simple frequency count could be expanded, for instance, to count how many times each logic exit path from a portion of code was utilized. It can also indicate what portions of code are executed most frequently even though most of the program time spent there cannot be positively verified.

Starting at line 62000 is the display subroutine. Figs. 2 and 3 give the displayed output. To change the routine to display more points, just change the value 4 in line 62020 to the number of points chosen, and also replace the value 4 in each occurrence in lines 62210 to 62260 to that same number (up to a maximum of 124 points).

Partition #	Inc. Time	Frequency	Total Time
1	9	10	84
2	21	10	216
3	8	10	86
4	13	10	125

Fig. 2

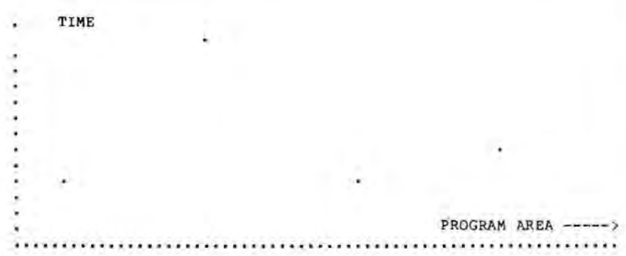


Fig. 3

Examination of the displayed results shows some of the power of systematic performance analysis. In our sample program, the most execution time spent is in area 2 of the program. In a real application, this would probably be the area that a programmer could get the best results optimizing his algorithm or coding.

One of the great advantages of systematic performance analysis is the ability to defeat that mightiest of opponents, the pre-conceived notion. If you look at what sections one and two of our sample program are doing, you find that they are doing the same thing in different ways. Some will remember that for most compilers on large computers, it was always faster to multiply a variable an integer number of times rather than raise it to an integer power. Our performance analysis tells us that it doesn't work that way. How many times have you pored over and over a listing just knowing that the flow had to go a certain way, finally giving up and placing print statements everywhere just to find that the flow didn't go the way you knew it did? A systematic approach to performance analysis could have helped there.

Before concluding, there is another observation built in to the performance analysis of our sample program—in sections three and four of the coding. The remark statement is probably the most under-used and over-used piece of coding in personal computing. Anyone who has ever struggled through a program listing trying to figure out a particular bug curses the programmer who doesn't use frequent remarks to explain his programming thought processes. On the other hand, I have also seen very well-remarked programs that suffered greatly in execution time from overextensive or poorly placed remarks. The sample program analysis bears this out, as loop three is almost 50 per cent faster than loop four, and the only difference is four blank remarks (REM is the only thing on those lines). I offer my philosophy on the REM statement:

• Active loops are no place for remarks when using non-compiled (interpreted) Basic. Keep remarks out of program flow as much as possible.

• Keep remarks short and to the point. Some interpretive Basics have been known to have to read every character

in a line even if it was just a remark.

• Consider keeping two versions of your program, one with remarks and the other with the same line numbers but with remarks deleted (programs are available to do this automatically). Make the listings available from the copy with remarks and run the stripped copy. If line numbers are kept the same and programming changes made first on the remarked copy, the convenience of remarks is kept and running time is not affected.

Systematic performance analysis is a powerful tool that professional programmers use to maximize the return on their investment of time in their programming. These techniques can also be applied to programming on the personal computer level and produce much the same result. I hope the presentation in this article has stimulated your thinking on how you could use these ideas to improve your programs and your understanding of how they function.■

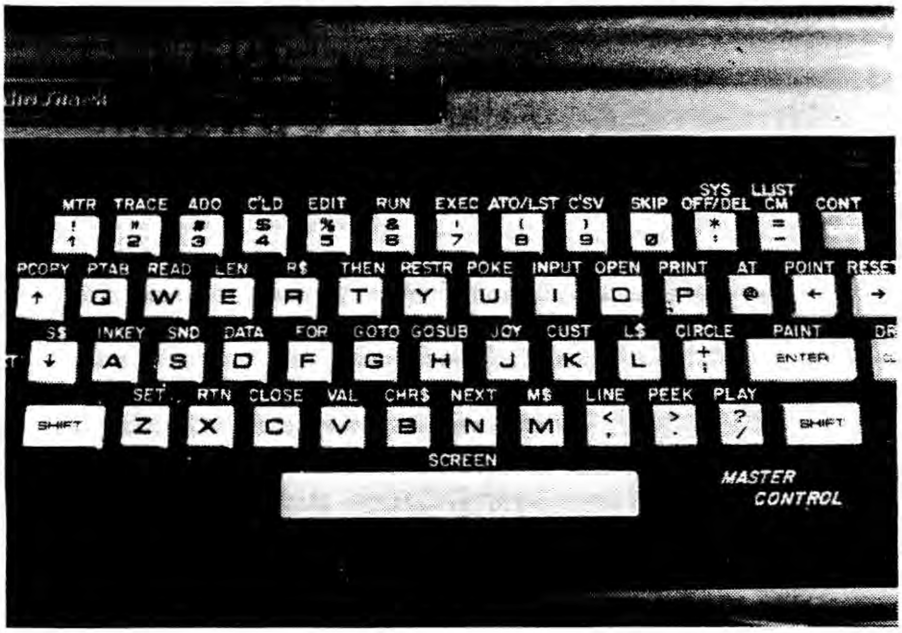
```
10 DEFINT I-N
20 GOSUB 60000
30 FOR M=1 TO 2
40 REM RAISE X TO A POWER
50 I=1:K=1:GOSUB 61000
60 FOR N=1 TO 1000
70 X=X[8
80 NEXTN
90 I=1:K=-1:GOSUB61000
100 REM RAISE X TO A POWER ANOTHER WAY
110 I=2:K=1:GOSUB61000
120 FOR N=1 TO 1000
130 X=X*X*X*X*X*X*X*X
140 NEXT N
150 I=2:K=-1:GOSUB61000
160 REM PLAIN LOOP
170 I=3:K=1:GOSUB61000
180 FOR N=1 TO 1000
190 X=X+X
200 NEXT N
210 I=3:K=-1:GOSUB61000
220 REM A REMARK-ABLE LOOP
230 I=4:K=1:GOSUB61000
240 FOR N=1 TO 1000
250 REM
260 REM
270 X=X+X
280 REM
290 REM
300 NEXT N
310 I=4:K=-1:GOSUB61000
320 NEXT M
330 GOSUB 62000
340 END
60000 DEFINT A
60010 IFM=1DIMA(4,4)
60020 FOR I=1TO4
60030 FOR J=1 TO 3
60040 A(I,J)=0
60050 NEXT J
60060 NEXT I
60070 RETURN
60080 REM ** TIMING SUBROUTINE STARTS HERE **
61000 A(I,1)=VAL(RIGHT$(TIME$,2))+60*VAL(LEFT$(RIGHT$(TIME$,5),2
))-A(I,1)
61010 IF K=1 RETURN
61020 A(I,2)=A(I,2)+1
61030 A(I,3)=A(I,3)+A(I,1)
61035 A(I,4)=A(I,1)
61036 A(I,1)=0
61040 RETURN
62000 REM ** DISPLAY RESULTS  **
62010 PRINT"PARTITION #","INC. TIME","FREQUENCY","TOTAL TIME"
62020 FOR I=1 TO 4
62030 PRINTI,A(I,4),A(I,2),A(I,3)
62040 NEXT I
62050 INPUT "HIT ENTER TO DISPLAY";A$
62060 CLS
62070 FOR Y=0 TO 47
62080 SET (0,Y)
62090 NEXT Y
62100 FOR X=0 TO 127
62110 SET (X,47)
62120 NEXT X
62130 FOR X=2 TO 126 STEP 2
62140 RESET (X,47)
62150 NEXT X
62160 FOR Y=47 TO 2 STEP -9
62170 RESET (0,Y)
62180 NEXT Y
62190 PRINT@936,"     PROGRAM AREA ----->"
62200 PRINT@5,"TIME"
62205 X=0
62210 FOR I=1 TO 4
62220 X=X+A(I,3)
62230 NEXT I
62240 T=X/2
62250 FOR I=1 TO 4
62260 X=10+(I-1)/4*124
62270 Y=47-(47*A(I,3)/T)
62280 SET(X,Y)
62290 NEXT I
62300 GOTO 62300
```

Program Listing

For Your Color Computer

MASTER CONTROL

Copyright ©1981 Soft Sector Marketing, Inc. – Written by A. Schwartz



Requries 16-32K

1. 50 preprogrammed command keys. Standard and Extended command.
2. Direct control of motor, trace, and audio from keyboard.
3. Automatic line numbering.
4. Programmable Custom Key.
5. Direct Run Button.
6. Keyboard overlay for easy program use.
7. Easy entry of entire commands into computer.

Load Master Control into your machine then either type in a BASIC program or load one in from tape to edit. Cuts programming time by 50% or more **$24.95**

For The Radio Shack
Color Computer*

COLOR BONANZA

some 4K, some 16K.

Written by E.R.I.

50 PROGRAMS In One Package

Distributed by:

SSM SOFT SECTOR MARKETING, INCORPORATED

*The Color Computer is a product of Radio Shack, division of the Tandy Corporation.

List **$49.95**

Tape 1		
Keys	4k	NE
Bagels	4k	NE
Find	16k	NE
Darts	4k	NE
Motor	4k	NE
Bomber	4k	NE
Football	16k	EXT
Kapow	4k	NE
Dodge	4k	NE
Tape 2		
Bounce	16k	EXT
Tank	32k	EXT
One Arm	4k	NE
Chute	16k	EXT
Where is it	16k	EXT
Lunar Lander	16k	EXT
Stock Market	4k	NE
Tape 3		
Multiply	16k	EXT
Divide	16k	EXT
Add Sub	16k	EXT
Simple Simon	4k	NE
Hangman	16k	NE
Beast	16k	NE
Count Down	4k	NE
Acey	16k	NE
Genie	16k	NE
Protect	16k	EXT
Tape 4		
Graphics	16k	EXT
Songs	16k	EXT
Joy	16k	EXT
Mortgage	16k	EXT
Checkbook	16k	EXT
Draw 1	16k	EXT
Morris	16k	EXT
Sound	16k	EXT
Tape 5		
Ram	16k	MA
Trace	16k	MA
MMaster	16k	MA
Demo	16k	NE
Disassembler	16k	NE
Basbug	16k	NE
Ohmlaw	4k	NE
Convert	4k	NE
Drawer 2	32k	EXT
Degrees	4k	NE
Tape 6		
Hurdler	32k	EXT
Entrap	16k	EXT
Search	16k	NE
Flip Flop*	16k	EXT
Lost Atom	16k	EXT
Attack	16k	EXT
Cartel*	32k	EXT

MA = Machine language
NE = Non Extended Basic
EXT = Extended Basic

C.O.D. – certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Out of the country orders add $10.00 extra shipping.
*TRS-80 is a product of Radio Shack, div. of the Tandy Corp.

SSM SOFT SECTOR MARKETING, INCORPORATED
6250 Middlebelt • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions **313-425-4020** ✓495

AMERICAN EXPRESS
The American Express® Card. Don't leave home without it.™
MasterCard
VISA®

Put 127 more spaces on your tab limit.

Tab Extender

David C. Hedinger
206 Chestnut Way
Salisbury, MD 21801

If you have a printer, you have probably found out no matter how hard you try, you cannot get it to tab past column 63. At first, you may have thought your printer was at fault. According to the Level II Basic Reference Manual, LPRINTTAB (X) should work with any number from 0–255. So what was wrong?

Actually, the tab is limited by the Level II ROM to a maximum of 63 spaces. This is apparently done to accommodate the 64-column format of the video display. Where does this leave someone with an 80-column or 132-column printer?

This program, Tab Extender, is a 28-byte patch which extends the tab limit to 127 spaces. The program is written for a TRS-80 Model I Level II with no disk. Two versions are supplied. One is an assembly program listing and the other is a Basic listing designed to be added to an existing Basic program. The program may be relocated, allowing it to be inserted in a Basic string and called by the ROM when needed.

The Key Box

Basic Level II
Model I

Getting Control From Basic

The code which limits the tab is in ROM at hexadecimal addresses 2137H–2140H. A Call 41D3H instruction is at address 2141H. This section of code is executed each time the TAB(token is encountered. Now 41D3H is in reserved RAM and normally contains a return instruction, which immediately returns to ROM without doing anything. If we substitute a JP (jump) instruction pointing to our Tab extender routine, we can take control from the interpreter long enough to accomplish our ends. Afterwards, we return control to the ROM, which continues as if nothing happened.

How the Program Works

Referring to Listing 1, the first section of TABPTC removes the return address from the stack and puts it into the HL register, while saving the previous contents of HL in its place. The LSB (least significant byte) is tested to see if the return address is 2144H, since there is another Call 41D3H in ROM we wish to ignore. If it is not 2144H, then the stack and HL are restored and control goes back to ROM with RET NZ instruction. Otherwise, the patch is executed.

When the patch is entered, HL is pointing to the right parenthesis,), of the current Tab(X) statement in the Basic program. At this point the tab has already been limited to its 63-space maximum, so we have to retrace a little to extend the limit. Backup (lines 150–180) changes the HL register to point to the TAB(token (0BCH) which precedes the number inside the parentheses. Next, a Call to ROM location 2B1BH evaluates the expression in the parentheses and returns with the number of spaces to tab in the A register. ANDing this number with 7FH limits the tab to 127 spaces and saves it in the E register.

Now, an RST 08H instruction tests the next HL register character to see if it is a right parenthesis. If it is, control returns to line 240. If not, then a syntax error has occurred and the SN error routine will be invoked. Assuming no syntax error, we need to restore the stack pointer and the HL register to their original values. If we do not do this, the ROM will get confused and a system "crash" will result. Lines 240–300 do the housecleaning chores. Line 240 restores HL to the proper place. The EXX instruction at 250 brings in the alternate register set of the Z 80 for temporary use. This allows us to do some fancy POPping and PUSHing to adjust the stack to its original condition. Another EXX restores the original register set. Line 300 replaces the return address in the stack with the current Basic character pointer. HL now contains the return address, the stack is restored, and the E register has the tab position from 1–127. The final instruction, JP (HL), returns us to ROM address 2144H.

Using the Assembler Version of Tab Extender

Locate TABPTC at 403EH if you do not already have a program there. There is no need to set memory size if you have it at this location. Another good place for the patch is in high memory. Use the top 28 bytes of your particular memory configuration (16K, 32K or 48K) and be sure to set memory size to protect it from Basic.

The Basic Version of Tab Extender

Program Listing 2 is the Tab Extender patch converted to Basic. This routine may be added to the beginning of any Basic program requiring the tab modification. It will eliminate the extra hassle of loading a

System tape before you load your Basic tape.

In Line one, A string is a dummy string which will have the machine code routine POKEd into it. The POKE statement corrects a data read flaw in some Level II ROMs. You may not need this.

Line two locates the pointer to A string and sets I1 and I2 equal to the least significant byte and most significant byte of the address of the first character in A string.

Line three puts the decimal value of the location of A string in I3, reads the data in line four, and POKEs the data into A string. The three final POKEs load locations 41D3H–41D5H with a jump to A string, where the patch is now located.

Using the Basic Version of Tab Extender

Make sure the variables used by this program do not conflict with those in your program. It is important to check A string, since the machine language routine is here. Change the variable name if necessary. Be sure there are at least 28 characters in A string to accommodate the routine. When the extender is added to your program, save a new copy of the combined programs on a cassette. Now type RUN and the patch will be set up before your program begins. From then on, any tab up to 127 will be honored by the ROM. No other changes to your program are necessary.

If your printer has more than a 132-column capability, you can further extend the tabbing by eliminating the AND 7FH instruction from the program, allowing tabs up to 255. This instruction is at line 200 in the assembly listing, and is item 16 and 17 (230,127) in the Basic data list. Also change line three to read: For I4=I3 to I3+25: This accomodates the shorter data list.■

```
            00000 ;         **************************
            00010 ;         * TAB EXTENDER VERS. 1.1 *
            00020 ;         * MARCH 15, 1981         *
            00030 ;         * BY DAVID C. HEDINGER   *
            00040 ;         **************************
            00050 ;
            00060 ;         ** SET UP PATCH RELAY  **
            00070 ;
41D3        00080           ORG     41D3H   ;PRINT TAB( RELAY POINT
41D3 C33E40 00090           JP      TABPTC  ;INSERT JUMP TO TAB PATCH
            00093 ;
            00095 ;**      MAIN PROGRAM        **
            00097 ;* ORG MAIN PROGRAM AT TOP OF MEMORY IF PREFERRED  *
            00098 ;
403E        00099           ORG     403EH   ;SAFE AREA FOR NON-DOS USERS
403E E3     00100 TABPTC    EX      (SP),HL ;GET RETURN ADDRESS IN HL
403F 7D     00110           LD      A,L     ;LSB
4040 FE44   00120           CP      44H     ;CHECK FOR PROPER CALL POINT
4042 E3     00130           EX      (SP),HL ;RESTORE STACK & HL
4043 C0     00140           RET     NZ      ;RETURN IF WRONG CALL
4044 2B     00150 BACKUP    DEC     HL      ;  ELSE FIND TAB( TOKEN
4045 7E     00160           LD      A,(HL)
4046 FEBC   00170           CP      0BCH    ;TAB( TOKEN
4048 20FA   00180           JR      NZ,BACKUP       ;JUMP IF NOT FOUND
404A CD1B2B 00190           CALL    2B1BH   ;ELSE EVALUATE #
404D E67F   00200           AND     7FH     ;LIMIT TO 127
404F 5F     00210           LD      E,A     ;PUT TAB INTO E REG
4050 CF     00220           RST     08H     ;CHECK FOR SYNTAX ERROR
4051 29     00230           DEFB    29H     ;RIGHT PAREN.  )
4052 2B     00240           DEC     HL      ;RESTORE HL
4053 D9     00250           EXX             ;SWAP REGISTER SETS
4054 E1     00260           POP     HL      ;GET RETURN ADDRESS
4055 C1     00270           POP     BC      ;ADJUST STACK POINTER
4056 E5     00280           PUSH    HL      ;SAVE RETURN ADDRESS
4057 D9     00290           EXX             ;RESTORE REGISTER SET
4058 E3     00300           EX      (SP),HL ;PUT HL ON STACK.
            00305 ;                          GET RETURN ADDRESS IN HL
4059 E9     00310           JP      (HL)    ;BACK TO THE ROM
0072        00320           END     0072H   ;BASIC ENTRY POINT
00000 TOTAL ERRORS
```

Program Listing 1.

```
1 A$="1234567890123456789012345678":POKE16553,255
2 I=VARPTR(A$):I1=PEEK(I+1):I2=PEEK(I+2):IFI2>127THENI2=I2-256
3 I3=I2*256+I1:FORI4=I3TOI3+27:READI5:POKEI4,I5:NEXT:POKE16851,1
95:POKE16852,I1:POKE16853,I2
4 DATA227,125,254,68,227,192,43,126,254,188,32,250,205,27,43,230
,127,95,207,41,43,217,225,193,229,217,227,233
5 REM
```

Program Listing 2.

✓See List of Advertisers on page 290

Does your Electric Pencil get the most from your Line Printer IV?

Print Whiz

John A. Parker
2505 Aspen Street
Alamogordo, NM 88310

Line Printer IV has many features that you cannot access from commercial word processors. The printer will do subscripts, superscripts, underlining, change of character size and change to double width characters.

The Electric Pencil has no capabilities to output the necessary codes to control these features of the printer, but through the use of a Pencil file, control codes can be moved into your text that will allow you to make use of these special Line Printer IV features. To create this file, simply:

- Make up a control code file.
- Load file before writing text.
- Move control codes into text using Block Move command.
- Save text without control code file.

> ***The Key Box***
>
> **Basic Level II**
> **Model I**
> **32K RAM**
> **Line Printer IV**
> **Electric Pencil**
> **Superzap or similar zap utility**

Procedure

The Line Printer IV (Centronics 737) is capable of many functions such as moving the platen up or down half a line for subscripts and superscripts (see Example 1). The procedure for controlling these functions is to precede your data with one or two control characters (see Table 1). These control characters can be sent to the printer in a Basic program by using the CHR$(X) function. For example, to go to proportional letters you could write: 10 LPRINT CHR$(27); CHR$(17), and the printer would receive the ASCII characters 27 and 17. This sequence would put the printer in proportional characters mode. The Electric Pencil stores all of its data in the ASCII format, but because the control characters are not represented by any letter of the alphabet, you can't key in these codes from the keyboard.

Here is an unglamorous way to get the correct codes into the data file without making difficult changes to the program. First, make up an Electric Pencil file containing statements defining what you want to do (see Example 2). Note: Leave at least two spaces for the control characters in front of each line. After making up this file, save it to disk with the name LPIV (for Line Printer IV) and exit Electric Pencil. Now run Superzap (or any file access program that can write data in a specified sector) with the <DFS> option. Answer the questions with LPIV and sector 0 as the sector to be modified (see Example 3). When Superzap displays your file, the section on the right will make it clear that you are in the correct place (see Example 4). You will be able to read the lines you keyed in before. Now, remembering that the ODs are carriage return characters, and 20s are spaces, with the MODxx command you can modify those leading spaces (using the hexadecimal codes supplied in Table 1) in each line to control codes for the Line Printer IV. Let's take an example. Looking at Example 4 you can see that the second line up from the bottom starts with E0. If you go over six hex characters you get to the character 0D. This is the carriage return character terminating the line "This is the Start and Stop Underline Sequence." Right behind the 0D you will see 2020 20. These are the space characters you left in front of each line when making up LPIV. In this case, these are the three spaces in front of the line "This is the Start Proportional Letters Sequence." Now if you type in <MODE6> the flashing cursor will appear at location E6, which is the first space (20). Using Table 1, to start proportional you need a 27, then a 17 sent to the printer. Typing in 27 and 17 will change the first two 20s to the needed 27 and 17. After doing the rest of the codes in the same manner, write the file to disk. Superzap will take care of this chore for you. It will ask you to answer yes to write the changed sector to disk. It takes care of the checksums for you.

Once the file is saved, it is easy to use. Before you write a document, load LPIV. Write your document right behind it, and whenever you need one of the control characters, you can get back to LPIV with the control B (cursor to beginning of text). You can use the Electric Pencil block move feature to put the code in the proper place in your text. To do this, simply move the cursor to the first control character and press shift and up arrow together. You then move the cursor to just beyond the second control character and again press shift and up arrow together. Now the control character sequence is marked and can be moved anywhere in an Electric Pencil text with the control H Block Move command. For convenience if you have an often used sequence, mark it before you start your document. Anytime it is needed you can simply use control H to get it in your text. Once your text is written, just move the cursor down to the beginning of your text and save the document. LPIV won't be saved with it.

More than one sequence can be put together, such as ½ platen up—½ platen down, which then can be moved into the text with one block move. The data that is to be in superscript or subscript can

A. General Properties of glyphosate:[7] (N-Phosphonomethylglycine)

$HOOCCH_2NH_2^+CH_2PO_3H^-$

Example 1

```
THESE ARE SOME COMMANDS TO BE USED WITH THE LP-4.
THEY CAN BE BLOCK MOVED INTO A DOCUMENT.....
  THIS IS THE START UNDERLINE SEQUENCE
  THIS IS THE STOP UNDERLINE SEQUENCE
           THIS IS THE START AND STOP UNDERLINE SEQUENCE
   THIS IS THE START PROPORTIONAL LETTERS SEQUENCE (M=0)
       THIS IS THE LINE FEED DOWN THEN UP SEQUENCE
       THIS IS THE LINE FEED UP THEN DOWN SEQUENCE
   THIS IS THE 1/2 LINE FEED DOWN SEQUENCE
   THIS IS THE 1/2 LINE FEED UP SEQUENCE
   THIS IS THE FULL LINEFEED REVERSE SEQUENCE
   THIS IS THE START DBL WIDTH LETTERS SEQUENCE
   THIS IS THE STOP DBL WIDTH LETTERS SEQUENCE
   THIS WILL SELECT 10 CPI CHARACTERS (M must =0)
   THIS WILL SELECT 16.7 CPI CHARACTERS(M must =0)
```

Example 2

```
APPARAT'S SUPERZAP/80.  INPUT ONE OF THE FOLLOWING FUNCTIONS:
'DD'  -  OR  NULL  -  DISPLAY DISK SECTOR
'DM'  -  DISPLAY MAIN MEMORY
'DFS'  -  DISPLAY FILE'S SECTOR
'DDRS'  -  DISPLAY DRIVE RELATIVE SECTOR
'DMDB'  -  DISPLAY MEMORY DUMP BLOCK
'VDS'  -  VERIFY DISK SECTORS
'ZDS'  -  ZERO DISK SECTORS
'CDS'  -  COPY DISK SECTORS
'CDD'  -  COPY DISK DATA
'EXIT'  -  END SUPERZAP, EXIT TO DOS
PRINTER OUTPUT.  APPEND ,P TO DD, DM, DFS, DDRS OR DMDB.
DFS
FILESPEC?  LPIV/PCL
REL-SECTOR-WITHIN-FILE #?  0_
```

Example 3

```
DRV 00  0D54 4845 5345 2041 5245 2053 4F4D 4520 .THESE.ARE.SOME.
0   10  434F 4D4D 414E 4453 2054 4F20 4245 2055 COMMANDS.TO.BE.U
0H  20  5345 4420 5749 5448 2054 4845 204C 502D SED.WITH.THE.LP-
    30  342E 0D54 4845 5920 4341 4E20 4245 2042 4..THEY.CAN.BE.B
TRK 40  4C4F 434B 204D 4F56 4544 2049 4E54 4F20 LOCK.MOVED.INTO.
29  50  4120 444F 4355 4D45 4E54 2E2E 2E2E 2E0D A.DOCUMENT......
1DH 60  2020 5448 4953 2049 5320 5448 4520 5354 ..THIS.IS.THE.ST
    70  4152 5420 554E 4445 524C 494E 4520 5345 ART.UNDERLINE.SE
TRS 80  5155 454E 4345 0D20 2054 4849 5320 4953 QUENCE...THIS.IS
0   90  2054 4845 2053 544F 5020 554E 4445 524C .THE.STOP.UNDERL
0H  A0  494E 4520 5345 5155 454E 4345 0D20 2020 INE.SEQUENCE....
    B0  2020 2020 2020 2020 5448 4953 2049 5320 ........THIS.IS.
FRS C0  5448 4520 5354 4152 5420 414E 4420 5354 THE.START.AND.ST
0   D0  4F50 2055 4E44 4552 4C49 4E45 2053 4551 OP.UNDERLINE.SEQ
0H  E0  5545 4E43 450D 2020 2054 4849 5320 4953 UENCE....THIS.IS
    F0  2054 4845 2053 5441 5254 2050 524F 504F .THE.START.PROPO
```

Example 4

then be inserted between them using the Electric Pencil insert feature.

In the Line Printer IV manual, Radio Shack cautions that you should not go from 10 characters per inch to proportional spacing and back in the same line. Part of the reason may be that the Line Printer IV will not respond to the change in the middle of a line. When executing a change from Electric Pencil you must set MØ (margin = 0). This is because the Electric Pencil adjusts the left margin with spaces.

Once these codes are in your Electric Pencil text, they can be printed just like any other Electric Pencil file. The text will look strange on those machines with the lowercase modification. The control characters display as some of those Greek characters you see if you have the lowercase switch on, but didn't load the software.

While the solution for controlling the Line Printer IV is not as handy as a control key in the program, it is effective and cheap. Once the most used sequences are put together, it is a quick and effective way to control the Line Printer IV without the expenditure or a great programming effort. ■

DEC	HEX	FUNCTION
13	(0D)	CARRIAGE RETURN
15	(0F)	START UNDERLINE
14	(0E)	STOP UNDERLINE
27;14	(1B;0E)	START DOUBLE-WIDTH LETTERS
27;15	(1B;0F)	STOP DOUBLE-WIDTH LETTERS
27;17	(1B;11)	SELECT PROPORTIONAL CHARACTERS
27;19	(1B;13)	10 CPI MONOSPACED CHARACTERS
27;20	(1B;14)	16.7 MONOSPACED CHARACTERS
27;28	(1B;1C)	1/2 LINE FEED UP
27;30	(1B;1E)	1/2 LINE FEED DOWN

Table 1. Control codes that can be used with the Line Printer IV.

✓See List of Advertisers on page 290

The Key Box

Centronics 737 line printer
48K RAM with 32K and 16K Modification

Better known as Radio Shack Line Printer IV.

Centronics 737

Arthur J. Welcher
360 S. Wetherly Drive
Beverly Hills, CA 90211

The Centronics 737, the Radio Shack Line printer IV, is an excellent printer with some beautiful features.

Most printers have a fixed space per letter format, either 10 characters per inch which, in the case of Line Printer IV, fills 80 columns on 8 1/2 inch-wide paper, or 16.7 characters per inch, which will produce 130 columns on the same width paper.

Proportional printing, however, is completely different from these formats. As the name would indicate, in proportional printing, the letters are of different width, according to the actual size of the character. The size is measured in "dot/width." The smallest, the lower case j, has a dot width of 6 dots, and the widest are the uppercase M and W, which have dot widths of 18. It is this format that gives proportional printing its deluxe appearance.

You will find in your printer manual that proportional printing may not be used with Scripsit if you want the right justification feature. The right justify feature is offered by many printers. It causes the right edge of the body of type to end on a vertical line, just as the left margin causes the body of type to begin on a vertical line. The result is a symmetrically looking printed page as pleasing to the eye as the pages of a book.

Now we have a word processor like Scripsit or, in my case, Pensadyne, which will justify monospaced letters, but is unable to justify the more pleasing proportional printed type face. Most of us would be inclined to use the proportional printing and forget the right hand justification option of the program. But wait! All is not lost. This conflict can be resolved with a little patience.

Although I use the Pensadyne Word Processor, the principles I will describe, and possibly even the program itself, should be

```
1 GOSUB4000:REM **** REPORT PRINTER ****
2 '
3 '
4 '
5 '  YOUR PROGPAM GOES HERE
6 '
7 '
8 '
190 END
1500 REM SUPER JUSTIFY
1510 F$=ST$
1520 F1$=" ":SC=0:TS=0
1530 F8=LEN(F$)
1535 IF F8<63 THEN1600
1540 FD=F8*11
1545 CD=0:    :CT=0
1546 K=VARPTR(F$):F9=PEEK(K)
1547 KH=PEEK(K+2):KL=PEEK(K+1)
1548 POKE &HEAAA,KH:POKE &HEAAB,(KL)   :POKE &HEAAC,F9
1550 C1=USR0(X9)
1560 C3=PEEK(&HEAAE):C4=PEEK(&HEAAD):CT=256*C3+C4
1590 CD=FD-CT
1600 REM
1610 FOR N4=1 TO F9
1630 LPRINT MID$(F$,N4,1);
1634 IF CD>CX THENLPRINTCHR$(27);CHR$(1);:CX=CX+1
1635 IF N4=>F9 THEN LPRINT CHR$(10)
1640 NEXT N4
1650 F$="":CX=0:CD=0:RETURN
4000 FOR I=0 TO 174
4010 READ A
4020 POKE(&HEA60)+I,A
4030 NEXT I
4040 DATA229,213,197,245,58,170,234,87,58,171,234,95,33,175,234,6
4042 DATA 0,58,172,234,79,62,0,50,174,234,26,214,32,205,158,234
4044 DATA183,128,71,62,0,23,197,71,58,174,234,128,50,174,234,193
4046 DATA 19,13,194,122,234,120,50,173,234,241,193,209,225,201,229,183
4048 DATA 133,111,62,0,23,132,103,126,225,201,0
4050 DATA              0,  0,  0,0,  7,7,10,15,12,16,14,7
4051 DATA7,7,12,12,7,12,7,12,12,12,12,12,12,12,12,12
4052 DATA12,12,7,7,12,12,12,12,14,16,15,14,16,14,14,16
4053 DATA16,10,14,16,18,18,12,16,14,14,15,12,14,16,16,18
4054 DATA16,16,10,12,12,12,12,12,7,12,12,10,12,12,10,12
4055 DATA12,8,6,12,8,16,12,12,12,12,10,12,10,12,12,16
4056 DATA12,12,10,10,7,10,12,0
4060  DEFUSR0 =&HEA60
4065 RETURN
```

Program Listing 1.

adaptable to any program that has been written in Basic.

The right justifier routines of most word processors work toward the same ends, and about the same way. The routine must first count the number of characters in a line, subtract that number from the maximum line length that you have pre-selected, then drop in an extra space between each word, usually starting from the left, until it has made up the difference between this line length and the maximum line length. The result is a justified right margin, even from top to bottom except for short lines that you choose not to have justified.

For proportionally printed letters the number of characters may add up to, say, 80 in a line (think of a space as another character). It will not necessarily have the same number of dot/widths as another line of 80 characters. For instance, on a typewriter that prints in monospaced characters, type a lowercase j 80 times and note where it ends on the page. Now type an uppercase W 80 times. You will find that the two lines are identical in length. If you make this same test with your printer in the proportional mode, you will find the j line to be one third the length of the W line.

To accomplish right justification in this format, we use the same logic, but completely different standards. We must now deal in dot/widths. We must know the maximum number of dots available to print a line across the page, and the total number of dots in each line to be printed. If we subtract one from the other, we know how many dot/widths are needed to fill each line out to the maximum. The printer has the ability to print a one-dot space, so we shall slip one in between each character until we use up the amount needed.

Next, add up the dot/widths. In the Line Printer IV manual, you will find a page containing the complete dot/width information for each ASCII character. We need to create a look-up table of each ASCII character and its corresponding dot/width, scan each line of copy, and add up the total dot/widths of the line.

I originally made the look-up table in Basic. Although it worked perfectly, it was extremely slow. It took about 20 seconds to process each line. I then turned to a friend who rewrote the line scan and look-up table into Assembly language. I translated it into a POKE routine so that it could be placed into a Basic subroutine for ease of entry. This speeded things up considerably. It now takes about 5 seconds per line, which is quite tolerable.

In the Pensadyne program, justification occurs after the body of text has been typed, but before it has been recorded on disk as a final text file. We do not "super justify" the text until the file goes to the printer. Since the Pensadyne Program justifies by adding full spaces to the lines, we have completely justified the text in the event we are printing monospaced letters, and we have coarse-justified if we are going to use proportional printing. We also know that we have the same number of characters per line, which then simplifies re-setting for dot/widths per line.

To justify while printing, line 1634 will add a one-dot space between each character until the correct amount has been added. CD is the number of dot/widths needed, CX is the number of one-dot spaces already dropped into the line, and CHR$(27);CHR$(1) is the one-dot space.

The point in your program where the line input is read into memory, centered, and sent to the printer, is the point of intercept. Instead of going to the printer, go to this program as a subroutine. In the Pensadyne program "TEXTPRIN/BAS" line 72 reads: 72 LPRINTTAB((CL-64/)2;ST$:N = N + 1. Change it to read: 72 LPRINTTAB((CL-64)/2;: GOSUB1500:N = N + 1. Make the same change in line 167 and you are all set.

Line 1510 sets F$ to be an identical string to the line input string ST$. VARPTR (F$) will give us the memory location of the length of F$ by PEEKing into this address +2 and this address +1. This results in a display of the most significant byte and least significant byte of the memory location of the start of F$. We now know where the line is stored in memory and how long it is.

Our machine language subroutine needs this information to know where to start the scan of addresses of the string. Go to the look-up table, add up the dot/widths of each character in the line, and place the total in locations EAAD and EAAE. Next, PEEK into these addresses and in line 1560 we have our total dot/width count for the line, CT.

We have already determined by trial and error that the number of maximum spaces (in Pensadyne, it's 64, which corresponds to the Radio Shack CRT format) multiplied by 11 (an average of the dot/widths) and have set this in line 1540 as the maximum number of dots that each line of text can be. Line 1610 and 1630 will scan the line for printing, line 1634 adds the necessary number of one dot spaces, and line 1635 adds a carriage return at the end of the justified line.

Line 1650 zeros everything and returns the operation to the word processor program for the next line. Line 1535 causes the line of text to go directly to the printer in the event of a short line that should not be justified.

This program has been written for a 48K computer. For a 32K computer, make following changes:

```
In data lines change all 234 to 170
In line 1548 EAAA to AAAA
             EAAB to AAAB
             EAAC to AAAC
In line 1560 EAAE to AAAE
             EAAD to AAAD
In line 4020 EA60 to AA60
In line 4060 EA60 to AA60
```

For a 16K computer, make following changes:

```
In data lines change all 234 to 106
In line 1548 EAAA to 6AAA
             EAAB to 6AAB
             EAAC to 6AAC
In line 1560 EAAE to 6AAE
             EAAD to 6AAD
In line 4020 EA60 to 6A60
In line 4060 EA60 to 6A60
```
■

✓See List of Advertisers on page 290

Using an Eddington Monkey makes a perfect puzzle.

Elementary, My Dear Primate

Richard C. Vanderburgh
9459 Taylorsville Road
Dayton, OH 45424

I was busy refining my language synthesis program when I came across the December 1980 issue of *80 Microcomputing* and the Dvorak Typewriter Keyboard article. Since my program requires typing in a lot of alphabetic text as input data, the Dvorak Keyboard looked like a winner.

This article explains what my program does, how the Dvorak or other non-standard keyboards can help, and how you can assemble and process a variety of language synthesis texts, and at the same time become proficient at non-standard keyboard typing.

Language Synthesis

A couple of years ago I came across the delightful text, *Scientific and Engineering Problem-Solving With The Computer,* in which Yale professor W.R. Bennett, Jr. introduces a variety of problems amenable to computer solutions. One of the more esoteric of these is related to the monkey-at-the-typewriter conjecture: A tireless monkey randomly pounding away at a typewriter would eventually type everything that was ever written.

The idea that great literary works might be generated by random processes goes back at least 300 years, as Professor Bennett notes, but he identifies Sir Arthur Eddington and his Gifford Lectures (given at Cambridge in 1927) as the source of most contemporary use of the concept. Hence his term "Eddington Monkey". However, the gap between possibility and probability is indeed large.

Bennett quantifies this gap by estimating that a monkey typing steadily at about ten characters per second would take 10^{36} years to punch out the first nine words of Hamlet's soliloquy, "To be or not to be, that is the question . . .". This zero order Eddington Monkey would generate a lot of other interesting and possibly profound one-liners during this enormous amount of time, but of course most of his output would be gibberish. So Bennett introduces the concept of higher-order Eddington Monkeys as a practical way to increase dramatically the sense to nonsense ratio.

First Order Eddington Monkeys

A first order monkey is provided with a customized typewriter equipped with many copies of the same key; lots of space bars, E's, and O's, but very few X's and Z's, all scattered about randomly on the keyboard. The number of copies of each key is determined by how many times the letter it produces appears in a selected text. Bennett uses Act III of Shakespeare's Hamlet as an example, noting that there are 6934 spaces, 3277 E's, and on through the alphabet to just 14 Z's. This adds up to a total of about 35224 characters, limited to the 26 letters of the alphabet, space, and apostrophe.

Fortunately a computer simulation is both realistic and fairly easy to accomplish. At first glance you might think that 35224 memory locations would be needed just to store numerical representations of all the keys—a requirement which would push even a 48K Model I TRS-80 to its limits. Not so. By applying a few mathematical maneuvers, Professor Bennett shows that the memory requirement can be reduced significantly.

Here is an attempt at a graphic explanation of how his scheme works: First we abandon the unwieldy typewriter, and build in its place a sturdy scale (the old fashioned two-sided balance type) equipped with two large baskets. Next we make 70448 plastic tokens, all of equal size and weight. We count out 6934 of these, put them in a net of negligible weight, and label it "space". Similarly we count out 3277 more tokens and put them in another net, marking it "E", and so on down to the 14 Z's, at which point we have 28 nets, each containing the number of tokens corresponding to the various characters that appear in the Shakespearean text. We then put the remaining

Contact the author for information on the text referred to and a copy of the LANGSYN1 disk with documentation.

The Key Box

Model I
32K RAM
1 Disk Drive

35224 tokens in a large box, and teach our Eddington Monkey to shovel a random number of these into one of the scales' baskets. He is then instructed to start loading the token-laden nets into the other basket until the scale tips. The character denoted by the label on the net which tips the balance is the one which is output (typed). Of course, to get ready to produce the next character, the monkey needs to remove all the nets, and empty the other basket's tokens back into the large box before repeating the cycle—not exactly an improvement over the typewriter method from his point of view, but as we shall see, suggests an efficient computer simulation.

Simulated First Order Eddington Monkeys

We begin by scoping out memory requirements, noting that the largest number we must deal with is 35224, and that the total number of variables and constants is about thirty. So we see at a glance that even a pocket TRS-80 will do nicely.

Now we compose an algorithm based on the scales-and-tokens analogy:

1. Pick an integer n in the range 1 to 35224.
2. Set s = 0 and i = 0.
3. Set i = i + 1, then set s = s + A(i), where A(1) = 6934 (the number of spaces), A(2) = 3277 (the number of E's)... A(28) = 14 (the number of Z's).
4. Repeat step three until s ≥ n, at which point i identifies the next character to be output, where space = 1, E = 2,...Z = 28.
5. Go to step one to output a new character, else terminate.

Second Order Eddington Monkeys

While an improvement over zero order Eddington Monkeys, first order ones still produce pretty unexciting results, so imagine the capabilities of a second order monkey. A pattern begins to emerge: zero order processing is completely random (or as random as we can make it), while first-order processing introduces a weighting factor. The more often a particular character occurs in a given text, the more likely it will be "randomly" selected. We can extend this pattern by causing the weighting to be based on the occurrences of paired characters, rather than on single ones processed individually. In our tokens-and-scales analog, we would label the nets spaceA, spaceB through spaceZ and EA, EB through EZ, etc., this time requiring 28 new nets for each of the original 28, for a total of 28 by 28 equaling 784 nets. We put as many tokens in a given net as there are occurrences of its labeled pair in the Hamlet text. For example, the letter A follows space 627 times, T follows space 962 times, and space follows E a record 1283 times. Quite a few nets will be empty (AA, ZB, etc) and in this scales-and-tokens analog you may dispense with them. However, in a computer simulation, you need to retain the corresponding empty bins for bookkeeping purposes.

Back to the second order monkey: We'll have him start off shoveling tokens randomly into one of the scale's baskets, just as his first order cousin did, but this time the total number of tokens initially in the large box is determined by the total count of how many times an arbitrarily selected character is followed by any character in the set. Space makes a good starter, since it is followed by a total of 6934 characters, with representatives from the whole set except for X, Z, and space itself (by design). We now direct the second order monkey to start piling on the 28 nets labeled spaceA, spaceB,—until the balance is tipped. Suppose this occurs with the spaceT net. T is the effectively typed character. We then add up all the T successor counts (the sum of all the tokens in the 28 TA, TB through TZ nets) and put this many (2557) tokens in the large box as the pool from which the next random sampling of tokens is taken. Now the monkey is directed to the 28 TA, TB through TZ nets, and he piles these into their side of the scales until the random pile is balanced. Chances are that this will be the TH net, since it has the most tokens (878) in it.

✓See List of Advertisers on page 290

-80 SPACE RAIDERS
Starship Simulator
and Combat Game

This is the most realistic combat space simulation available for any computer system anywhere! Operator is placed in a rapidly moving ship looking out through a window into the depths of space. Thousands of stars appear in the distance and part toward the screen edges as the ship flashes past. Forward and aft views available at any time. Accurate celestial navigation using both XYZ and spherical Coordinates! Comment from Aug '81 *80 Microcomputing*: "May be the ultimate Star Trek type game and, possibly, the most complex microcomputer game ever attempted"! Fast action graphics never stop!

Now available for Model III!

$24.95 (MasterCard, VISA, check, etc.) includes shipping. COD OK

Specify Model I or III, Level 2 16K

Bosen Electronics
445 E. 800 N.
Spanish Fork, UT 84660
(801)-798-9553 ✓416

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 + POSTAGE & HANDLING

DOCUMENTATION ONLY $10.00 DEDUCTIBLE ON PURCHASE

DISK SORT

- MENU DRIVEN
- RANDOM FILES
- CHAINS TO SYSTEM OR BASIC PROGRAMS
- EASY TO USE
- FAST!
- SPECS SAVED ON DISK
- YOU DON'T HAVE TO BE A PROGRAMMER TO USE IT!
- EASY TO INSERT INTO JOB STREAM FOR NON STOP RUNNING!

$69.99 + 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 + 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.

Simput® ✓288

UTILITY PROGRAM FROM HIKARI GROUP

SIMPUT is a flexible random access program for TRS 80 Model I* that lets you manage extremely complex data with great simplicity. It allows you to create records from one byte to more than 85,000 bytes long, with as many subrecords as you need, and it can be used with Microsoft's Basic Compiler.

SIMPUT has been tested to be compatible with TRSDOS* (can be compiled), DBLDOS* (also can be compiled), NEWDOS+, NEWDOS 80*, NEWDOS 80 V.2*, and NEWDOS 80, with Double-Zap/II*

Available on Disk
$29.95
(206) 382-6632

VISA MasterCard

*TRS 80 MODEL I, TRSDOS TMS, TANDY CORP. DBLDOS TM, PERCOM DATA CO., NEWDOS+, NEWDOS 80, NEWDOS 80 V.2 TM'S, APPARAT, INC.; DOUBLE-ZAP/II TM, SOFTWARE, ETC.

Hikari Group 3032 Fourth Avenue West
Seattle, Washington 98119

TECO MONITORS

RATED BEST BUY BY THE COMPUTER SHOPPER

NOW AN EVEN BETTER BUY

B & W MONITOR HIRES	119.95
GREEN PHOSPHOR HIRES	129.95
GREEN PHOSPHOR ULTRARES	144.95
TRS-80 CABLE	5.00
5¼" DISK DRIVES WITH PS & CASE	289.95

FROM

THE MAINE SOFTWARE LIBRARY
P.O. BOX 197
STANDISH, ME 04084

CHECK, MO, COD (20% DEPOSIT ON COD) ADD $10 SHIPPING FOR MONITOR, $6 FOR DRIVES

LNW-80-OWNERS

WE HAVE SOME TERRIFIC IDEAS AND PRODUCTS TO MAKE THIS GREAT COMPUTER EVEN BETTER. WRITE TO US FOR PACKAGE BEFORE YOU BUILD.

✓268

SOFTWARE YOU CAN BET ON!

OPTION80

If you gamble in the big casino – Wall Street – you need a program to evaluate call options. OPTION80 tells you when it's time to buy and when it's time to sell. Suggests hedges to minimize your risk. Takes only 15 minutes a week to update and run. Comes with data for 25 calls. Easily expanded to include as many as you wish. Sold only on diskette.

Only $25.00

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. Now 30% faster.

Only $15.00

TOUT3 and DRAW5 supplied on 500 baud cassette. Add $4.00 for Model I or III diskette. All programs available on 8" disk for Model II at $6.00 additional. PA residents add 6% sales tax. North Star diskette version of DRAW5 can be obtained from Real World Simulations, Box 4107, Torrance, CA 90510.

WILSON
Software Division • 539 Springhouse Lane • Camp Hill, PA 17011

TRS-80 is a Tandy Corporation trademark.

✓279

CASM Z80* EDITOR/ASSEMBLER

TRS-80* Model I or Model III

- SOURCE FILES CREATED AND EDITED IN RAM
- SOURCE FILES ASSEMBLED FROM DISK
- CHAINING OF SOURCE FILES PERMITTED
- MACHINE LANGUAGE OBJECT CODE STORED IN RAM OR TO DISK

*TRS-80 is a trademark of Tandy Corp
Z80 is a trademark of Zilog Corp

POWERFUL DISK ASSEMBLER

MINIMUM REQUIREMENTS:
32K OF RAM
ONE DISK DRIVE

COMPREHENSIVE INSTRUCTION MANUAL
CASM W/MANUAL $99
CASM MANUAL ONLY $20

CROWN MicroProducts ✓108

P.O. Box 892 • Marysville, WA 98270 • (206) 659-4279

DEALER INQUIRIES INVITED

Here is what a second order algorithm might look like:

1. Set m equal to the sum of: space(1), space(2),. . . space(28).
2. Pick an integer n in the range 1 to m.
3. Set s = 0 and i = 0.
4. Set i = i + 1, then set s = s + space(i), where space(1) = 627 (the number of times the letter A follows space), etc.
5. Repeat step 4 until s≥n, at which point i identifies the next character to be output, where A = 1, B = 2, space = 27, and apostrophe = 28.
6. Set m equal to the sum of i(1), i(2), . . . i(28).
7. Pick an integer n in the range 1 to m.
8. Set s = 0, and j = 0.
9. Set j = j + 1, then set s = s + i(j), where i(1) is the number of times the letter A follows the letter identified by the numerical value of i.
10. Repeat step 9 until s≥n, at which point j identifies the next character to be output.
11. To output another character, set i = j and go to step 6; else terminate.

I intentionally made this algorithm longer than necessary in order to show the special nature of the first time through (steps 1-5) contrasted with the more general form (steps 6-11). Note that the main feature of second order processing is the same as for first order: One by one, the contents of up to 28 bins are summed until a number is reached that equals or exceeds an arbitrary integer in the range between one and the sum of the contents of all 28 bins. It's just that more bookkeeping is involved in second order processing.

Third Order Eddington Monkeys

Well, if you try out second order processing, you find a few scattered "real" words—an improvement over first order processing, but still nothing to write home about. So, onward to third order algorithm generation for which follows the second and first order patterns, with one more array dimension to track. It turns out that if we limit the maximum bin count to 255, a third order machine language program fits nicely in a 32K Model I TRS-80 with a single disk drive. The 28 by 28 by 28 equals 21952 bytes required by all the bins for the AAA, AAB . . . counts leave enough room for the main program, data scratch, and fast (DOS independent) disk I/O. And all this fits on a non-DOS 35-track disk along with four pre-stored bin-count arrays.

Text Input

Before we delve further into what the main program does, let's take a look at how best to convert text into bin counts. Ideally, we want to convert each triplet of characters input via the keyboard into an increment to the corresponding bin in the 21952-element array, and be able to do this at stenographic speeds. This is not possible in Basic, but machine language is so fast, you still need a keyboard debounce routine to eliminate the pesky repetitions. Some of the things you want your input processing to do are: 1) Lock out keys not in the prescribed character set, 2) Provide error correction (backspace rubout, with corresponding bin-decrement), 3) Provide fast increment/decrement of the appropriate bin, and 4) Tell the user when a bin has been filled. For most English texts, this is the spaceTH bin after about 14000 characters have been input.

Keyboard Choice

This brings us to a fifth feature to enhance text input processing: Provide a choice of keyboards (via software). The Dvorak Keyboard opens the door to a wide range of possibilities, since changing the apparent location of any key is simply a matter of changing a table entry. While the Dvorak configuration may be the best under certain circumstances, other configurations may turn out to be better under other conditions. In any case, typing in text for language synthesis provides a good opportunity to try out various keyboard configurations on a variety of alphabetic texts by a variety of people, in a variety of languages.

Keyboard Optimization

An organized approach to optimization can help produce the best keyboard configuration for specific types of text to be typed: 1) The individual should rate key positions by ease of activation, 2) From a representative sample of expected text, determine character and character-pair frequencies (grist from the first and second order monkey mills), 3) Allocate the most often used characters to the preferred key positions, alternating hand-sides where character-pair frequencies are high.

What Good Is Language Synthesis?

Apart from honing your typing skills, you may be wondering by now what the point is in trying to synthesize language "randomly." Well, with third order processing, certain detected patterns identify individual author style, as well as the language being used. New pronounceable words may suggest themselves, along with new variants on old words, both kinds of which might well be introduced into an evolving language by innovative writers. Professor Bennett covers a broader range of applications, leading the reader of his book into related areas such as entropy in language, and language encryption, and I recommend his text to the curious.

The LANGSYN1 Program

This program along with four prestored text-statistics arrays is available on a standard 35-track self-loading floppy disk. Stored text can be loaded and processed, outputting 64-character lines to the screen at about one line per second, with optional printing. New text can be generated, or previously stored partially completed text conversion added to in either a standard or Dvorak Keyboard configuration. Disk I/O is optimized, requiring only a few seconds to load/save the 21952-byte array from or to one of the four designated disk areas.■

✓See List of Advertisers on page 290

The Key Box

Basic Level II
Model I
16K RAM

Discover the real meaning of those abbreviated error messages.

Error Code Expanded

Roger C. Alford
2633 Braeburn Circle
Ann Arbor, MI 48104

When I decided to upgrade my TRS-80, I thought about the things that irritated me most, and began making improvements. I added a 2716 EPROM (Erasable, Programmable, Read Only Memory) to my computer so I would always have a few useful routines at my disposal. The routines I chose were a keyboard debounce routine, a short terminal program for my RS-232 port, a small machine code monitor, and an error message expansion routine.

The two-letter Level II error abbreviations can be very irritating if one does not have all 23 of them memorized. Therefore, I found a way to expand the Level II error codes into *readable* error messages.

The 23 Level II error codes and messages—listed on page B/1 of the Level II Reference Manual—are the same ones I used in my program, except the L3 error reads "Disk Basic Feature," instead of "Disk Basic Only." Note only the messages on the screen are changed; if an error occurs in a program, the line number is still printed after the error message, and syntax errors still invoke the editor.

The listing with this article shows the program used to expand the Level II error codes. It is addressed to take up the top 485 bytes in a 16K TRS-80, but can be moved to other locations to meet various user requirements. For the computer owner using it, the memory size should be set to 32282.

The operating principle of the program is simple. The error code is in the E register when the Error routine is called. To arrive at the actual code (listed on page B/1 of the reference manual), the value in the E register must be divided by two (RRCA) and increased by one (INC A). The IX register is then set at a value of two less than the beginning of the table which contains the addresses of the messages. These addresses are in the same order as the error codes on page B/1 of the reference manual. By looping the number of times indicated by the actual error code value and increasing the IX register twice during each loop, the IX register will point to the address of the desired error message. This address is loaded into the HL register pair and the error message is printed. The current line number is then loaded into HL, and control is returned to the TRS-80 ROM. ■

Program Listing

```
7E1B          00100          ORG     7E1BH    ;=32283
              00105 ;
28A7          00110 PRMSG    EQU     28A7H    ;TRS-80 PRINT MESSAG
E ROUTINE
1A11          00120 ERRCNT   EQU     1A11H    ;ERROR ROUTINE CONTI
NUE
40A2          00130 CURLIN   EQU     40A2H    ;CURRENT LINE #
06CC          00132 BASIC    EQU     06CCH    ;BASIC SOFT ENTRY PO
INT
41A6          00134 INTCPT   EQU     41A6H    ;ERROR ROUT. INTERCE
PTION
              00136 ;
              00138 ;INITIALIZATION OF ERROR INTERCEPTION
              00140 ;
7E1B 3EC3     00150 ERRMSG   LD      A,0C3H           ;'JP'
7E1D 32A641   00160          LD      (INTCPT),A       ;INTERCEPT E
RROR ROUT.
7E20 21297E   00170          LD      HL,ERROR
7E23 22A741   00180          LD      (INTCPT+1),HL
7E26 C3CC06   00190          JP      BASIC            ;RETURN TO B
ASIC
              00200 ;
              00202 ;ERROR SERVICE ROUTINE
              00204 ;
7E29 E1       00210 ERROR    POP     HL
7E2A FB       00220          EI
7E2B 7B       00230          LD      A,E              ;GET ERROR C
ODE
```

Program continues

Program continued

```
7E2C FE2D       00240           CP      2DH             ;VALID?
7E2E 3802       00250           JR      C,NXT1          ;YES, REAL C
ODE
7E30 3E26       00260           LD      A,26H           ;NO, UNPRINT
ABLE ERROR
7E32 0F         00270 NXT1      RRCA                    ;DIVIDE BY 2

7E33 3C         00280           INC     A               ;ADD 1
7E34 DD214C7E   00290           LD      IX,ERRTBL-2     ;PREP. TO LO
OK-UP ADDR
7E38 3D         00300 LOOP      DEC     A               ;DEC. ERROR
CODE
7E39 DD23       00310           INC     IX              ;INC ADDR TB
L PTR.
7E3B DD23       00320           INC     IX
7E3D 20F9       00330           JR      NZ,LOOP         ;LOOP IF NOT
 ZERO
                00332 ;
                00334 ;DONE LOOPING, PRINT ERROR MESSAGE AND RETUR
N TO ROM
                00336 ;
7E3F DD6E00     00340           LD      L,(IX)          ;GET ADDRESS

7E42 DD6601     00350           LD      H,(IX+1)
7E45 CDA728     00360           CALL    PRMSG           ;PRINT ERROR
 MSG
7E48 2AA240     00370           LD      HL,(CURLIN)     ;GET CURRENT
 LINE
7E4B C3111A     00380           JP      ERRCNT          ;RETURN TO T
RS-80 ROM
                00390 ;
                00400 ;ERRTBL IS A TABLE OF ADDRESSES FOR THE 23 E
RROR
                00402 ;MESSAGES THE ADDRESSES ARE IN THE NUMERICAL
 ORDER OF THE
                00404 ;ERROR CODES USED BY THE TRS-80; I.E., THE A
DDRESS NF IS
                00406 ;FIRST AND THE ADDRESS FOR L3 IS LAST.
                00420 ;
7E4E 7C7E       00430 ERRTBL    DEFW    NF
7E50 8D7E       00440           DEFW    SN
7E52 9A7E       00450           DEFW    RG
7E54 AF7E       00460           DEFW    OD
7E56 BB7E       00470           DEFW    FC
7E58 D17E       00480           DEFW    OV
7E5A DA7E       00490           DEFW    OM
7E5C E87E       00500           DEFW    UL
7E5E F77E       00510           DEFW    BS
7E60 0E7F       00520           DEFW    DD
7E62 227F       00530           DEFW    D0
7E64 337F       00540           DEFW    ID
7E66 427F       00550           DEFW    TM
7E68 507F       00560           DEFW    OS
7E6A 647F       00570           DEFW    LS
7E6C 747F       00580           DEFW    ST
7E6E 8F7F       00590           DEFW    CN
7E70 9E7F       00600           DEFW    NR
7E72 A87F       00610           DEFW    RW
7E74 BD7F       00620           DEFW    UE
7E76 CF7F       00630           DEFW    MO
7E78 DF7F       00640           DEFW    FD
7E7A ED7F       00650           DEFW    L3
                00660 ;
                00670 ;THESE ARE THE ERROR MESSAGES
                00680 ;
7E7C 4E         00690 NF        DEFM    'NEXT WITHOUT FOR'
7E8C 00         00700           DEFB    0
7E8D 53         00710 SN        DEFM    'SYNTAX ERROR'
7E99 00         00720           DEFB    0
7E9A 52         00730 RG        DEFM    'RETURN WITHOUT GOSUB'
7EAE 00         00740           DEFB    0
7EAF 4F         00750 OD        DEFM    'OUT OF DATA'
7EBA 00         00760           DEFB    0
7EBB 49         00770 FC        DEFM    'ILLEGAL FUNCTION CALL'
7ED0 00         00780           DEFB    0
7ED1 4F         00790 OV        DEFM    'OVERFLOW'
7ED9 00         00800           DEFB    0
7EDA 4F         00810 OM        DEFM    'OUT OF MEMORY'
7EE7 00         00820           DEFB    0
7EE8 55         00830 UL        DEFM    'UNDEFINED LINE'
7EF6 00         00840           DEFB    0
7EF7 53         00850 BS        DEFM    'SUBSCRIPT OUT OF RANGE'
7F0D 00         00860           DEFB    0
7F0E 52         00870 DD        DEFM    'REDIMENSIONED ARRAY'
7F21 00         00880           DEFB    0
7F22 44         00890 D0        DEFM    'DIVISION BY ZERO'
7F32 00         00900           DEFB    0
7F33 49         00910 ID        DEFM    'ILLEGAL DIRECT'
7F41 00         00920           DEFB    0
7F42 54         00930 TM        DEFM    'TYPE MISMATCH'
7F4F 00         00940           DEFB    0
7F50 4F         00950 OS        DEFM    'OUT OF STRING SPACE'
7F63 00         00960           DEFB    0
7F64 53         00970 LS        DEFM    'STRING TOO LONG'
7F73 00         00980           DEFB    0
7F74 53         00990 ST        DEFM    'STRING FORMULA TOO COMPLEX'

7F8E 00         01000           DEFB    0
7F8F 43         01010 CN        DEFM    'CAN'
7F92 27         01020           DEFB    27H
7F93 54         01030           DEFM    'T CONTINUE'
7F9D 00         01040           DEFB    0
7F9E 4E         01050 NR        DEFM    'NO RESUME'
7FA7 00         01060           DEFB    0

7FA8 52         01070 RW        DEFM    'RESUME WITHOUT ERROR'
7FBC 00         01080           DEFB    0
7FBD 55         01090 UE        DEFM    'UNPRINTABLE ERROR'
7FCE 00         01100           DEFB    0
7FCF 4D         01110 MO        DEFM    'MISSING OPERAND'
7FDE 00         01120           DEFB    0
7FDF 42         01130 FD        DEFM    'BAD FILE DATA'
7FEC 00         01140           DEFB    0
7FED 44         01150 L3        DEFM    'DISK BASIC FEATURE'
7FFF 00         01160           DEFB    0
7E1B            01170           END     ERRMSG
00000 TOTAL ERRORS
```


✓See List of Advertisers on page 290

Convert uppercase to lowercase with this program.

Lots of Little Letters to Litter Your Listings

John R. Olsen, Jr.
P.O. Box 181
Newberg, OR 97132

While more and more TRS-80 owners are adding upper/lowercase to their computers, most of their old software still displays all capital letters, and that's just not right!

One old and tedious solution is to retype every program in lowercase. I have an easier way! This program converts all print statements to lowercase. It does not convert the program logic to lowercase, only the alphabetical characters inside quotes.

It will work with the Radio Shack lowercase modification or any other mod that recognizes the standard ASCII codes for lowercase letters.

Correct capitalization is sometimes a problem. Different programming styles make it impossible to capitalize every sentence properly. You may have to edit the modified program to capitalize the rest.

Since every letter inside quotes will be converted to lowercase, all string comparisons will also be modified. For example, the YES in IF A$ = "YES" THEN 30 will be changed to lowercase. You may or may not want that capitalized.

ASCII disk files convert the program. First, save your target program on disk in ASCII. Then, run this lowercase program. It reads each line of your target program from disk, one at a time.

The conversion changes every letter of the alphabet that is enclosed in quotes to lowercase. If the letter is preceded by a period and two spaces, then the letter is left capitalized; the program assumes that this is the beginning of a sentence.

The line of your target program is then saved back to disk, and the next line is read.

If you don't want to modify some portion of your program, save just that portion under a different name on the disk. Modify the entire program to lowercase, and then merge the previous portion with the modified program.

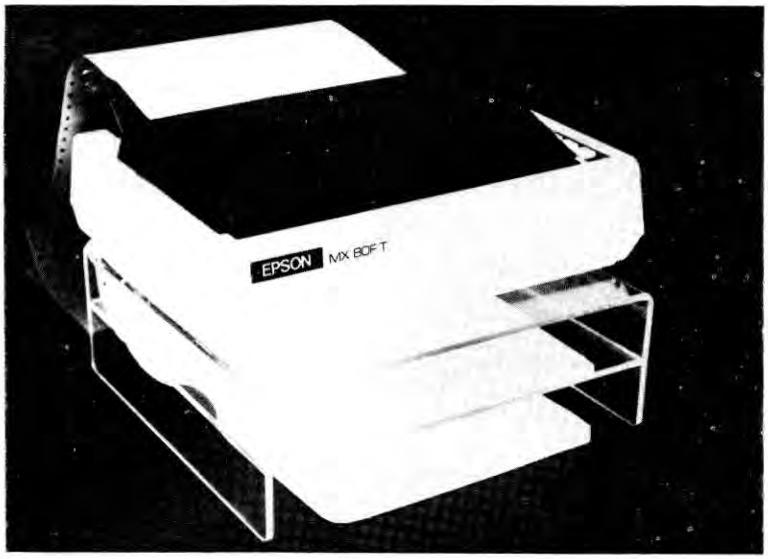
Your software should run exactly as before, but with the added advantage of showing off your lowercase modification. ■

The Key Box

Basic Level II
Lowercase modification

```
20 CLEAR 900
30 ON ERROR GOTO 620
40 DEFINT A-Z
50 A$=CHR$(34)
60 CLS
70 PRINT @ 20,"Lower Case Conversion"
80 PRINT @ 88,"by John Olsen"
90 PRINT
100 PRINT"This program will modify your BASIC program so that ev
erything"
110 PRINT"that appears on the screen is in lower case (requires
the"
120 PRINT"necessary hardware change).  To use this program you m
ust:"
130 PRINT
140 PRINT"1. Load your program."
150 PRINT"2. Save it back on disk in ASCII format ("A$"filespec"
A$",A)."
160 PRINT"3. The disk must have enough space free for a 2nd ASCI
I copy."
170 PRINT"4. Run this program."
180 PRINT
190 INPUT"Are you ready for step 4";L$
200 IF LEFT$(L$,1)="Y" THEN 220
210 END
220 CLS
230 INPUT"What's the filespec of the program to be modified";F1$
240 PRINT
250 PRINT"As your program is modified, it will be saved back on
disk"
260 PRINT"under a different filespec, in modified form."
270 PRINT
280 INPUT"What's the filespec of the modified program";F2$
290 IF F2$=F1$ THEN PRINT"Must be a different filspec.": GOTO 28
0
300 Q=0
310 OPEN"I",1,F1$
320 IF EOF(1) THEN CLOSE: GOTO 360
330 LINE INPUT#1,L$
340 Q=Q+1
350 GOTO 320
360 CLS
370 PRINT @ 0, Q"lines left to modify"
380 OPEN"I",1, F1$
390 OPEN"O",2, F2$
400 IF EOF(1) THEN 680
410 LINE INPUT#1,L$
420 N=LEN(L$)
430 P=1
440 I=INSTR (P,L$,A$) + 1
450 IF I=1 THEN 540
460 J=INSTR (I,L$,A$) - 1
470 IF J=-1 THEN J=N
480 FOR Z=I TO J
490 A=ASC (MID$(L$,Z,1))
500 IF (A>64) * (A<91) THEN MID$(L$,Z)=CHR$(A+32)
510 NEXT Z
520 P=J+2
530 GOTO 440
540 P=1
550 I=INSTR (P,L$,".  ")
560 IF I=0 THEN PRINT#2, L$: Q=Q-1: PRINT @ 0, Q: GOTO 400
570 IF I>N-3 THEN I=0: GOTO 560
580 A=ASC (MID$(L$,I+3,1))
590 IF (A>96) * (A<123) THEN MID$(L$,I+3)=CHR$(A-32)
600 P=I+4
610 GOTO 550
620 CLS
630 IF ERR<>106 PRINT"Error" ERR/2+1 "at line" ERL: END
640 PRINT"There is no file on this disk named " A$F1$A$"."
650 PRINT"Please check and try again."
660 PRINT
670 RESUME 230
680 CLOSE
690 PRINT @ 320,"The modified program is being loaded.  you shou
ld edit it to"
700 PRINT"get capital letters.  Then save a copy on disk (it wil
l load"
710 PRINT"faster than the ASCII version)."
720 LOAD F2$
730 END
```

Program Listing



PRINTER STANDS

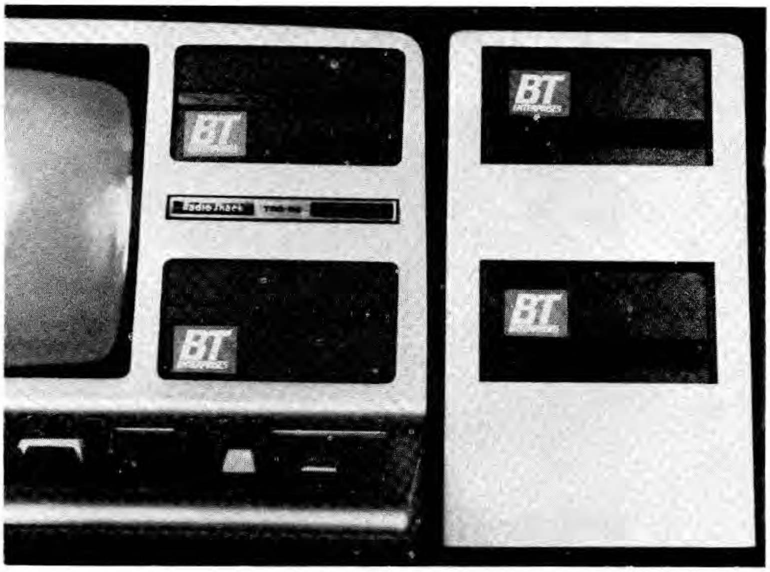
End the mess of paper from your desktop printer. Paper is fed from under the printer and allowed to stack neatly behind your machine. Made from the highest quality clear plexiglass, this is a handsome addition to home or office. Available in two sizes - Regular for printers such as MX-80, Microline80, etc. or Large for printers such as MX-100, DataRoyal, DWII etc. Available with extra shelf for quick and easy forms change.

Regular **$24.95**
Large **$29.95**
Extra Shelf . . . . **$ 9.95**

MODEL III EXTERNAL DISK DRIVE CASE

When you add disk drives to your Model III TRS-80, put them in the case that was designed for the Model III. This drive case is essential for the office where novice users can become confused with a combination of horizontal and vertical drives. Comes complete with mounting hardware and power supplies for two disk drives and 2 drive cable. Available with and without drives.

Case without drives **$179.95**
Case w/1 40 track drive **$429.95**
Case w/2 track drives **$654.95**



MX-80 RIBBON RELOAD

Replace your worn MX-80 Ribbon with a new, fresh high quality black cloth ribbon. Easy to load in your cartridge. You save over $10.00 compared to purchasing a new cartridge! Complete with easy to follow instructions. Each ribbon individually sealed in plastic bag for long life.

$3.95 each, 12 for **$39.95,** 25 for **$74.95**

(COMING SOON . . . COLOR RELOADS!!!)

MODEL III DISK UPGRADE

Upgrade your cassette Model III to disks with this easy to use controller kit from Micro Mainframe! Kit is complete with Controller Board, Switching Power Supply (for 2 internal drives), and complete mounting hardware (for 2 internal drives) and instructions. Just add drives! Supports single or double headed drives!

Complete kit **$279.95**

(CALL FOR LOW, LOW DISK DRIVE AND COMPLETE SYSTEM PRICES!)

BT ENTERPRISES

171 Hawkins Road
Centereach, New York 11720

(516) 981-8568 (Voice)
(516) 588-5836 (Data)
MNET-70331, 105

✓122

VISA MasterCard

Dealer Inquiries Welcome

Add $2.00 S & H
NYS res. add appr. tax

The Key Box

Basic Level II
Model I
16K RAM

Model I owners: Protect some of your screen from scrolling.

As the Screen Scrolls

M. Keller
13423 Desert Hills NE
Albuquerque, NM 87111

Imagine this scene: You are listing a large group of data items on your Model I, say from a disk file. The top of the screen displays the column titles. Below that, there are perhaps 13 or 14 lines of data, and on the bottom line there is a prompt message telling you to press Enter or some other key to display more items. When you press Enter the screen goes blank, the same column titles reappear, the next group of items is displayed, and so on . . .

```
                00100 ;THIS ROUTINE DOES A PROTECTED FIELD VIDEO SCROLL WITH
                00110 ;BOTTOM LINE AND 1-13 TOP LINES PROTECTED  SOURCE ADDRESS
                00120 ;FOR THE LDIR INSTRUCTION MUST BE PASSED FROM BASIC
                00125 ;
BB81            00130         ORG     48001   ;CHANGE THIS IF YOU LIKE
BB81 0000       00140 SOURCE  DEFW    0000    ;SOURCE ADDR  TO BE STORED HERE
4020            00150 CURVAL  EQU     16416   ;LSB OF CURSOR LOCATION KEPT HERE
3FC0            00160 LSTLOC  EQU     16320   ;LAST POSITION OF 15TH LINE
                00165 ;
                00170 ;------------ CALCULATE THE BYTE COUNT ----------------
BB83 CD7F0A     00180         CALL    0A7FH           ;GET SOURCE ADDR INTO HL
BB86 2281BB     00190         LD      (SOURCE),HL     ;SAVE IT AT  SOURCE
BB89 21C03F     00200         LD      HL,LSTLOC       ;READY HL FOR SUBTRACT
BB8C ED4B81BB   00210         LD      BC,(SOURCE)     ;SOURCE TO BE SUBTRACTED
BB90 ED42       00220         SBC     HL,BC           ;HL NOW HAS BYTE COUNT
BB92 44         00230         LD      B,H             ;TRANSFER THE
BB93 4D         00240         LD      C,L             ;    BYTE COUNT TO BC
                00242 ;
                00245 ;------------- CALCULATE DESTINATION ADDRESS -----------
BB94 2A81BB     00250         LD      HL,(SOURCE)     ;SRC TO HL FOR SUBTRACT
BB97 114000     00260         LD      DE,64           ;NO  TO BE SUBTRACTED
BB9A ED52       00270         SBC     HL,DE           ;HL NOW HAS DESTINATION
BB9C 54         00280         LD      D,H             ;TRANSFER THE DESTINATION
BB9D 5D         00290         LD      E,L             ;    ADDRESS TO DE
BB9E 2A81BB     00300         LD      HL,(SOURCE)     ;REINITIALIZE HL FOR LDIR
                00305 ;
                00310 ;----------- SCROLL & FOLLOW WITH CURSOR --------------
BBA1 EDB0       00320         LDIR                    ;SCROLL AREA UP ONE LINE
BBA3 3A2040     00340         LD      A,(CURVAL)      ;GET LSB OF CURSOR VALUE
BBA6 D640       00350         SUB     64              ;LESS 64 IS ONE LINE UP
BBA8 322040     00360         LD      (CURVAL),A      ;STORE NEW CURSOR VALUE
BBAB C9         00370         RET                     ;RETURN TO BASIC
0000            00380         END
00000 TOTAL ERRORS
CURVAL 4020 00150   00340 00360
LSTLOC 3FC0 00160   00200
SOURCE BB81 00140   00190 00210 00250 00300
```

Program Listing 1

Sound familiar? It's a common method of displaying large amounts of data on the Model I, but wouldn't it be nice if you could do it like this: The column titles show at the top of the screen and a prompt message shows on the bottom line. The "window" between these lines displays your data and scrolls through the window as long as you hold the down-arrow key, *without affecting the column titles or prompt message!* This is an example of the protected field concept. It is a built-in feature of the Model II and Model III, but you can do it on a Model I, and the results are pretty slick.

Theory of Operation

To do a selective scroll of this type, it is necessary to use machine language. Basic is too slow, and you would probably fall asleep by the time your display was updated. We have to replace the normal scrolling routine found in the TRS-80's ROM with a scrolling routine of our own in RAM. It turns out that the LDIR block transfer instruction of your Z80 chip is ideal for this task, since its sole purpose is to move a block of data from one section of memory to another. After all, the TRS-80 screen is merely a section of memory, and video characters are nothing more than data bytes.

Assume we only want to include lines 2–15 of the display in our scroll. This will leave the top and bottom lines to display more permanent information. Now, the LDIR is a "self-looping" instruction (it Loads, Increments, and Repeats), and it needs three pieces of information to do its thing. It requires that register pair HL be loaded with the source address. This is the memory address from which the first character will be grabbed. Register pair DE must contain the destination address where that first character will be deposited by the LDIR. Finally, register pair BC must contain the number of contiguous bytes to be moved in this fashion, or the *size* of the block to be moved.

Since we want every character in our window to be moved up one line, we load HL with the value 15488. This tells the LDIR instruction that the first character to be moved (character A in Fig. 1) is located at the beginning of line 3 on the screen. Loading DE with 15424 identifies the beginning of video line 2 as the place to which that character is to be transferred. Since we want to move a total of 13 lines with 64 characters each, we load BC with 832 and execute the LDIR instruction. This, in effect, moves each of the 832 characters "backwards" by 64 memory locations, or one video line. Thus, each line within our window moves up to the line above it. Voila! One protected field scroll. So much for theory . . .

Back to the Real World

In actual practice, we will probably call such a routine from different places in a Basic program, and the instruction sequence shown above would only work if we always wanted to scroll lines 2–15. Not much flexibility there, so I ended up with Program Listing 1. It allows us to pass a source address from a subroutine in Basic. It then calculates the byte count and the destination address, loads each register pair with the proper values for the LDIR instruction, and moves the block. The routine then moves the cursor up one line, so it will be in the right place to print our next item of data when we return to Basic.

Our next task is to write a subroutine in Basic that will allow us to use this protected field scroller in an already existing program. The subroutine has to be flexible enough so that the original Basic program will not need extensive modification.

Take a Sample

To see how all this might fit into a program, look at Program Listing 2. It is a sample Level II program with a conventional method for listing data on the TRS-80 screen. We will see how it operates in its original form, and then modify it to use our protected field scroller.

Before keying in the program in Listing 2, make sure you answer the memory size prompt with 32720. This way, some high memory will be available for the machine language program you will be POKEing. If you don't do this, there will be trouble with the modified version. When you run the program, some header information is printed on the top three lines. Then 12 fake check numbers, payees, account numbers and amounts are displayed, and you are prompted to press the down-arrow key to continue listing the data. Pressing the key clears the screen from line 4 to the bottom, then displays the next screenful, or "page", of our data, This is about as close as we can come to a protected field using only Basic. It's not bad, but we are going to give it some real style.

Metamorphosis

Let's modify the program so it can use our protected field scroller. Listing 3 shows the modified version and includes notations at the left margin to show which lines have been added or changed (there are no deletions). Type in these modifications and after saving a copy, run the new program. What you have now is a true scroll that does not disturb the prompt message on the bottom line or the header information on the top three lines. An immediately obvious advantage of this display method is that now you can study any one of the listed checks *in the context of the other checks surrounding it.* Also, you need not clear the entire screen just to display one additional check.

How Did We Do That?

Here is how our changes worked: New line 95 (a copy of old line 65040) has been inserted to display the prompt message. This is now done *before* printing the header information, to assure that the cursor will be properly located when it's time to print the first check information. New lines 65090 through 65140 contain a subroutine (called by line 25) which POKEs our machine language scroller into high memory. Line 45 has been added to tell the Basic interpreter where the routine can be found when it's time to use it.

Changed lines 65040 through 65060 contain the main subroutine, and bear further examination. (Both this subroutine and the one for POKEing the scroller were given large line numbers. This should make it easier to save them in a "subroutine library" for merging with existing programs.) The first line scans INKEY$ to see if you are asking for the program to return to a menu (remember, the prompt message is already on the screen). Line 65050 scans to see if you want to scroll the unprotected portion of the screen and list some more checks. This scan is continuous, and as long as you hold the down-arrow key the scroll will continue.

The scroller itself is activated by the USR statement in this line. A number is done on PF to calculate the source address needed by our scroller, and this address is passed to the routine in high memory. Line 65060 repeats the whole scanning process if you didn't press either key. Incidentally, if you're wondering about that gibberish in line 65030, it checks to see where the cursor is located. If the cursor has not yet reached the 15th line, there's no need to execute a scroll at all, so control is returned to the Print statement at line 160 for printing of the next check data.

Some Caveats

I wrote these routines to be as flexible as possible, but you will have to keep some things in mind when fitting them to your program:

1. You must initialize the variable PF as the number of top lines to be protected before calling the subroutine. PF must be a positive number from 0–13. If it is outside these limits, the results will be disastrous.
2. Your program must prepare the screen for the listing of the data. Make sure this preparation leaves the cursor in the right place for printing the first data item. Once you have done this, the subroutine will automatically prevent anything from being printed below line 14.
3. The subroutine works on the assumption that each data item is no more than one line in length. If your data items take up more space, you'll have to get

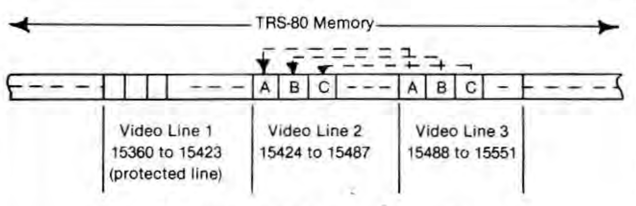


Fig. 1. LDIR Used for Video Scrolling.

```
10 '       DEMONSTRATOR FOR PROTECTED FIELD SUBROUTINE
20 '
30 '------------------ INITIALIZE VARIABLES -----------------
40 CLEAR200:DEFINT A,F,K,P,X:DIM FD(1,5Ø)
50 H$="CK.#    PAID TO:                        ACCT#          AMOUNT"
60 F$=" ##.    %                           %   ####          $####.##"
70 PRINT"   GENERATING FAKE DATA......"
80 FORX=1TO5Ø:FD(Ø,X)=RND(1ØØØ):FD(1,X)=RND(1ØØ):NEXTX:CLS
90 '------------ INITIALIZE SCREEN FOR LISTING -------------
100 PRINT@1Ø,"*******   PUT ANY MESSAGE HERE   *******"
110 PRINT H$
120 PRINTSTRING$(63,"-")
130 '------------------ LIST THE FAKE DATA -------------------
140 PF=3  'TOP 3 LINES WILL BE PROTECTED
150 FOR X=1 TO 5Ø: GOSUB 65Ø3Ø
160   PRINTUSINGF$;X,"PAYEE NAME HERE",FD(Ø,X),FD(1,X)
170 NEXT X
180 PRINT"END OF ROUTINE. MAIN PROGRAM WOULD CONTINUE HERE...."
190 GOTO 19Ø
65000 '*******************************************************
65010 '------------- PROTECTED FIELD SUBROUTINE ---------------
65020 '  MAIN PROGRAM MUST SUPPLY PF (NO. OF LINES TO PROTECT)
65030 IF PEEK(16417)*256+PEEK(16416) < 16256 THEN RETURN
65040 PRINT@97Ø,"PRESS: <";CHR$(92);"> TO CONTINUE, <M> FOR MENU
";
65050 A$=INKEY$:IFA$=CHR$(1Ø)THEN PRINT@PF*64,CHR$(31);:RETURN
65060 IF A$="M"THEN PRINT:GOTO 18Ø: ELSE 65Ø5Ø
65070 '*******************************************************
```

Program Listing 2

creative to prevent the Print statement from crashing into the bottom of the screen before it can be trapped by line 65030.

4. Line 15 of video is kept blank by the program in order to provide a cleaner separation between the check data and the prompt message. If you want to make line 15 available for printing an extra line of data, follow the Print statement in line 160 with a semicolon and make these changes:

```
150 FOR X = 1 TO 50
170 GOSUB 65030 : PRINT : NEXT X
```

5. Remember, the only way we can prevent Basic from doing its normal scroll of the entire screen and messing up our display is to see that we do not print anything at the bottom of the screen. We can get away with it in our prompt message only if the message is followed by a semicolon. And even there, we must reposition the cursor before printing anything else.
6. The routine can be used not only for printing data from an array, as in our sample program, but also for listing data from a disk file. And it is a dream come true for entering large amounts of data from the keyboard when you want some instructions to remain at the top of the screen. Of course, you won't need the whole subroutine for this particular application.
7. If you are operating under Disk Basic with 32K memory, answer the memory size question with 49100 and make the following program changes:

```
45 DEFUSR0 = &HBFCE 'DEFINE ENTRY
POINT FOR SCROLLER
65090 FOR X% = -16436 TO -16394
```

In line 65050, insert the character 0 between R and (.

Model Which?

There you have it—another sneaky trick to make your Model I behave like a more expensive machine. The next time you sit down to list a group of data items, imagine you're at the keyboard of a Model II. Well, maybe not quite . . . ■

```
10 '       DEMONSTRATOR FOR PROTECTED FIELD SUBROUTINE
20 '
25 GOSUB 65090  'POKE MACHINE LANGUAGE SCROLLER
30 '------------------ INITIALIZE VARIABLES -----------------
40 CLEAR200:DEFINT A,F,K,P,X:DIM FD(1,50)
45 POKE16526,214:POKE16527,127 'DEFINE ENTRY POINT FOR SCROLLER
50 H$="CK.#    PAID TO:                         ACCT#        AMOUNT"
60 F$=" ##.    %                           %    ####        $####.##"
70 PRINT"   GENERATING FAKE DATA......"
80 FORX=1TO50:FD(0,X)=RND(1000):FD(1,X)=RND(100):NEXTX:CLS
90 '------------ INITIALIZE SCREEN FOR LISTING --------------
95 PRINT@970,"PRESS: <";CHR$(92);"> TO CONTINUE, <M> FOR MENU";
100 PRINT@10,"*******   PUT ANY MESSAGE HERE   *******"
110 PRINT H$
120 PRINTSTRINGS(63,"-")
130 '------------------- LIST THE FAKE DATA -------------------
140 PF=3  'TOP 3 LINES WILL BE PROTECTED
150 FOR X=1 TO 50: GOSUB 65030
160   PRINTUSINGF$;X,"PAYEE NAME HERE",FD(0,X),FD(1,X)
170 NEXT X
180 PRINT"END OF ROUTINE. MAIN PROGRAM WOULD CONTINUE HERE...."
190 GOTO 190
65000 '*********************************************************
65010 '------------- PROTECTED FIELD SUBROUTINE ---------------
65020 '  MAIN PROGRAM MUST SUPPLY PF (NO. OF LINES TO PROTECT)
65030 IF PEEK(16417)*256+PEEK(16416) < 16256 THEN RETURN
65040 IF INKEY$="M"THEN 180
65050 KB=PEEK(14400):IFKB=16THEN A=USR(PF*64+15424):RETURN
65060 GOTO 65040
65070 '*********************************************************
65080 '-------------- MACHINE LANGUAGE SCROLLER ---------------
65090 FOR X% = 32724 TO 32766
65100 READ B% :POKE X%,B% :NEXT X%
65110 DATA 0,0,205,127,10,34,129,187,33,192,63,237,75,129,187
65120 DATA 237,66,68,77,42,129,187,17,64,0,237,82,84,93
65130 DATA 42,129,187,237,176,58,32,64,214,64,50,32,64,201
65140 RETURN
```

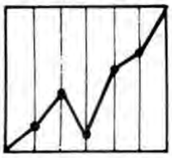
Program Listing 3

6 ways to give your TRS-80* a +PLUS

Exhaustive research and professional standards bring you the best in business software.

EASYTRAK™

KEEP YOUR SALES CURVING UPWARD WITH EASYTRAK™, a powerful, all-purpose activity tracking system. User-specified options enable you to monitor sales activity by territory, product line, salesman, or by combinations. Can monitor inquiries, sales calls and order booking. Production activity (by shift, by product, by day, by team, etc.), spending or income activity are simple to set up with EASYTRAK™. Extensive video displays make available individual, group or total summaries, relative averages, and percentage data all at the touch of a button. Use of an 80-column printer is optional.



MODEL I/III $125
MODEL II $150

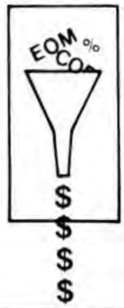
+ACCOUNTS RECEIVABLE

PUT MORE CASH IN YOUR BANK ACCOUNT with +Account Receivable's many collection/credit features, including average payment days, aging schedule and statements. Open-item system records sales activity this year and last, optional automatic discounts and salesman assignment. Cash receipts and A/R adjustments feature extensive on-line validation. And, yes, +Accounts Receivable handles the pesky new 9-digit zip code. Other optional features include service charges to selective customers; partial and "on-account" payments. Posts, at your option, to +General Ledger.

MODEL I/III $325
MODEL II $350

+ACCOUNTS PAYABLE

FUNNEL THAT STACK OF BILLS into +Accounts Payable system and create an orderly, manageable flow of vendor histories, cash requirements forecasts, check registers and checks, aging schedules, etc. Will accept manual checks, reversals, and adjustments. Pay by selected vendor, invoice, due date, discount dates, and optionally post directly to your +General Ledger. Supports 5-digit vendor numbers.



MODEL I/III $325
MODEL II $350

+PAYROLL

KEEP YOUR EMPLOYEES HAPPY and smiling with prompt, accurate and professional payroll records. Will handle weekly, biweekly, semimonthly, monthly payrolls, produce checks, check registers, labor reports, earnings reports, Form 941's, W2's, etc. Can handle fixed/voluntary deductions, hourly/salary employees, bonuses, city and state taxes, etc. Easy to use and update tax tables. Automatic +General Ledger posting.

MODEL I/III $325
MODEL II $350

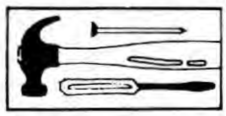
+GENERAL LEDGER

SLICE OUT MORE PROFIT by using +General Ledger to monitor your financial status. Features include fully-flexible financial statements, both budget and last year comparisons, special report options, monthly and quarterly trial balances, manual and/or automatic journal entries all with extensive audit trails. You structure the chart of accounts with up to 5 digit account numbers.

MODEL I/III $325
MODEL II $350

+FIXED ASSETS

A POWERFUL SET OF TOOLS TO CALCULATE and monitor fixed asset expenditures, book and tax depreciation, investment tax credit, etc. Will even forecast future depreciation for you. (Handy at budget time!) Code structure enables you to classify assets by departments, buildings, groups or other categories. Handles seven depreciation methods. Can post to +General Ledger for you.



MODEL I/III $325
MODEL II $350

SPECIAL! All of the above: MODEL I/III $1600. MODEL II $1750.

+PLUS SOFTWARE comes to you feature-packed from on-line validation to machine-language file handling. Extensive documentation and thorough installation guide accompany each order. All systems are furnished on disc. Source code is included. Texas residents add 5% sales tax.

VISA master charge

To order, call TOLL-FREE 24 hours
1-800-231-7315

+PLUS

PLUS COMPUTER TECHNOLOGY, INC.
POST OFFICE BOX 1152
ANGLETON, TEXAS 77515 (713) 849-1108 ✓131

Dealer inquiries invited

*TRS-80 is a registered trademark of Tandy Corp.
EASYTRAK is a trademark of Plus Computer Tech. Inc.

LOAD 80

Use the powers of your TRS-80 to perpetuate profits.

The Profit Prognosticator

R. B. Nottingham
1619 SE 3rd Court
Deerfield Beach, FL 33432

Program Listing

```
5 CLS
10 REM  1979 DATA. WEEK NUMBER, WEEK'S SALES.
20 DATA 19,1524,20,1707,21,2045,22,1903,23,1917,24,1196,
        25,2291,26,2008,27,1451,28,2099,29,1771,30,2337,
        31,2236,32,2163,33,2163,34,2172,35,2437,36,1734
30 DATA 37,1853,38,2016,39,2081,40,2602,41,2510,42,2373,
        43,2455,44,1816,45,2912,46,3091,47,2848,48,2510,
        49,2410,50,2382,51,3305,52,3615
40 REM 1980 DATA
50 DATA 1,2364,2,3478,3,4099,4,4656,5,4391,6,4163,7,4701,
        8,3852,9,5678,10,4053,11,4574,12,4482,13,4473,14,4428,
        15,4017,16,4446,17,3423,18,3286,19,3378,20,3003,
        21,3861,22,2389,23,3095,24,2885,25,3341,26,3022
60 DATA 27,3003,28,3341,29,3551,30,3560,31,3286,32,3980,
        33,3916,34,4053,35,22830,36,3332,37,3469,38,3798,
        39,3022,40,2629,41,2566,42,2939,43,3158,44,3222
70 DATA 45,2967,46,3907,47,3286,48,3998,49,3734,50,2839,
        51,3679,52,3962
80 REM 1981 DATA
90 DATA 1,5313,2,6089,3,4893,4,3889,5,5952,6,5541,7,5770,
        8,6290,9,5660,10,5914,11,6214,12,5231,13,5669,
        14,5258,15,4482,16,5176,17,4848,18,4108
100 REM LINES 20 THRU 70   ARE PREVIOUS YEAR'S DATA
110 CLEAR 1000 :PRINTTAB(20)"SALES FORECASTING":FOR X=1 TO 1000:
 NEXT
120 INPUT"WOULD YOU LIKE AN EXPLANATION OF THIS PROGRAM";A$
130 IF LEFT$(A$,1)="Y" GOTO 750
140 INPUT"WOULD YOU LIKE TO UPDATE PROGRAM";B$
150 IF LEFT$(B$,1)<> "Y" THEN 180
160 PRINT"EDIT LINE 20.  DELETE FIRST WEEK # AND WEEKS SALES."
170 PRINT"EDIT LINE 90. ADD LATEST WEEK NUMBER AND WEEKS SALES."
:END
180 RESTORE: CLS : DLR$="  $###,###."
190 DIM PW(52),PS(52),CW(52),CS(52)
200 PRINT TAB(25),"COMPUTING"
210 FOR X=1 TO 52
220      READ PW(X):REM PREVIOUS YEAR WEEK NUMBER
230      READ PS(X):REM    "          "  SALES
240      PT=PT+PS(X):REM   "          "  TOTAL
250 NEXT X
260 FOR X=1 TO 52
270      READ CW(X):REM CURRENT YEAR WEEK #
280      READ CS(X):REM    "         "  SALES
290      CT=CT+CS(X):REM   "         "  TOTAL
300 NEXT
310 PC=CT/PT   :REM % CHANGE SINCE LAST YEAR
320 WF=PS(52)*PC :REM FORECAST FOR WEEK
330 FOR X=1 TO 4
340      MT=MT+PS(X)  :REM TOTAL FOR CORRESPONDING FOUR WEEKS
350 NEXT
360 MF=MT*PC*1.08 : REM MONTHLY FORECAST
370 AF=CT*PC: REM ANNUAL FORECAST
380 A$="SOMETHING SHOPPE" : 'SUBSTITUTE NAME OF BUSINESS
390 H$="7000" : M$="3500": L$="0":'SUBSTITUTE APPROPRIATE VALUES
```

Program continues

No aspect of management is more vital than sales forecasting. All planning rests on the sales forecast. Profit or loss, success or failure depend on its accuracy.

I entered retailing 13 years ago, after many years in corporate management. I have since established and operated two successful retail businesses. Because I came to retailing with a technical and statistical background, my first business was probably the most over-managed small business in history. Investing most of my personal capital, I resolved that if I failed, it was not going to be due to any lack of diligence and management on my part.

All procedures were in writing, and I had a perpetual inventory system. But my greatest challenge was sales forecasting where a major problem is seasonal correction. The south Florida Gold Coast is more seasonal than most areas, and seasonal trends have been changing markedly over the years.

Until 20 years ago, this was strictly a winter resort community. Many retailers closed for the summer, remaining open only from November to Easter. More and more people began living there for at least nine months of the year. Many major corpora-

The Key Box	**Model III**
	Basic Level II

✓See List of Advertisers on page 290

KISS BASIC GOODBYE

Announcing the new quick and easy way to write custom business applications in hours instead of weeks.

At last, you can get a microcomputer business application development system that is designed for one purpose only . . . fast production of bug-free professional-looking custom business applications.

"Thinks" the Way Business Thinks
Quic-N-Easi is the revolutionary screen-format-oriented application development system that thinks in terms of transactions, records and fields. The same way business thinks. **Quic-N-Easi** expects an application to include custom formatted key entry. It lets you set up the character, field, and record validity checks business wants by merely filling in the blanks. It automatically signals in real time when errors occur and "explains" what is wrong in plain English messages you select for each custom application.

Much Faster than BASIC
Quic-N-Easi programming is much faster than BASIC because the standard business applications (key input, file handling, and output reporting) are handled via pre-programmed parameter driven subroutines. To produce finished professional looking custom business programs, you merely:

- Draw the business formats directly on the CRT in minutes.
- Fill in the blanks for field attributes, validations, tables, etc.
- Invoke the **Quic-N-Easi** interpreter to check fields against tables, check limits, access data files, and perform business calculations, etc.
- Optionally define file and output formats right on the screen.

Pays for Itself in a Week
By eliminating the coding drudgery of writing screen, field, file, and format programs in BASIC, programmers are free to concentrate on the unique business aspects of each custom application. No professional programmer can afford to develop one more business application without **Quic-N-Easi**. The savings are so significant with **Quic-N-Easi**, it can actually pay for itself in only one week.

Gives You Much More than a Screen Builder . . . the Only Complete Business Development System for Microcomputers
Quic-N-Easi handles the entire application development job from key entry to final output. You get everything you need, including detailed documentation to begin writing professional programs the first day. **You get . . .**

- A singularly capable **Quic-N-Easi** screen builder
- A comprehensive parameter driven **Quic-N-Easi** content editor
- A full **Quic-N-Easi** interpreter language
- A complete **Quic-N-Easi** file management system with Index Sequential, Random, and Sequential File Access Method
- A complete **Quic-N-Easi** print format handler
- A detailed **Quic-N-Easi** reference manual
- A **Quic-N-Easi** self-teaching guide
- A **Quic-N-Easi** quick reference card
- An interface to other program files

ORDER NOW—Don't Waste One More Day Coding BASIC
Your time is too valuable to waste hours on end writing BASIC code. The first week you use **Quic-N-Easi**, your finished business programs will look better, run better, and return more dollars to you. Don't delay. Order **Quick-N-Easi** today.
Phone . . . 215 968-0689

QUIC-N-EASI™
STANDARD MICROSYSTEMS INC.
136 GRANITE HILL COURT, LANGHORNE, PA 19047

Pricing Information
- Complete **Quic-N-Easi** system **$395**
- Manual only **$60**
- **Visa** and **Mastercard** accepted
- **Dealer Inquiries Invited**

Minimum System Requirements
Z80 • 48K • Floppy Disc
• CP/M (except TRS80 Mod III)

Other Disk Formats • 8" Single Density Vector Graphics • Micropolis Model 2

Customized Versions • TRS80 Mod II, TRS80 Mod III, APPLE, OSBORNE, INTERTEC, VECTOR , ZENITH

Apple, CP/M, Intertec, Micropolis, and TRS80 are trademarks of Apple Computer, Digital Research, Intertec Data Systems, Micropolis Corp., and Tandy Corp. ✓317

"I tried everything I knew to adjust for personal variations. . ."

tions—including IBM—opened plants in the area. A state university and three colleges are within 25 miles of my shop.

The area's seasonal pattern looks like this: From January to Easter, sales are constantly high. From Easter to July 4, sales stair-step down. From July 4 to Labor Day, sales are uniform, but less than winter sales. From Labor Day to the end of the year, sales climb. While this is the sales curve now, it is steadily becoming more uniform.

I tried everything I knew to adjust for seasonal variations, including studying historical trends exceeding five years. Then it suddenly occurred to me the best predictor of this year's sales is last year's sales. To forecast next week, next month, or the next 12 months, I took the ratio of the last 52 weeks to the previous 52, and used that ratio as the multiplier on the corresponding period of last year.

Program Listing 1 shows how I did this. For example, in line 310, PC gives the ratio, which in this case is 1.52. In other words, the last 52 weeks are running 52 percent better than the previous two. The last week shown is the 18th week. To forecast the next week, you look at the 19th week last year: Sales were $3378. Multiply that by 1.52 percent. Your forecast: $5147.

The program is written in Microsoft Basic for the TRS-80, but can be easily converted to any other language. It is liberally sprinkled with remarks making it self-explanatory. If you have sales records going back 104 weeks for your business, you can substitute them in the data statements. In line 390 adjust H string—the high chart limit—to suit your data. M string should be ½ of H string. These entries make the graph self scaling. As shown in line 460, a minus sign is used for the previous year's sales, and a plus sign is used for this year's sales. Lines 580–630 determine which will be printed first.

Every Saturday night, at the close of the week's business, it takes me about five minutes to fire up the computer, load the program and edit lines 20 (to delete the first week), and 90 (to enter the current week). Then I run the program, and have a graphic display of the sales for the last year plotted against the corresponding previous week, as well as the forecast for the year, month and week.

I have tried many other methods, and this is the most accurate I have found. Since this program works as well as it does in our area of extreme seasonal variation, it should work as well or better elsewhere. If you have data for your business for more than two years, enter older data and see how well it predicts what actually happened. You will be surprised with how well it works. ■

Program continued

```
400 CLS : J=0
410 H=VAL(H$) : M=VAL(M$) : L=VAL(L$)
420 Q$=L$+STRING$(27,"-")+M$+STRING$(27,"-")+H$
430 C=32 -(LEN(A$)/2) : 'CENTERS TITLE
440 PRINT@C,A$ : 'PRINT TITLE
450 PRINTQ$
460 PRINT"WEEK #  PREVIOUS SALES (-)
ALES (+)"
470 SF=62/(H-L) : 'SCALE FACTOR
480 FOR J=1 TO 52
490 N=X
500 K=Y : WP=X
510 D=H/2
520 K=(K-D)/D : 'SCALING
530 A=PS(J) : REM (READ VALUES TO BE PRINTED)
540 Y=CS(J) : IF CW(J)=0 GOTO 680
550 AV=(AV-D)/D
560 P=INT((Y-M)*SF)+32
570 AV=INT((A-M)*SF)+32
580 IF AV>P GOTO 620
590 PRINT CW(J);TAB(AV)"-";
600 PRINTTAB(P)"+"
610 GOTO 640
620 PRINT CW(J);TAB(P)"+";
630 PRINTTAB(AV)"-"
640 NEXT
650 GOTO680
660 INPUT"PRESS ENTER TO CONTINUE";:RETURN
670 PRINT@C,A$ : PRINTQ$ : RETURN
680 PRINT"FORECAST FOR YEAR=";:PRINTUSING DLR$;AF;
690 PRINT" WEEK=";: PRINTUSING DLR$;WF;
700 PRINT" MONTH=";:PRINTUSING DLR$;MF
710 PRINT"      (MONTH IS NEXT 4 1/3 WEEKS)"
720 END
730 INPUT" END OF DATA. PRESS ENTER FOR FORECAST";ZZ
740 GOTO 680
750 CLS:PRINT"IN ANY BUSINESS, A SALES FORECAST MUST TAKE INTO A
CCOUNT"
760 PRINT"SEASONAL TRENDS. EXPERIENCE HAS SHOWN THAT THE BEST"
770 PRINT"SEASONAL CORRECTION IS BASED ON THE PREVIOUS YEAR."
780 PRINT"THIS PROGRAM DOES THIS BY ASKING YOU TO STORE THE PREV
IOUS"
790 PRINT"SALES BY WEEKS."
800 PRINT"ANY FORECAST ASSUMES THAT THE FUTURE WILL FOLLOW PAST"

810 PRINT"TRENDS. IT NEVER DOES PRECISELY, BUT THIS FORECAST IS
AS"
820 PRINT"RELIABLE AS ANY WE KNOW AND CAN BE VALUABLE IN SHOWING
"
830 PRINT"SIGNIFICANT CHANGES IN SALES."
840 INPUT"      (PRESS ENTER TO CONTINUE)";ZZ$
850 GOTO 140
890 REM RUN 900 TO VERIFY DATA IF NEEDED.
900 FOR T=1 TO 104:READ W:READ S:PRINTW;S;:NEXT
```


✓See List of Advertisers on page 290

UTILITY

LOAD 80

> ***The Key Box***
>
> **Disk Basic**
> **Model I**
> **TRSDOS**
> **Printer**

Take a picture of those variables.

Snapshot

Robert Rice
P.O. Box 218
West Hurley, NY 12491

Have you ever written a program that had a bug in it? Well, I have. As a matter of fact, I don't think that I can claim to have ever written any bug-free programs.

In order to help track down and kill these bugs, I decided that it was about time to expend some effort to write some kind of utility program. I needed something more helpful than the TRS-80's trace capability. I needed something that could show me how the values of my variables were changing as my program proceeded through its execution. After all, these values often control the flow of a program. If they are not correct, results won't be what you expected. So I wrote a program which I call Snapshot.

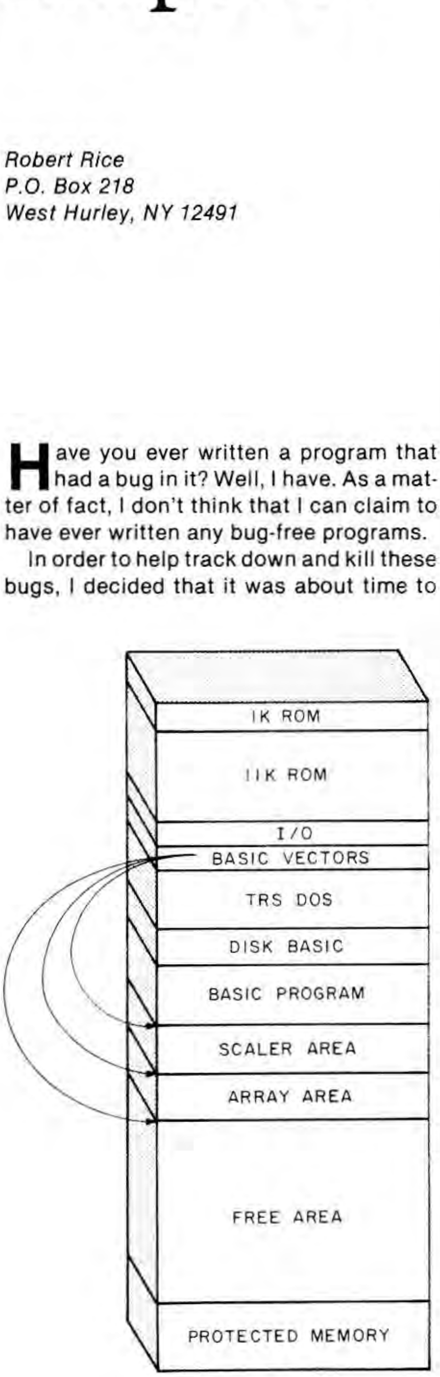


Fig. 1

So that you can appreciate the flavor of Snapshot, I will first tell you a little about it. Then, I'll describe how to use it and give some examples. In addition, I'll present some background material concerning Basic's variables and how they are stored in machine memory.

Program Description

Snapshot is written in Disk Basic for the Model I, Level II machine operating under TRSDOS with a line printer. Although a printer is a requirement for any serious work, the subroutine can be modified to dump to the video screen if you wish. The subroutine is designed to be appended to the Basic program you are developing when you get into trouble. It might even be a good idea to load Snapshot first and then develop your programs around it so that you'll be spared the trouble of trying to append it later.

The line numbers shown in the listing are deliberately small, to help keep the discussion as simple as possible. However, the code should be renumbered to some convenient values above 60000 so that it will not conflict with the code under development in case you want to append it to a partially developed program. If you have a renumbering facility, you can enter the code as shown and then let your computer handle the renumbering. Otherwise, you'll have to add, say 60000, to all the line numbers and their references as you key them in.

Although the subroutine appears rather formidable, only about half of the statements are comments. I have taken care not to branch to any of the comment lines; you can simply omit keying them in. If you are concerned about memory space, you may also want to remove spaces from the code and even make multiple-statement lines where possible. I would suggest that you first key it in as listed (with or without the comments) to assure yourself that it works. Then, begin compressing it if you wish.

In an effort to minimize name conflicts, Snapshot uses only one name: ZZ. Of course, ZZ is used in almost all of its forms: ZZ%, ZZ!, ZZ#, ZZ$, and the arrays ZZ!() and ZZ$() as variables; and ZZ%, ZZ! and ZZ$ as functions. Thus, if you avoid using ZZ in your application program, you will avoid any name conflicts with the SNAPSHOT subroutine. Using a single name means that the resulting code is not as compact as it could be because an array is used for storage. But that is a small concession for the freedom of use that is gained.

How to Use Snapshot

Snapshot can be called from any line of your Basic program, even from Basic's command mode! The subroutine has been set up so that the values of several internal variables can be specified prior to each call to control Snapshot's operations (see Table 1).

The first (and most important) variable is ZZ!(0). This is a switch which controls Snapshot's operation. If it is set to −1, then Snapshot will be active when it is called. If it is set to 0 (or any value other than −1), then Snapshot is deactivated and will do nothing. This feature allows you to place several calls to Snapshot throughout your Basic program while it is being developed. Then, by simply setting ZZ!(0) to −1 at the beginning of the program, they all become active. When debugging is complete, setting ZZ!(0) to 0 will

"Alternatively, you can place a Stop statement at some strategic point in the ailing program. . ."

shut off all the dumping. This, of course, allows you to leave the program calls in place even when they are not needed.

The next important variable is ZZ$(1). This can be used to specify which variables you wish to dump. For example, ZZ$(1) = "A SU DE V" will cause only the variables listed in the string to be dumped. If you need to list a lot of variables, simply continue in ZZ$(3). If that is still not enough, then continue in ZZ$(5). Each of these three strings can be up to 254 characters in length.

As an alternative to listing the variables you want dumped, you may specify those you do not wish dumped. In this case, all names not specified are dumped. ZZ$(2) is used to specify which variables you do not wish to see. For example, ZZ$(2) = "G" will cause all variables except G to be dumped. If you need more space than that, continue first in ZZ$(4) and then in ZZ$(6). Each of these three strings can be a maximum of 252 characters long.

Note that for ZZ$(2), ZZ$(4) or ZZ$(6) to be recognized by Snapshot, ZZ$(1) must be null, that is, ZZ$(1) = " ". If ZZ$(1), ZZ$(2), ZZ$(4) and ZZ$(6) are all null, then all variables are dumped. Note also, that if any arrays are to be dumped, either specifically or by default, then the statement DIM ZZ!(30) must be placed somewhere in the calling program where it will be executed only once before Snapshot is called.

Another useful variable is ZZ$(0). This is the marker string. Whenever Snapshot is called, it prints the contents of ZZ$(0) on your listing before dumping any variables. For example, this string can be used to indicate on the dump the statement number of the Basic statement calling Snapshot. For example: ZZ$(0) = "This dump called from statement number 1010".

In addition, if the TRSDOS Date command has been executed to set a date of 1980 or later, then the date and time are also included in the marker string.

None of the values of these arguments are modified by Snapshot during execution. So when you specify these values in your program, they remain unchanged for subsequent calls to Snapshot.

A typical program under development or Debug is shown in Example 1. The code for the program is represented by the ellipsis (...). Several calls to SNAPSHOT are shown. (Snapshot is presumed to be installed beginning at line 61000.) For purposes of discussion, we'll assume that this is straight line code, that is, the statements shown are executed in top-to-bottom order with no branching.

Since we expect to dump one or more arrays, line 230 contains the required dimension statement. In line 1400, the master Snapshot switch, ZZ!(0), is turned on. We then set the marker string, ZZ$(0), and call Snapshot. Since we did not specify otherwise, all existing variables are dumped. Line 2300 respecifies the marker string and then calls Snapshot again. Note that the master switch is still on.

The program under development now has defined a great many variables so we decide to limit what will be dumped. In line 4500, we specify that we want to see only A, B, FG, HR and S. Line 4900 attempts to specify that the only variables we do not wish dumped are DE, R, and FG. Since ZZ$(1) is still set from line 4500, the result is a dump of the variables that line 4500 dumped. Of course, if their values have changed, we'll see the new values.

ZZ$(1) is nulled in line 5700. This results in the dumping of all variables except DE, R and FG since ZZ$(2) is still set from line 4900. Line 6800 dumps only the variable FG but the marker string from line 5700 is used since it was not respecified.

Another way to use Snapshot is to leave out calling statements in the program. When the program gets into trouble, simply hit Break and then type ZZ!(0) = −1: GOSUB 61000 from command mode. All your variables will be dumped. Alternatively, you can place a Stop statement at some strategic point in the ailing program and when it stops, enter the above line from command mode.

Description of Variable Structure

Within the computer, the storage of the Basic program variables immediately follows where the Basic program code is stored. Any variable must belong to one and only one of two classes: scalars and arrays. Scalars are variables which can have only one value at any given time. Arrays hold several values simultaneously. These values are addressed by means of indexing into the arrays.

Please note that although a variable may be coded in a Basic program residing in memory, space for that variable in either the scalar area or the array area is not allocated until the program is executed and the variable is either assigned a value or referenced.

Just as there are two distinct classes of variables, there are also two distinct

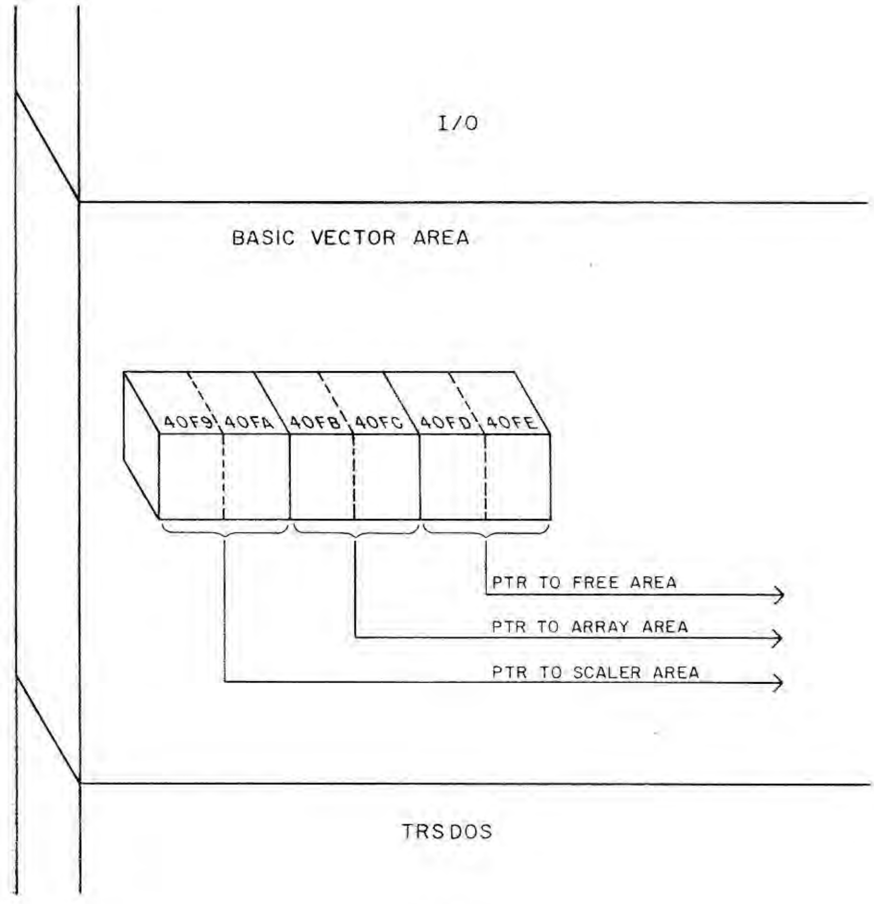


Fig. 2

"The Basic program code is followed by the scalar storage area."

storage areas in machine memory for these variables. The Basic program code is followed by the scalar storage area. This, in turn, is followed by the array storage area. Fig. 1 shows a bird's eye view of your TRS-80's software storage configuration. The top of the figure represents low memory. As you work your way down the diagram, you will be progressing through memory until you finally reach high memory at the bottom.

Where Are All the Variables?

As we've seen, the Basic program, scalars and arrays are all stored immediately adjacent to one another. Since the size of the Basic program can change from program to program and even during the development of a single program, then the actual locations where the variables are stored will also vary. Exactly where are the variables stored?

There are several areas of fixed storage which contain pointers to the areas where the variables are kept. Follow the pictorial presented in Fig. 2. Locations 16633 (40F9H) and 16634 (40FAH) contain a pointer to the first byte of the scalar storage area. The first byte of the array area is pointed to by locations 16635 (40FBH) and 16636 (40FCH). Note that since the array area follows the scalar area, the start of the array area is one byte beyond the end of the scalar area. Locations 16637 (40FDH) and 16638 (40FEH) point one byte beyond the end of the array area; that is, the first byte of the free storage area. The free storage area is that portion of memory after the end of the array area and the top of memory as defined by your response to Basic's Memory Size query.

Well, now that we've located the variables in memory, let's examine just how they are represented in storage. We will first examine the scalars and then the arrays.

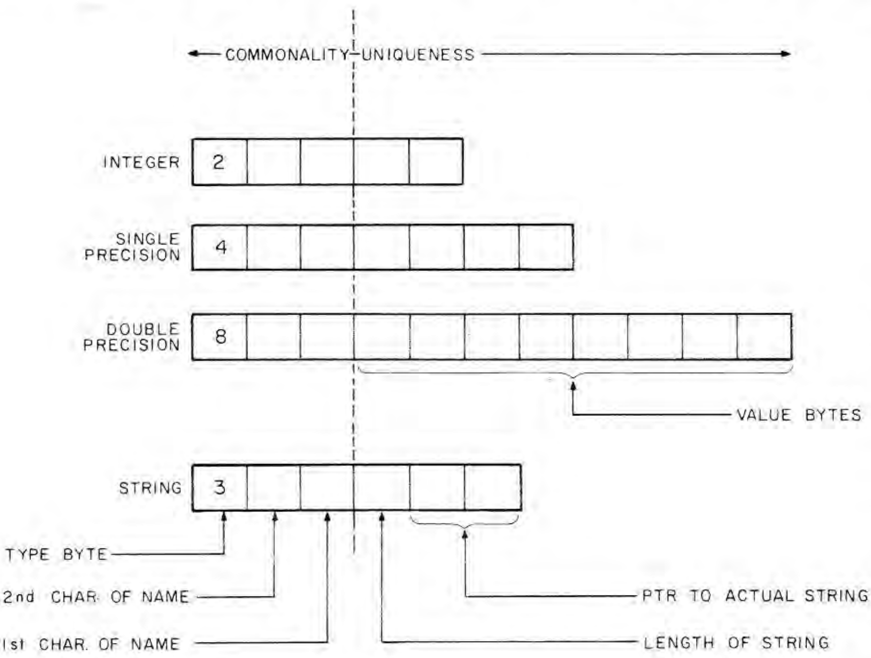


Fig. 3

```
 100 REM PROGRAM UNDER DEVELOPMENT OR DEBUG.
 230 DIM ZZ!(30)
...
1400 ZZ!(0) = - 1: ZZ$(0) = "STMT 1400": GOSUB 61000
...
2300  ZZ$(0) = "STMT 2300": GOSUB 61000
...
4500 ZZ$(0) = "STMT 4500": ZZ$(1) = "A B FG HR S": GOSUB 61000
...
4900 ZZ$(0) = "STMT 4900": ZZ$(2) = "DE R FG": GOSUB 61000
...
5700 ZZ$(0) = "STMT 5700": ZZ$(1) = "": GOSUB 61000
..
6800 ZZ$(1) = "FG": GOSUB 61000
```

Example 1

Scalar Representation

There are four types of variables: integer, single precision, double precision and string. Each variable is stored in consecutive bytes of memory and variables of a given class are stored adjacent to one another. The first three types are stored in a format similar to one another. The strings are stored differently from the numeric types.

Fig. 3 shows the byte-by-byte layout of the four different variable types. The first three bytes of all four types are identical in format. The first byte contains a binary value representing the type of the variable as follows:

Value	*Type*
2	Integer
3	String
4	Single Precision
8	Double Precision

At first glance, the values seemed to have been arbitrarily assigned to represent the type of the variable. Yet, if three is added to the values, the resultant number is the actual number of bytes of storage the particular type of variable occupies in the scalar area. It is important to note that this is the space required in the scalar area. As we shall see later, the string type also occupies other space.

The second byte of the format contains the ASCII value of the second character of the variable name. The third byte contains the ASCII value of the first character of the name. If the variable name is only one character in length, then the second byte of the format will contain a binary zero (00H). The third byte will always contain a true ASCII character. This is where commonality among the four types of variables ends.

The remaining bytes of the format of each of the numeric types contain the value of the variable; that is, the value you stored there when you executed an assignment statement in your Basic program. The remaining three bytes of the string type variable contain a one byte binary value representing the length of the string, and a two byte pointer to some location in the free area of memory where the actual character string is stored.

Array Representation

The format of an array—a two dimen-

STOCK MARKET SOFTWARE

By H & H Trading Company for Mod. I, II, & III TRS-80®s

STOCK TRACKER™ times trades on individual stocks and options. Uses a technical volume analysis. Completely objective. Gives buy, sell signals.

MARKET TRACKER™ times tops & bottoms on the Dow Industrials; a composite of six technical indicators. Some judgement required. Gives buy, sell signals.

These two programs are accurate.

Hundreds of satisfied users.

ASK FOR THE EVIDENCE!

FOR MORE INFORMATION OR TO ORDER, CONTACT:

H & H TRADING COMPANY
POST OFFICE BOX 549
CLAYTON, CALIFORNIA 94517
Telephone 415/672-3233 ✓223

VISA & MASTERCARD

®Regd. T.M. of Radio Shack ©

TIME ...

METHUSELAH™ has time for you and your TRS·80®

The name Methuselah has always been associated with long life. The Methuselah computer clock board with its 24 hour clock and its perpetual calendar is indeed worthy of the title. Methuselah puts state of the art technology into action with the on board four year lithium battery back-up to keep it running even when your computer is off. That means no more fooling around with the software or hardware "clocks" that become Rip Van Winkles when the computer is reset or turned off.

Methuselah has many other timely features. Software patches keep the DOS and BASIC time and clock commands ticking. Only two screws mount Methuselah inside the expansion interface without soldering, clipping, cutting, or jumpering. There is even a spare I/O port and four different interrupts available for people who love to tinker.

SPECS: MM/DD/YY, HH:MM:SS and day of the week. Four year lithium battery back-up. Crystal controlled timing (adjustable 32.768 oscillator). Twenty four hour clock and perpetual calendar.

Send check or money order to:

NDM DESIGNS ✓135
6330 W. North Ave.
Wauwatosa, WI 53213
(414) 259-0120

M-S1 .... $97.00
(add $4.50 P and H)
S-S1 .... $17.50
(add $2.50 P and H)
(If you wish to use Methuselah and the RS-232 board (26-1145) together, order this option which includes a new cover and connector.)
WI residents add 4% sales tax.

METHUSELAH is a trade mark of NDM Designs

® Reg. TM Tandy Corp.

TRS·80®

DISCOUNT COMPUTERS

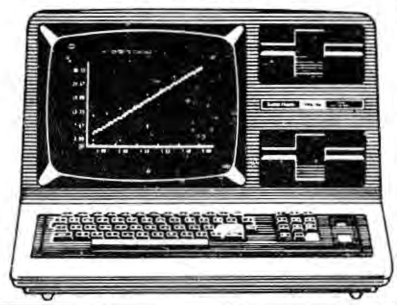
It is the stated goal of **CLEVELAND ELECTRONICS** to sell TRS-80 computers and peripherals at the lowest prices in the **WORLD.** We ask that you call other dealers, check your local Radio Shack store and mail order houses and then call us **LAST (Toll Free, of course).** We think that you will be pleasantly surprised. Please keep in mind that **CLEVELAND ELECTRONICS** is an authorized Radio Shack dealer and sells only pure Radio Shack products (you will find no inexpensive RAM memory, or non Radio Shack Disk Drives in our computers!) Being an **AUTHORIZED** Radio Shack store also means that you will be able to obtain warranty work at any of Radio Shack's more than 8000 stores. So... give us a call **TODAY!**

EPSON—WE NOW SELL THE ENTIRE EPSON LINE

FOR TOLL FREE ORDERING AND PRICE QUOTES, CALL:
1-800-331-3896

CLEVELAND ELECTRONICS
AUTHORIZED STORE # S-133
121 BROADWAY ✓191
CLEVELAND, OK. 74020
1-800-331-3896

TRS-80 MODEL III IN STOCK



". . . the values seemed to have been arbitrarily assigned."

sional, single precision type—is depicted in Fig. 4. The first three bytes of a variable in the array area are in a format identical to the first three bytes of the scalar format. Bytes four and five contain a two-byte binary value representing the remaining number of bytes in the variable. This value can be determined from the following formula: SIZE = 1 + (2*N) + (T*P). N represents the number of dimensions, P is the product of all dimensions and T is the variable type value.

Program Listing

```
1000 REM SUB: SNAPSHOT -- 06/01/80 14:28:00
1010 'Return If dump switch is off.
1020 IF NOT ZZ!(0) THEN RETURN
1030 'Convert address in range 0 to 65535 to range -32768 to 327
67.
1040 DEFFN ZZ%(ZZ!)=ZZ!+(ZZ!>32767)*65536
1050 'Evaluate a 2-byte pointer.
1060 DEFFN ZZ!(ZZ!)=PEEK(FNZZ%(ZZ!))+256*PEEK(FNZZ%(ZZ!+1))
1070 'Convert given positive number to a string without leading
blank.
1080 DEFFN ZZ$(ZZ%)=MID$(STR$(ZZ%),2)
1090 'Print marker string.
1100 LPRINT "===> SNAPSHOT";
1110 IF MID$(TIME$,7,1)="8" THEN LPRINT" (";TIME$;"): ";ZZ$(0) E
LSE LPRINT": ";ZZ$(0)
1120 'Reset number of variables already printed per line.
1130 ZZ!(8)=0
1140 'Specify number of values to be printed per line.
1150 ZZ!(9)=4
1160 '------ PROCESS SCALARS ------
1170 'Get address of first scalar variable.
1180 ZZ!(1)=FNZZ!(&H40F9)
1190 'Get the type of variable.
1200 ZZ!(2)=PEEK(FNZZ%(ZZ!(1)))
1210   GOSUB 1940 'Get variable name and validate it.
1220   'Ignore it if validation fails.
1230   IF NOT ZZ! THEN 1350
1240   'Ignore it if it is a function.
1250   IF ZZ!(4) THEN PRINT"Function ";ZZ$(7),: GOTO 1350
1260   'Set pointer to value bytes of variable.
1270   ZZ!(3)=ZZ!(1)+3
1280   GOSUB 2070 'CR if necessary.
1290   'Print variable name.
1300   LPRINT TAB((ZZ!(8))*21) ZZ$(7);"=";
1310   GOSUB 2120 'Print value.
1320   'Indicate that another value has been printed on the line
.
1330   ZZ!(8)=ZZ!(8)+1
1340   'Increment pointer to next scalar variable.
1350   ZZ!(1)=ZZ!(1)+3+ZZ!(2)
1360   'Loop to do next scalar if any.
1370   IF ZZ!(1)<FNZZ!(&H40FB) THEN 1200
1380 '------ PROCESS ARRAYS ------
1390 'Get address of first array variable.
1400 ZZ!(1)=FNZZ!(&H40FB)
1410 'Specify number of variables to be printed per line.
1420 ZZ!(9)=3
1430 'Get the type of variable.
1440 ZZ!(2)=PEEK(FNZZ%(ZZ!(1)))
1450   GOSUB 1940 'Get variable name and validate it.
1460   'Ignore it if validation fails.
1470   IF NOT ZZ! THEN 1850
1480   'Set pointer to first value byte of first element.
1490   ZZ!(3)=ZZ!(1)+5+2*PEEK(FNZZ%(ZZ!(1)+5))+1
1500   'For each potential dimension, reset the index counter.
1510   FOR ZZ%=0 TO 9: ZZ!(10+ZZ%)=0: NEXT
1520   'Carriage return if necessary to begin each array on a ne
w line.
1530   IF ZZ!(8)<>0 THEN LPRINT: ZZ!(8)=0
1540   'For each actual array dimension...
1550   FOR ZZ%=0 TO PEEK(FNZZ%(ZZ!(1)+5))-1
1560   '...set the size of the dimension.
1570     ZZ!(20+PEEK(FNZZ%(ZZ!(1)+5))-ZZ%)=FNZZ!(6+ZZ!(1)+ZZ%*2)

1580     NEXT
1590   'Determine the location of the last array value.
1600   ZZ!(10)=ZZ!(3)+FNZZ!(ZZ!(1)+3)-2*PEEK(FNZZ%(ZZ!(1)+5))-2
1610   GOSUB 2070 'CR if necessary.
1620     'Print the variable's name.
1630     LPRINT TAB(ZZ!(8)*30) ZZ$(7);"(";
```

Program continues

NEW PRINTERS. NEW PERIPHERALS. SAME OLD RELIABLE QUALITY AND VALUE.

1982 will find more OEM's, businesses, dealers and personal computer users turning to MICROTEK than ever before.

TekWriter-1

80 Column Dot Matrix Printer (Formerly BYTEWRITER-1)

The Tekwriter-1 printer is, dollar for dollar, the finest value in the industry. And we've proved it by comparing the Tekwriter-1 to the Epson MX-80. Our print speed is 14 lines per minute faster, our life expectancy is longer, the character sets are the same, and the interface, warranty and printhead replacement cost are all identical.* But the biggest difference is the price. The Tekwriter-1 is about $300 less.

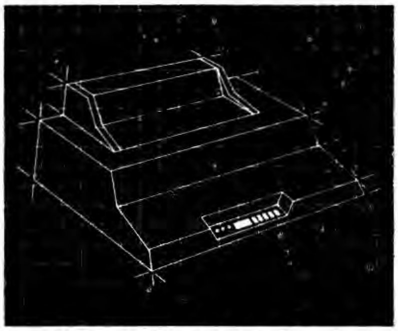
Our extensive testing has proved that the Tekwriter-1 interfaces problem-free to most parallel Centronics and serial (RS-232) computers.

The Tekwriter-1 is tough to beat for performance and quality.

*Data Source: Epson MX-80 Operation Manual

Parallel $349
Serial $389

TekWriter-2



NEW! 80/132 Column Dot Matrix Printer

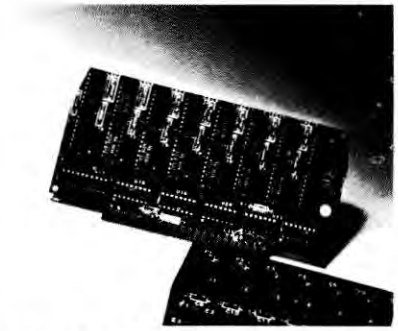
The Tekwriter-2 is perfectly suited to personal, business or OEM applications. Tekwriter-2 is designed to accept single sheet, roll or pin feed paper. It has a 9-wire dot matrix impact print head which produces crisp characters and has underlining capability. The printer is manufactured to run extremely quietly even while operating at peak output levels.

Tekwriter-2 is especially well suited to handle an abundance of text entry because of its data buffer expansion capability to 25K. This ability makes it an efficient graphics generator.

Parallel interface (Centronics type). Interfaces all models of TRS-80, Apple, and Atari 400/800, and most computers with Centronics printer interface.

$695

Peripherals



16K Apple Memory Board

Expands Apple II to 64K RAM Memory. Works with MICROSOFT Z-80 Softcard, Apple PASCAL and Visicalc software.

16K-32K Atari Memory Board

4116 RAM (200NS) Compatible with Atari 400/800

Parallel/Serial Data Buffer Converter

Interfaces with most computers and printers on the market today.
Switch selectable parallel or serial input/parallel or serial output.
Data buffer 2K standard — expandable to 62K.
Serial I/O BAUD rates switch independently selectable.

Atari Parallel or Serial Printer Cables

Pre-tested. Centronics or (RS-232) compatible.

APPLE PARALLEL INTERFACE CARD

Quantity and OEM discounts available.

Continuing our quest for excellence.

362

MICROTEK inc.

MICROTEK
9514 Chesapeake Drive
San Diego, CA 92123
(714) 278-0633
Outside CA call
Toll Free (800) 854-1081
TWX. 910-335-1269

"... two-byte addresses and 16-bit binary values (must be) stored least significant byte first..."

Program continued

```
1640     'For each dimension...
1650     FOR ZZ%=1 TO PEEK(FNZZ%(ZZ!(1)+5))
1660        '...increment next dimension if necessary,
1670        IF ZZ!(10+ZZ%)>ZZ!(20+ZZ%)-1 ZZ!(10+ZZ%+1)=ZZ!(10+ZZ%
+1)+1:ZZ!(10+ZZ%)=0
1680        '...print current value of current dimension,
1690        LPRINT FNZZ$(ZZ!(10+ZZ%));
1700        '...if not last dimension, print ",".
1710        IF ZZ%<>PEEK(FNZZ%(ZZ!(1)+5)) THEN LPRINT ",";
1720        NEXT
1730     'Print closing parenthesis.
1740     LPRINT ")=";
1750     'Increment first dimension.
1760     ZZ!(10+1)=ZZ!(10+1)+1
1770     GOSUB 2120 'Print the value.
1780     'Indicate that another value has been printed on the li
ne.
1790     ZZ!(8)=ZZ!(8)+1
1800     'Advance pointer to next array value.
1810     ZZ!(3)=ZZ!(3)+ZZ!(2)
1820     'Loop to do next value if any.
1830     IF ZZ!(3)<=ZZ!(10) THEN 1610
1840   'Increment pointer to next array variable.
1850   ZZ!(1)=ZZ!(1)+5+FNZZ!(ZZ!(1)+3)
1860   'Loop to do next array if any.
1870   IF ZZ!(1)<FNZZ!(&H40FD) THEN 1440
1880 'Issue final carriage return if required.
1890 IF ZZ!(8)<>0 THEN LPRINT
1900 RETURN
1910 '====== SNAPSHOT SUBROUTINES ======
1920 'SUB: Form variable's name.
1930 'Null the name string.
1940 ZZ$(7)=""
1950 'If second character of name does not exist, shift name str
ing right.
1960 IF PEEK(FNZZ%(ZZ!(1)+1))=0 THEN ZZ$(7)=" "
1970 'Get first character of name.
1980 ZZ$(7)=ZZ$(7)+CHR$(PEEK(FNZZ%(ZZ!(1)+2)))
1990 'Get character of name if it exists.
2000 ZZ$(7)=LEFT$(ZZ$(7)+CHR$(PEEK(FNZZ%(ZZ!(1)+1))),2)
2010 'Add variable type character to end of name.
2020 ZZ$(7)=ZZ$(7)+MID$(" %$!   #",ZZ!(2),1)
2030 GOSUB 2700 'Validate the name.
2040 RETURN
2050 '
2060 'SUB: Issue carriage return if necessary.
2070 IF ZZ!(2)=3 AND ZZ!(8)<>0 THEN LPRINT: ZZ!(8)=0
2080 IF ZZ!(2)<>3 THEN IF ZZ!(8)=ZZ!(9) THEN LPRINT: ZZ!(8)=0
2090 RETURN
2100 '
2110 'SUB: Get the value of the variable and print it.
2120 ON INSTR("2483",FNZZ$(ZZ!(2))) GOTO 2150,2230,2310,2390
2130 'Handle INTEGER VALUES.
2140 'Establish integer staging area.
2150 ZZ%=0
2160 'Set pointer to staging area.
2170 ZZ!(4)=VARPTR(ZZ%)
2180 GOSUB 2570 'Move value to staging area.
2190 'Print the value.
2200 LPRINT ZZ%;: RETURN
2210 'Handle SINGLE PRECISION VALUES.
2220 'Establish single precision staging area.
2230 ZZ!=0
2240 'Set pointer to staging area.
2250 ZZ!(4)=VARPTR(ZZ!)
2260 GOSUB 2570 'Move value to staging area.
2270 'Print the value.
2280 LPRINT ZZ!;: RETURN
2290 'Handle DOUBLE PRECISION VALUES.
2300 'Establish double precision staging area.
2310 ZZ#=0
2320 'Set pointer to staging area.
2330 ZZ!(4)=VARPTR(ZZ#)
2340 GOSUB 2570 'Move value to staging area.
2350 'Print value.
2360 LPRINT ZZ#;: RETURN
2370 'Handle STRING VALUES.
2380 'Establish string staging area.
2390 ZZ$=""
2400 'Set pointer to staging area.
2410 ZZ!(4)=VARPTR(ZZ$)
2420 GOSUB 2570 'Move value to staging area.
2430 'Reset current number of values on the line.
```

Program continues

The sixth byte is a binary value representing the number of dimensions in the array. Following the sixth byte there are as many pairs of bytes as indicated by the value of the sixth byte. Each pair has a two-byte binary value which is the number of array values in each of the dimensions. Beyond these pairs of bytes are the bytes containing the actual array values you assigned in Basic.

Each array value occupies the number of bytes indicated by the type number. The array values are stored in memory in order of the first dimension varying fastest. For example, if DIM A(2,2,2) were specified in a Basic program, the values of the array from low to high memory would be: A(0,0,0) A(1,0,0) A(2,0,0) A(0,1,0) A(1,1,0) A(2,1,0) A(0,2,0) A(1,2,0) A(2,2,0) A(0,0,1) A(1,0,1) A(2,0,1)... A(1,2,2) A(2,2,2).

Since the format of an array is rather involved, I will present an example. Suppose that we executed the statement DIM A% (4,5,6) from Basic. Remember that this is really an array 5 by 6 by 7 elements for a total of 210 elements or values. Example 2 shows the order and the values of the first few bytes of A%( ) as they would appear in memory. The byte number is shown in column one. This is followed by the hex and decimal values of the byte(s) in columns two and three. The last column shows what the byte represents.

Program Details

In Program Listing 1, line 1020 causes an immediate return to the calling program if the master dump switch, ZZ!(0), is not on. Otherwise, Snapshot begins execution. Next, three functions required by Snapshot are defined. Function ZZ% converts a single precision value in the range 0 to 65535 to an integer value in the range −32768 to 32767. This is required for the PEEK function and POKE statement. (See your Level II manual, page 8/5.) Function ZZ! evaluates a two-byte pointer. Since a peculiarity of the Z80 requires two-byte addresses and 16-bit binary values to be stored least significant byte first, this

Variable	Use
ZZ!(0)	Master Snapshot on/off switch. −1 = on 0 = off
ZZ$(0)	Marker string.
ZZ$(1)	Primary "dump" variable list.
ZZ$(3)	Secondary "dump" variable list.
ZZ$(5)	Tertiary "dump" variable list.
ZZ$(2)	Primary "no-dump" variable list.
ZZ$(4)	Secondary "no-dump" variable list.
ZZ$(6)	Tertiary "no-dump" variable list.

Table 1.

"My printer is capable of printing more than 80 columns of information."

Program continued

```
2440 ZZ!(8)=Ø
2450 'Return if string is null.
2460 IF ZZ$="" THEN LPRINT: GOTO 253Ø
2470 'If first character of string is a control code,
2480 'print first character only and return.
2490 IF ASC(ZZ$)<32 THEN LPRINT"CHR$(";FNZZ$(ASC(ZZ$));") Len";
                    PEEK(FNZZ%(ZZ!(3))): GOTO 253Ø
2500 'Print the character string and return.
2510 LPRINT CHR$(34);ZZ$;CHR$(34);" Len";PEEK(FNZZ%(ZZ!(3))): GO
TO 253Ø
2520 'Reset number of variables printed on the line.
2530 ZZ!(8)=-1: RETURN
2540 '
2550 'SUB: Move value of variable to staging area for printing.
2560 'Initialize byte counter.
2570 ZZ!(5)=Ø
2580 'Get byte from its resident location.
2590 ZZ!(6)=PEEK(FNZZ%(ZZ!(3)+ZZ!(5)))
2600    'Put it in the staging area.
2610    POKE FNZZ%(ZZ!(4)+ZZ!(5)),ZZ!(6)
2620    'Increment byte counter.
2630    ZZ!(5)=ZZ!(5)+1
2640    'Loop if not all bytes moved.
2650    IF ZZ!(5)<=ZZ!(2)-1 THEN 259Ø
2660 RETURN
2670 '
2680 'SUB: Variable Name Validation.
2690 'Clear "function" flag.
2700 ZZ!(4)=Ø
2710 'Check if name is a function instead of a variable.
2720 FOR ZZ%=1 TO 2
2730    ZZ!=ASC(MID$(ZZ$(7),ZZ%,1))
2740    IF ZZ!>192 THEN MID$(ZZ$(7),ZZ%,1)=CHR$(ZZ!-128): ZZ!(4)=
-1
2750    NEXT
2760 'Check if name is to be dumped.
2770 IF ZZ$(1)="" THEN 282Ø
2780 FOR ZZ%=1 TO 5 STEP 2
2790    IF INSTR(" "+ZZ$(ZZ%),LEFT$(ZZ$(7),2))<>Ø THEN ZZ!=-1: RE
TURN ELSE ZZ!=Ø
2800    NEXT: RETURN
2810 'Check if name is not to be dumped.
2820 FOR ZZ%=2 TO 6 STEP 2
2830    IF INSTR("ZZ "+ZZ$(ZZ%),LEFT$(ZZ$(7),2))<>Ø THEN ZZ!=Ø: R
ETURN ELSE ZZ!=-1
2840    NEXT:RETURN
```

function takes that into account. The third function, ZZ$, accepts a positive number and returns that number as a string but without the usual leading blank. It is used primarily for formatting array indices for the dump.

In lines 1100–1110, the marker string is printed. Lines 1130 and 1150 reset the counter which keeps track of the number of variables already printed on a line and the maximum allowed on a line, respectively. My printer is capable of printing more than 80 columns of information. You may have to respecify ZZ!(9) smaller if you have a narrow-format printer.

Lines 1180–1370 process the scalar variables. Lines 1400–1870 process the arrays. The remaining lines are dedicated to the various subroutines used by Snapshot. I will describe first the scalar processing, then the array processing and lastly, the subroutines.

Scalar Processing

Line 1180 sets the address of the start of the scalar area as discussed earlier. Line 1200 determines the type of the variable and is also the start of the scalar loop. The name of the variable is determined in line 1210. This line also causes validation of that name. Name validation will be addressed later during the subroutine discussion. If the name is not a valid name to be dumped, then line 1230 branches to avoid any further processing of the name. In the event the program discovers that it has stumbled upon a function name instead of a variable name, line 1250 indicates on the video screen that a function has been found and then branches to skip processing that name.

In line 1270, the pointer is set to the location of the actual value bytes of the numeric type variables. In the case of the string variables, the pointer is set to the location of the length byte of the string. Lines 1280–1330 handle the necessary bookkeeping for the dump formatting. The number calculated for the tab value in line 1300 must evaluate to a value between 0 and 255 according to the Level II manual, page 3/2. This statement allots 21 columns for the variable's name and value. You can change it to suit your needs as required. Note that variable counter ZZ!(8) will take on values ranging from 0 to one less than ZZ!(9).

When line 1350 is reached, processing for a particular variable has been completed. This line then increments the pointer to the next variable. Line 1370 checks that the pointer still points within the scalar storage area. If it does, a branch is taken to repeat the scalar loop. Otherwise, array processing begins.

Array Processing

The structure of arrays is similar to that for scalars except that once an array is located, its multiplicity of values must be dumped before proceeding to the next array. Lines 1440–1870 encompass the major array loop. Each execution of this loop will process one complete array. Lines 1610–1830 constitute the minor array loop.

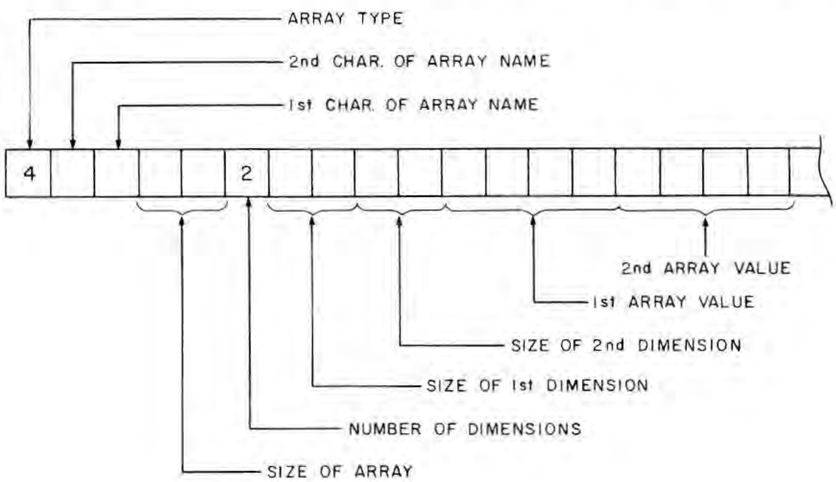


Fig. 4

"... Snapshot can handle arrays in up to ten dimensions."

Each circuit of this loop will process a single element of the array and is repeated once for each of the array's values. The following description will address those details of array processing where the code differs from that for scalar processing.

Two sets of array indices are stored in the array ZZ!( ). Beginning at ZZ!(11) and continuing to ZZ!(20), the values of the indices for the current array element being processed are stored. Note that Snapshot can handle arrays in up to ten dimensions. These ten values are cleared in line 1510. The maximum values which the dimensions may attain for the current array being processed are stored beginning at ZZ!(21) and continuing to ZZ!(30). These values are determined in the For...Next loop in lines 1550–1580.

Line 1600 sets the limit for the minor loop to ensure that we don't attempt to dump more values than an array has. This limit is checked in line 1830 which will terminate the loop when the limit is reached.

The For...Next loop in lines 1650–1720 handles the incrementing and printing of array indices. The fastest-varying index is incremented in line 1760. Line 1810 moves the pointer to the next array element. Line 1850 increments another pointer to the next array when the minor loop terminates. Line 1870 will execute the major loop again if there are other arrays to be processed. Otherwise, line 1890 issues a final carriage return if necessary to achieve a neat printout.

The Subroutines

Snapshot uses five subroutines to aid its processing. The subroutine at lines 2070–2090 simply issues a carriage return when the specified number of values have been printed on a line. Each string value is printed on a new line.

The name validation subroutine, lines 2700–2840, determines whether the variable's name is correct and whether it is to be dumped. User-defined functions are also stored in the scalar variable area. The fact that a name belongs to a function is indicated by the fact that a value of 128 has been added to the ASCII value of its first character. Lines 2720–2750 examine the name to determine if it is a function name and if so, set the function flag ZZ!(4) after reducing the ASCII value by 128 to make it printable. Line 2770 checks whether specific variables have been determined and are to be dumped. If so, then the For...Next loop in lines 2780–2800 determines whether the current value is listed, sets the validation flag appropriately and then immediately returns. If not, a branch is taken to the For...Next loop of lines 2820–2840 where a determination is made regarding whether or not any names have been specified not to be dumped. Note that line 2830 appends "ZZ " to any possible user specification thus preventing its own names from being dumped. If a name is determined to be valid, then flag ZZ! is set to −1. Otherwise, the flag is set to 0. After the flag is set, the validation subroutine returns to the caller.

The character string, ZZ$(7), containing the name of the variable or function is formed by the subroutine residing at lines 1940–2040. The first two characters of ZZ$(7) contain the name. If the name is a single character name, it is placed in the second position and the first is filled with a blank. The third character of ZZ$(7) contains the standard type indicator: %, !, #, or $. Before the subroutine returns, it calls the name validation subroutine thus causing flags ZZ! and ZZ!(4) to be set.

The subroutine at lines 2120–2530, while seemingly quite extensive, is really quite simple. Its purpose is to get the actual value of the variable and print it. The first line determines what type of variable needs to be printed and branches accordingly to one of four different sections of the subroutine where each of the four types is handled independently. Each of the four sections is virtually identical to the remaining three.

Processing of single precision variables begins at line 2230 where the program establishes a single precision staging area. Line 2250 sets a pointer to the location of

Variable	Use
ZZ!(0)	Master Dump control switch.
ZZ!(1)	Pointer to beginning of variable.
ZZ!(2)	Type of variable.
ZZ!(3)	Pointer to value bytes of variable.
ZZ!(4)	Pointer to staging area.
	Also, validation "function name" flag.
ZZ!(5)	Loop control for move to staging area.
ZZ!(6)	Value of byte being moved to staging area.
ZZ!(7)	Loop control for array index development.
ZZ!(8)	Current number of values per line.
ZZ!(9)	Maximum number of values per line.
ZZ!(10)	Location of last array value.
ZZ!(11)	Current value of first dimension.
ZZ!(12)	Current value of second dimension.
ZZ!(13)	Current value of third dimension.
ZZ!(14)	Current value of fourth dimension.
ZZ!(15)	Current value of fifth dimension.
ZZ!(16)	Current value of sixth dimension.
ZZ!(17)	Current value of seventh dimension.
ZZ!(18)	Current value of eighth dimension.
ZZ!(19)	Current value of ninth dimension.
ZZ!(20)	Current value of tenth dimension.
ZZ!(21)	Maximum value of first dimension.
ZZ!(22)	Maximum value of second dimension.
ZZ!(23)	Maximum value of third dimension.
ZZ!(24)	Maximum value of fourth dimension.
ZZ!(25)	Maximum value of fifth dimension.
ZZ!(26)	Maximum value of sixth dimension.
ZZ!(27)	Maximum value of seventh dimension.
ZZ!(28)	Maximum value of eighth dimension.
ZZ!(29)	Maximum value of ninth dimension.
ZZ!(30)	Maximum value of tenth dimension.
ZZ%	Staging area for integer values.
	Also, validation loop control.
	Also, array dimension size loop control.
ZZ!	Staging area for single precision values.
	Also, validation "valid name" flag.
ZZ#	Staging area for double precision values.
ZZ$	Staging area for string values.
ZZ$(0)	Marker string.
ZZ$(1;3;5)	List of variables to Snapshot.
ZZ$(2;4;6)	List of variables not to Snapshot.
ZZ$(7)	Variable's name.

Table 2

Function	Use
ZZ%	FN to convert addresses to integer.
ZZ!	FN to evaluate pointers.
ZZ$	FN to remove leading blank from number.

Table 3

"Snapshot is really just like a camera. . ."

this area. Next, the program calls a subroutine to place the value bytes of the single precision variable in the staging area. (String variables are processed differently. The one-byte length value and the two-byte pointer to the string are moved.) Line 2280 accesses the value bytes by referring to the name of the staging area and prints the variable.

The last remaining subroutine resides at lines 2570–2660. All this subroutine does is copy, one byte at a time, the value bytes of a variable from their normally resident location to the staging area. Tables 2 and 3 list all the Snapshot variables and functions, respectively.

As you can see, although Snapshot looks rather involved, it isn't. The program was not written with any thoughts of speed or efficiency in mind. Look at line 1760; ZZ!(10 + 1). Certainly, ZZ!(11) is better, but was written that way to help user comprehension.

Snapshot is really just like a camera, only instead of freezing visual images for later scrutiny, it freezes the varying values of your program's variables for your convenient analysis. ■

Byte	Hex Value	Dec Value	Representing
1	02	2	Type indicating integer
2	00	0	No second letter of name
3	41	65	ASCII "A"
4, 5	AB, 01	427	Size of array
6	03	3	Number of dimensions
7, 8	00, 05	5	Size of first dimension
9,10	00, 06	6	Size of second dimension
11,12	00, 07	7	Size of third dimension
13,14	?	?	Value of first element

Example 2

Conversion to Level II Basic

Snapshot uses only two features of Disk Basic. In one instance, in line 1110, TIME$ is used to print the current date and time. To use Snapshot from Level II Basic, simply replace this line with the following:

```
1110 LPRINT": ";ZZ$(0)
```

In the second instance, MID$ is used on the left of the equal sign in line 2740. It is used in the For. . .Next loop which checks for user defined function names. Since Level II doesn't allow such functions, this loop can be completely eliminated. Delete lines 2720–2750. ■

✓See List of Advertisers on page 290

GET YOUR TRS-80 DIRECT

FROM

AUTHORIZED SALES CENTER

FORT WORTH COMPUTERS AND VIDEO

377 PLAZA, HWY 377 GRANBURY
FORT WORTH, TX 76048 ✓462

AND SAVE $

FREE! Zenith Color T.V. NEW 1982 Model, with every MOD II or equivalent.

Get our quote "before" you buy.
Full manufacturers' warantee.
Texas residents add 4% sales tax.

817-573-4111

COST OF CALL REFUNDED FOR ORDERS BY TELEPHONE

MODELS I & III DISK BASIC PROGRAMS

RANDOM ACCESS PAYROLL VER. 1.1

- No complicated initialization
- EDIT & LIST
- NO SPECIAL CHECKS
- USES NEB 9020 checks
- PAY any employee anytime
- SALARIED or hourly
- SPECIAL PAY—special hours
- TWO SAVINGS—including RIA
- CLASSED by occupation or dept
- PAYSTUB shows Year-to-Date
- STATE—FICA—FED. TAXES—SAVINGS

SEND YOUR STATE TAX SCHEDULE—FREE—customized to your state tax

Documentation $10.00
Disk & Documentation $95.00

RANDOM ACCESS DEPRECIATION

- LISTS any one year's depreciation
- OR complete list of all property
- PRINTOUT shows—I.D. Number—description—
- Yr. purchased—life—method—
  1st yr. additional depreciation—
- Reg. deprec.—deprec. prior yrs.—balance
- SUMMARY—total value prop.—additional 1st yr
- REGULAR deprec.—deprec. prior yrs.
- PERMANENT records for your taxes

Documentation $15.00
Disk & Documentation $195.00

MIN: 32K—1 DISK—PRINTER/132 CPI
TERMS: Personal checks require 3 weeks to clear.
USE—VISA—MASTER CHARGE—MONEY ORDERS
TEL 7 PM-9:30 PM EASTERN—(617) 359-2364/6370
SEND: SASE for additional information.

MEDFIELD COMPUTER SOFTWARE
39 GREEN ST., MEDFIELD, MA 02052 ✓421

TRS-80 16K Level II

LEARN MATHEMATICS THE FUN WAY!

Programs that develop mathematical reasoning and estimation skill
Beyond drill and practice!!
All ages will enjoy these games that teach

The Estimation Game (Animated!) Cassette · $12.95

The Distance Game Cassette $12.95

We take Visa or Master Charge (include card# and expiration date), checks, money orders.

EDUCATIONAL PROGRAMS
Programs that Teach

P.O. 2345
West Lafayette, IN 47906
(317)463-4778 ✓303

TOOLS

MODEL III CASSETTE COPY

- Copy and verify cassette programs
- Split large programs into sections
- Convert between 500/1500 baud
- Produce multiple cassettes efficiently
- Copy programs larger than avail. memory
- Rename programs

$47.75

MODEL II SCREEN EDITOR

- Edits all ASCII files
- Scrolling
- Find/Change
- Block Move/Delete
- Improved listings
- Full Screen Insert/Delete
- Split/combine lines
- On-screen HELP
- Auto line numbers

WRITTEN IN BASIC — FULL SOURCE PROVIDED

Requires 64K Mod II
Specify DOS 1.3 or 2.0

$97.95

CLARK CONSULTANTS

363 Lafayette Street, Salem, MA 01970
Massachusetts residents add 5% sales tax ✓431

ALL NEW!

COMPACT 232-IS INTERCONNECT SET

- Easy, full configuration of line signals
- All 25 signals passed to headers
- Designed for wire-wrap or jumpers
- Additional header for break-in round cable
- Made for permanent installation

LOW COST - *only* $35.00

JIS - Jumper Set 6.50

SYZYGY™ ✓286
"the interface people"
256 W. San Bernardino Rd.
Covina, CA 91723 • (213) 332-3320

---

80 microcomputing

Back Issues

January 1980	$5.00
February to June 1980 Single back issue	3.00
July 1980 on Single back issue	3.50

Add $1.00 per magazine for shipping.

10 or more back issues
add $7.50 per order for shipping.

MICRO-DESIGN PRESENTS THE MDX-4

PHONE MODEM

P.C. BOARD & USER MANUAL

$39.95

- ★ 300 BAUD
- ★ DIRECT CONNECT
- ★ ANSWER AND ORIGINATE MODES
- ★ SOLDER-MASKED & SILK-SCREENED
- ★ ON BOARD POWER SUPPLY

MDX-4 PC BOARD & USERS MANUAL	39.95
USERS MANUAL	8.95

(Texas Residents Add 5% Tax)
(Add $3.00 Shipping for Board)

CALL OR WRITE ✓379

MD
MICRO DESIGN
P.O. BOX 748
MANCHACA, TX 78652
(512) 282-0225

DEALER INQUIRIES INVITED

TAX PREPARERS!

PROFESSIONAL FEDERAL INCOME TAX PROGRAMS

- For TRS-80 Model I or Model III. Requires 2 Disk Drives and 32K
- Determines Least Tax Quickly
- Essential Information Relayed between Forms Automatically—Avoids Duplicating Entries
- Uses IRS prescribed Computer Generated Print-outs
- Prohibits Bypassing of Mandatory Entries
- Accuracy Assured by *Triple Check Logic*
- Enter only Pertinent Lines—Much Faster than Line by Line Entry
- Single Line Correction with Automatic Update of Succeeding Totals

Basic Package: 1040, 1040A, Schedules A, B
All for $100.00   Additional Schedules $25.00 each

Single Schedules may be purchased separately.
Send $2.50 for Full Description and Samples (applied to purchase price)

MICRO-TAX
P.O. Box 4262, Mountain View, CA 94040
Call: (415) 964-2843 ✓486

BARGAINBYTEBARGAINBYTEBARGAINBYTEBARGAIN

BARGAIN PRICE PROGRAMS FOR TRS-80*!

On tape for Model I and Model III 16K LII.
Instructions included

- For the whole family - Solitaire Card Games: Four (excellent) for only $15.
- How Internal Combustion Engines Work! 2-cycle $7, 4-cycle $7, Both $13.
- Get the feel of COBOL! Sampler for beginners (32K Disk required) $9. With book $18.

Satisfaction Guaranteed
Not Available Anywhere Else!

Send check or money order to:
BARGAINBYTE,™ Box 23195, Harahan, LA 70183

* Tandy Corp. Trademark

✓See List of Advertisers on page 290

Wayne Green Books

NEW ARRIVALS

Annotated BASIC—A New Technique for Neophytes.

BASIC programming was supposed to be simple—a beginner's programming language which was so near to English that it could be easily understood. But, in recent years, BASIC has become much more powerful and therefore much more difficult to read and understand. BASIC simply isn't basic anymore.

Annotated BASIC explains the complexities of modern BASIC. It includes complete TRS-80* Level II BASIC programs that you can use. Each program is annotated to explain in step-by-step fashion the workings of the program. Programs are flowcharted to assist you in following the operational sequence. And—each chapter includes a description of the new concepts which have been introduced.

Annotated BASIC deals with the hows and whys of TRS-80 BASIC programming. *How* is a program put together? *Why* is it written that way? By observing the programs and following the annotation, you can develop new techniques to use in your own programs—or modify commercial programs for your specific use.

Annotated BASIC Volume 1 contains Projecting Profits, Surveyor, Things to Do, Tax Shelter, Introduction to Digital Logic, Camelot, The Soundex Code, Deduction, Op Amp, Contractor Cost Estimating.
BK7384 $10.95 ISBN 0-88006-028-X

Annotated BASIC Volume 2 contains Rough Lumber List, Trip Mileage, Flight Plan, OSCAR Data, SWR/Antenna Design, Supermaze, Petals Around the Rose, Numeric Analysis, Demons, Air Raid, Geography Test, Plumbing System Design. (available March) **BK7385 $10.95** ISBN 0-88006-037-9

Order Both Volumes and Save! **BK738402 $18.95**

Kilobaud Klassroom—
A practical course in digital electronics
by George Young and Peter Stark

Learning electronics theory without practice isn't easy. And it's no fun to build an electronics project that you can't use. *Kilobaud Klassroom*, the popular series first published in *Kilobaud Microcomputing*, combines theory with practice. This is a *practical* course in digital electronics. It starts out with very simple electronics projects, and by the end of the course, you'll construct your own working microcomputer!
Authors Young and Stark are experienced teachers, and their approach is simple and direct. Whether you're learning at home or in the classroom, this book provides you with a solid background in electronics—and you'll own a computer that you built yourself!

Kilobaud Klassroom contains Getting the Ball Rolling, Gates and Flip-Flops Explained, J.K. Flip-Flops and Clocked Logic, PC Boards and Power Supplies, Hardware Logical Functions, Voltage, Current and Power Supplies, Transistors, Diodes and OP Amps, Pulses and More Pulses, Counters and Registers, Bus Traffic Control, ROM and RAM Memories, I/O Circuitry, Parallel and Serial I/O Ports, Computer I/O III, Computer I/O IV, Computer I/O V, Processor Connections, Finally..., The Kilobaud Krescendo, Eproms and Troubleshooting, Expansions and Programming, Machine-Language Programming, Assembly-Language Programming, Connecting to the Outside World.
ISBN 0-88006-027-1 (available March) **BK7386 $14.95**

The New Weather Satellite Handbook
By Dr. Ralph E. Taggart WB8DQT

Here is the completely updated and revised edition of the best-selling *Weather Satellite Handbook*—containing all the information on the most sophisticated spacecraft now in orbit. Dr. Taggart has written this book to serve both the experienced amateur satellite enthusiast and the newcomer. The book is an introduction to satellite watching that tells you how to construct a complete and highly effective ground station. Not just ideas, but solid hardware designs and all the instructions necessary to operate the equipment are included. An entire chapter is devoted to microcomputers and the Weather Satellite Station. And for the thousands of experimenters who are operating stations, *The New Weather Satellite Handbook* details all the procedures necessary to follow the current spacecraft.

Weather Satellite contains Operational Satellite Systems, Antenna Systems, Weather Satellite Receivers, A Cathode Ray Tube (CRT) Monitor for Satellite Picture Display, A Direct-Printing Facsimile System for Weather Satellite Display, How to Find the Satellite, Test Equipment, Microcomputers and the Weather Satellite Station, Station Operations.
ISBN 0-88006-015-8 available now! **BK7383 $8.95**

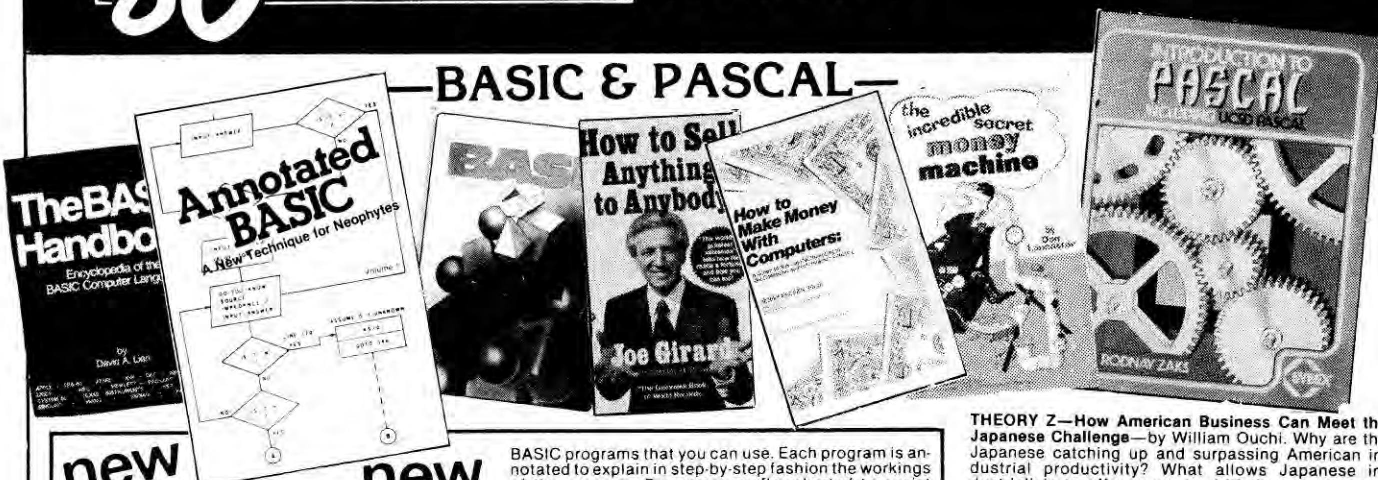
*TRS-80 is a trademark of Radio Shack Division of Tandy Corp.

FOR TOLL-FREE ORDERING CALL 1-800-258-5473 **WG02**
WAYNE GREEN BOOKS • PETERBOROUGH NH 03458

Use the order card or itemize your order on a separate piece of paper and mail to Wayne Green Books Att: Sales • Peterborough NH 03458.
Be sure to include check or detailed credit card information. (Visa, Master Charge or American Express accepted.)
No C.O.D. orders accepted. All orders add $1.50 for the first book, postage and handling; $1.00 each additional book; $10.00 per book foreign air mail.
Please allow 4-6 weeks after publication for delivery. Questions regarding your order? Please write to Customer Service at the above address.

80 microcomputing™ bookshelf

—BASIC & PASCAL—



new **new**

ANNOTATED BASIC—A NEW TECHNIQUE FOR NEOPHYTES.—BASIC programming was supposed to be simple—a beginner's programming language which was so near to English that is could be easily understood. But, in recent years, BASIC has become much more powerful and therefore much more difficult to read and understand. BASIC simply isn't basic anymore.

Annotated BASIC explains the complexities of modern BASIC. It includes complete TRS-80* Level II BASIC programs that you can use. Each program is annotated to explain in step-by-step fashion the workings of the program. Programs are flowcharted to assist you in following the operational sequence. And—each chapter includes a description of the new concepts which have been introduced.

Annotated BASIC deals with the hows and whys of TRS-80 BASIC programming. *How* is a program put together? *Why* is it written that way? By observing the programs and following the annotation, you can develop new techniques to use in your own programs—or modify commercial programs for your specific use.
Annotated BASIC Volume 1 BK7384 $10.95
Annotated BASIC Volume 2 BK7385 $10.95
Order Both Volumes and Save! BK738402 $18.95

INTRODUCTION TO TRS-80 LEVEL II BASIC AND COMPUTER PROGRAMMING—by Michael P. Zabinski. Written by an experienced educator, this is the book for those beginners who want to learn about computers without having to become an expert. It has practical programs, useful line-by-line comments, excellent flowcharts accompanied by line numbers and over 200 exercises which help the reader assess progress, reinforce comprehension, and provide valuable practical experience. BK1219 $10.95.*

50 BASIC EXERCISES—by J. P. Lamoitier. This book is structured around the idea that the best way to learn a language is through actual practice. It contains 50 completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. Program subjects include mathematics, business, games, and operations research, and are presented in varying levels of difficulty. This format enables anyone to learn BASIC rapidly, checking their progress at each step. BK1192 $12.95*.

THE BASIC HANDBOOK—SECOND EDITION—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. BK1174 $19.95.*

LEARNING LEVEL II—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC. It explores every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. BK1175 $15.95.*

BASIC BASIC (2ND EDITION)—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. BK1026 $10.50.*

ADVANCED BASIC—Applications including strings and files, coordinate geometry, area, sequences and series, simulation and graphing and games. BK1000 $10.75.*

INTRODUCTION TO PASCAL—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL—the most widely used dialect for small computers. No computer or programming experience is necessary. BK1189 $14.95.*

PROGRAMMING IN PASCAL—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language and therefore suitable for an introductory course. BK1140 $12.95.*

—GAMES—

40 COMPUTER GAMES FROM *KILOBAUD MICROCOMPUTING*—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. BK7381 $17.95.*

BASIC COMPUTER GAMES—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. BK1074 $7.50.*

MORE BASIC COMPUTER GAMES—Edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. BK1182 $7.50.*

WHAT TO DO AFTER YOU HIT RETURN—PCC's first book of computer games...48 different computer games you can play in BASIC...programs, descriptions and many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. BK1071 $14.95

THEORY Z—How American Business Can Meet the Japanese Challenge—by William Ouchi. Why are the Japanese catching up and surpassing American industrial productivity? What allows Japanese industrialists to offer guaranteed lifetime employment to their workforce? This book will help you understand the Theory Z managerial philosophy and its implications for the American corporate future. Examples are given of the American industrial giants already operating under Z-style management, and the impact of this style on the quality of their executives and workers is explored. A must for the alert businessman, large or small. BK1226 $12.95*

SO YOU ARE THINKING ABOUT A SMALL BUSINESS COMPUTER—by Richard G. Canning and Nancy C. Leeper. For a well-organized manual on the process of selecting the right computer system for your small business, this text can't be excelled. Designed to introduce the novice in data and word processing to the real benefits of computerization, the book is filled with money- and time-saving tips, photos of equipment, lists of suppliers, prices, explanations of computer terminology, and helpful references to additional sources of information. Everyone contemplating a first computer installation should have this book. BK1222 $14.00*

PAYROLL WITH COST ACCOUNTING—IN BASIC—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). BK1001 $20.00.*

SOME COMMON BASIC PROGRAMS— Published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. BK1053 $14.99 paperback.

PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. BK1009 $11.95.*

—MONEYMAKING—

HOW TO MAKE MONEY WITH COMPUTERS—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. BK1003 $15.00.*

HOW TO SELL ANYTHING TO ANYBODY—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. BK7306 $2.25.*

THE INCREDIBLE SECRET MONEY MACHINE—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. BK1178 $5.95.*

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

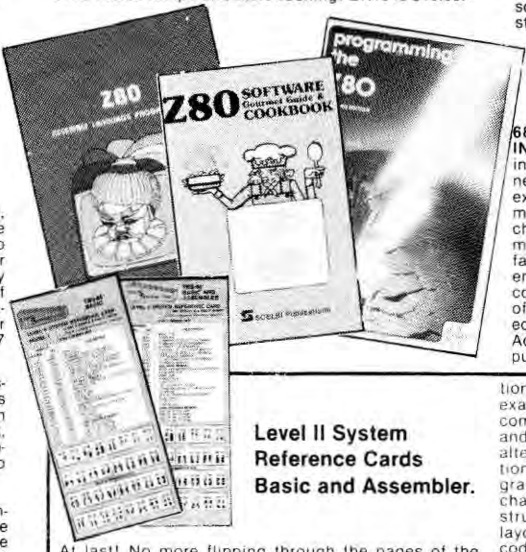
FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing bookshelf

***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.*

Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK—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! BK1045 $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.*



PROGRAMMING —Z-80—

TRS-80 ASSEMBLY LANGUAGE—by Hubert S. Howe, Jr. This book incorporates into a single volume all the pertinent facts and information you need to know to program and enjoy the TRS-80. Included are clear presentations of all introductory concepts, completely tested practical programs and subroutines, details of ROM and RAM and disk operating systems, plus comprehensive tables, charts and appendices. Suitable for the first time user or more experienced users. BK1217 $9.95.*

PROGRAMMING THE Z-80—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. BK1122 $14.95.*

INSIDE LEVEL II—For machine language programmers. This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. BK1183 $15.95.*

6809

6809 MICROCOMPUTER PROGRAMMING AND INTERFACING—by Andrew C. Staugaard, Jr. Getting involved with Tandy's new Color Computer? If so, this new book from the Blacksburg Group will allow you to exploit the awesome power of the machine's 6809 microprocessor. Detailed information on processor architecture, addressing modes, register operation, data movement, arithmetic logic operations, I/O and interfacing is provided, as well as a review section at the end of each chapter. Four appendices are included covering the 6809 instruction set, specification sheets of the 6809 family of processors, other 6800 series equipment and the 6809/6821 Peripheral Interface Adapter. This book is a must for the serious Color Computer owner. BK1215 $13.95.*

Level II System Reference Cards Basic and Assembler.

At last! No more flipping through the pages of the Basic manual! No more working through the maze of machine language instructions! These cards completely summarize the Basic and Assembler manuals! FEATURES INCLUDE:
memory map, eyeball graphics, math instructions, basic commands, store instructions, basic functions, load instructions, basic statements, move instructions, special keys, exchange instructions, print using examples, shift instructions, basic special characters, compare instructions, basic and assembler messages and codes, branch instructions, basic facts, data alteration instructions, reserved words, I/O instructions, ROM routines, complete character chart with graphics and space-compression codes, hex-dec chart, control code cross-reference, assembler instructions commands and operators, screen line layout, editor commands & subcommands, condition code easy access
Plus—"magic graphics number—a mystery until you learn how to use it!

Designed as a fold-up, accordion-style card, fits in your pocket. Panels organized for optimum speed for reference. Basic only FC1001 $2.95
Basic and Assembler FC1002 $4.95
ALSO AVAILABLE FOR MOD III FC1003 $5.95

ALL NEW—FOR 80 OWNERS ENCYCLOPEDIA FOR THE TRS-80*

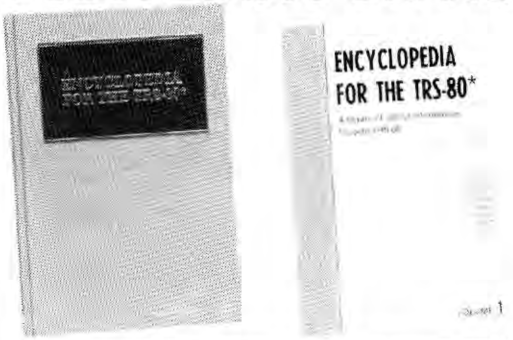
ENCYCLOPEDIA FOR THE TRS-80** is the newest peripheral for your microcomputer. This ten-volume reference series contains programs and articles especially selected to extend the documentation that comes with the TRS-80. Volumes of the **Encyclopedia** are being issued one-at-a-time, approximately six weeks apart. This means that each new volume will reflect the latest developments in TRS-80 microcomputing. **Encyclopedia for the TRS-80** is available in two editions. The deluxe COLLECTOR'S EDITION has a handsome green and black hard cover with a dust jacket. A soft cover edition is also available.

To order single hard cover volumes of the **Encyclopedia** please specify:

Volume 1	**EN8101**	**$19.95***
Volume 2	**EN8102**	**$19.95***
Volume 3 (avail. Nov.)	**EN8103**	**$19.95***
Volume 4 (avail. Jan.)	**EN8104**	**$19.95***
Volume 5 (avail. Feb.)	**EN8105**	**$19.95***

To order single softcover volumes of the **Encyclopedia** please specify:

Volume 1	**EN8081**	**$10.95***
Volume 2	**EN8082**	**$10.95***
Volume 3 (avail. Nov.)	**EN8083**	**$10.95***
Volume 4 (avail. Jan.)	**EN8084**	**$10.95***
Volume 5 (avail. Feb.)	**EN8085**	**$10.95***



SPECIAL INTRODUCTORY OFFER
Order the entire set (the first ten volumes) of the **Encyclopedia for the TRS-80** and receive the tenth volume **FREE!**

10 Volume deluxe hard cover COLLECTOR'S EDITION —a $199.50 value.
EN8100 $164.00 postpaid

10 Volume soft cover edition, a $109.50 value.
EN8080 $83.00 postpaid

ENCYCLOPEDIA LOADER is the software companion to the Encyclopedia for the TRS-80. This special series of cassettes will allow you to load selected programs from the Encyclopedia instantly. Hours of keyboarding time and the aggravating search for typos is eliminated by using the **Encyclopedia Loader.**

Encyclopedia Loader for Volume 1 of the Encyclopedia of the TRS-80 is available at a **Special Introductory** Price of $12.95 (a $14.95 value). To take advantage of this offer specify:
EL800111 $12.95*

Additional Encyclopedia Loaders will be available as each volume of the Encyclopedia for the TRS-80 is issued. To order specify **Encyclopedia Loader:**

Volume 2	**EN8002**	**$14.95***
Volume 3	**EN8003**	**$14.95***
Volume 4	**EN8004**	**$14.95***
Volume 5	**EN8005**	**$14.95***

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing bookshelf

—INTRODUCTORY—

HOBBY COMPUTERS ARE HERE! If you want to come up to speed on how computers work—hardware and software—this is an excellent book. It starts with fundamentals and explains the circuits and the basics of programming, along with a couple of TVT construction projects, ASCII, Baudot, etc. This book has the highest recommendations as a teaching aid. $4.95.* BK7322

THE NEW HOBBY COMPUTERS—This book takes it from where Hobby Computers Are Here! leaves off, with chapters on Large-Scale Integration, how to choose a microprocessor chip, an introduction to programming, low-cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.* BK7340

AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. BK1130 $7.95.*

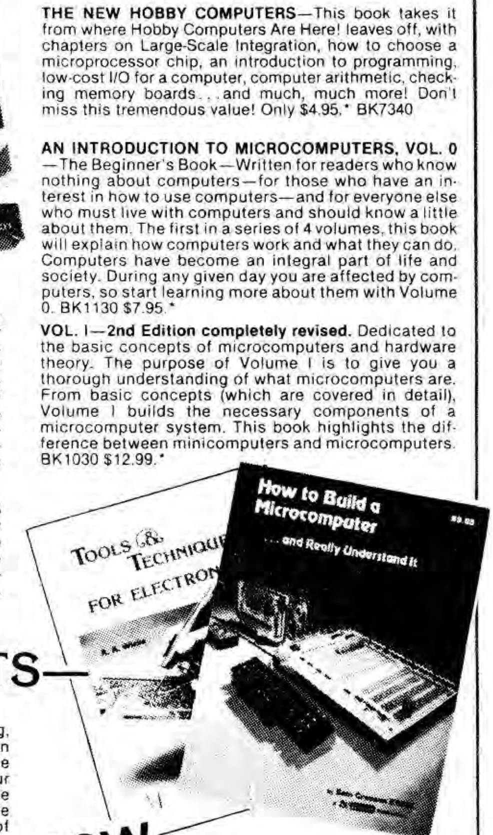
VOL. I—2nd Edition completely revised. Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. BK1030 $12.99.*

• **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved. $10.95*

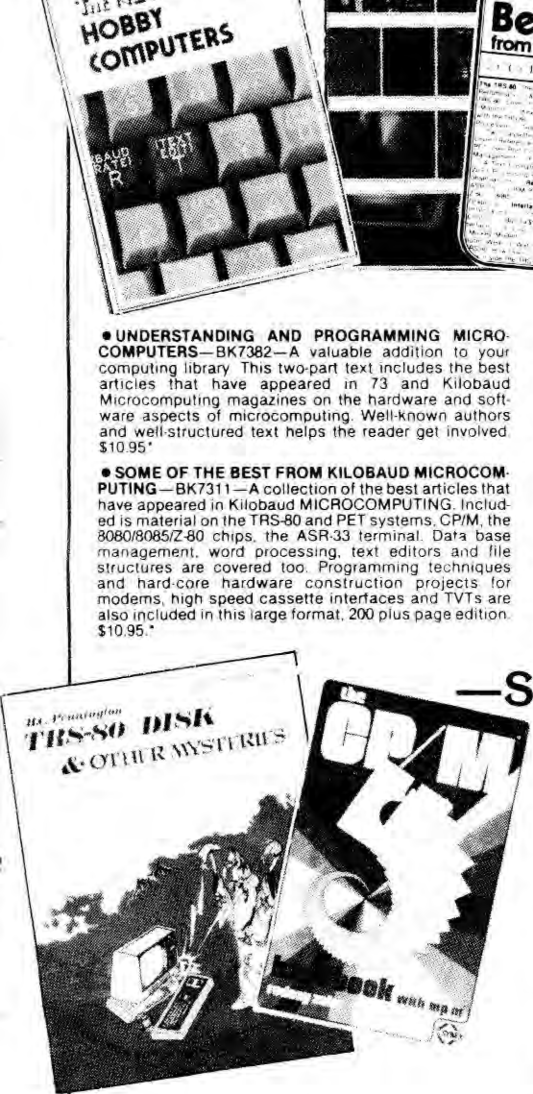
• **SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING**—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

YOUR FIRST COMPUTER—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) BK1191 $8.95*

MICROPROCESSOR INTERFACING TECHNIQUES—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. BK1037 $17.95.*



—SPECIAL INTERESTS—



MOD III ROM COMMENTED—Soft-Sector marketing, 1981. This book is not an instruction course on machine language, but rather an information source that you can use time and time again for writing your own program or patching old Mod I machine language programs. It contains an explanation of ROMs in the latest machine from Tandy, with most every location of the 14K ROMs listed, with comments. BK1235 $22.50.

THE CUSTOM TRS-80 AND OTHER MYSTERIES—by Dennis Kitsz. More than 300 pages of TRS-80 customizing information. With this book you'll be able to explore your computer like never before. Want to turn an 8 track into a mass storage unit? Individual reverse characters? Replace the BASIC ROMs? Make Music? High speed, reverse video, Level I and Level II? Fix it if it breaks down? All this and much, much more. Even if you have never used a soldering iron or read a circuit diagram, this book will teach you how! This is the definitive guide to customizing your 80! BK1218 $29.95.*

BASIC FASTER AND BETTER AND OTHER MYSTERIES—by Lewis Rosenfelder. You don't have to learn assembly language to make your programs run fast. With the dozens of programming tricks and techniques in this book you can sort at high speed, swap screens in the twinkling of an eye, write INKEY routines that people think are in assembly language and add your own commands to BASIC. Find out how to write elegant code that makes your BASIC really hum, and explore the power of USR calls. BK1221 $29.95.*

THE CP/M HANDBOOK (with MP/M)—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. BK1187 $14.95.*

TRS-80 DISK AND OTHER MYSTERIES—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to use," information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. BK1181 $22.50.*

MICROSOFT BASIC DECODED AND OTHER MYSTERIES—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*. Contains more than 6500 lines of comments for the disassembled Level II ROMs and six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. BK1186 $29.50.*

new

KILOBAUD KLASSROOM—by George Young and Peter Stark. Learning electronics theory without practice isn't easy. And it's no fun to build an electronics project that you can't use. *Kilobaud Klassroom*, the popular series first published in *Kilobaud Microcomputing*, combines theory with practice. This is a *practical* course in digital electronics. It starts out with very simple electronics projects, and by the end of the course you'll construct your own working microcomputer!

Authors Young and Stark are experienced teachers, and their approach is simple and direct. Whether you're learning at home or in the classroom, this book provides you with a solid background in electronics—and you'll own a computer that you built yourself! BK7386 $14.95

TOOLS & TECHNIQUES FOR ELECTRONICS—by A. A. Wicks is an easy-to-understand book written for the beginning kit-builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects, as well as specialized metal-working tools and the chemical aids which are used in repair shops. $4.95.* BK7348

HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing, and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.* BK7352

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book. $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 LIST OF ADVERTISERS

For further information from our advertisers, please use the Reader Service card.

READER SERVICE

This card is valid until March 31, 1982

Please help us to bring you a better magazine—
by answering these questions.

A. What is your occupation?
1 Professional
2 Engineer
3 Data Processing
4 Business
5 Education
6 Technician
7 Government
8 Student
9 Other

B. What are your primary interests in applications of your TRS-80? Rate all that apply on a scale of 1 (least) to 5 (most).
1 Business
2 Hobby
3 Home Management/Finance
4 Education
5 Science or Technical
6 Other

C. Is your computer use:
1 Connected with your occupation
2 Not connected with your occupation
3 Both of the above apply

D. What of type of TRS-80 do you own?
1 Model I Level I
2 Model I Level II
3 Model II
4 Model III Level I
5 Model III Level II
6 Color
7 Pocket Computer

E. On average, how many of each issue's program listings do you actually type into your microcomputer?
1 0-2
2 3-5
3 6-8
4 9 or more

F. Are you currently a member of a network?
1 Yes Which one?
2 No

G. To what types of software users groups do you belong?
1 Hardware Exclusive
2 General Club
3 College Organization
4 Other

H. From what companies have you purchased software?
1 Hayden
2 Hewlett-Packard
3 Instant Software
4 Microsoft
5 Personal Software
6 SAMS
7 Other

I. How many people read your copy of *80 Microcomputing*?
1 1
2 2
3 3
4 4 or more

J. Where did you obtain this copy of *80 Microcomputing*?
1 Subscription copy
2 Newsstand
3 Computer store
4 Friend
5 Other

K. Please enter the month and page number of your favorite *80 Microcomputing* article published during the last 12 months.

Month Page Number

L. On a scale of 0 (no interest) to 5 (great interest) please rate your interest in the following columns:
1 Remarks
2 Inside 80
3 80 Applications
4 Education 80
5 News
6 Exclusive 80
7 80 Accountant
8 Kitchen Table Software
9 The Assembly Line
10 Input
11 New Products
12 Reviews

M. If you are not a subscriber please circle 500.

80 Microcomputing • February 1982

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	151	156	161	166	171	301	306	311	316	321	451	456	461	466	471
2	7	12	17	22	152	157	162	167	172	302	307	312	317	322	452	457	462	467	472
3	8	13	18	23	153	158	163	168	173	303	308	313	318	323	453	458	463	468	473
4	9	14	19	24	154	159	164	169	174	304	309	314	319	324	454	459	464	469	474
5	10	15	20	25	155	160	165	170	175	305	310	315	320	325	455	460	465	470	475
26	31	36	41	46	176	181	186	191	196	326	331	336	341	346	476	481	486	491	496
27	32	37	42	47	177	182	187	192	197	327	332	337	342	347	477	482	487	492	497
28	33	38	43	48	178	183	188	193	198	328	333	338	343	348	478	483	488	493	498
29	34	39	44	49	179	184	189	194	199	329	334	339	344	349	479	484	489	494	499
30	35	40	45	50	180	185	190	195	200	330	335	340	345	350	480	485	490	495	500
51	56	61	66	71	201	206	211	216	221	351	356	361	366	371	501	506	511	516	521
52	57	62	67	72	202	207	212	217	222	352	357	362	367	372	502	507	512	517	522
53	58	63	68	73	203	208	213	218	223	353	358	363	368	373	503	508	513	518	523
54	59	64	69	74	204	209	214	219	224	354	359	364	369	374	504	509	514	519	524
55	60	65	70	75	205	210	215	220	225	355	360	365	370	375	505	510	515	520	525
76	81	86	91	96	226	231	236	241	246	376	381	386	391	396	526	531	536	541	546
77	82	87	92	97	227	232	237	242	247	377	382	387	392	397	527	532	537	542	547
78	83	88	93	98	228	233	238	243	248	378	383	388	393	398	528	533	538	543	548
79	84	89	94	99	229	234	239	244	249	379	384	389	394	399	529	534	539	544	549
80	85	90	95	100	230	235	240	245	250	380	385	390	395	400	530	535	540	545	550
101	106	111	116	121	251	256	261	266	271	401	406	411	416	421	551	556	561	566	571
102	107	112	117	122	252	257	262	267	272	402	407	412	417	422	552	557	562	567	572
103	108	113	118	123	253	258	263	268	273	403	408	413	418	423	553	558	563	568	573
104	109	114	119	124	254	259	264	269	274	404	409	414	419	424	554	559	564	569	574
105	110	115	120	125	255	260	265	270	275	405	410	415	420	425	555	560	565	570	575
126	131	136	141	146	276	281	286	291	296	426	431	436	441	446	576	581	586	591	596
127	132	137	142	147	277	282	287	292	297	427	432	437	442	447	577	582	587	592	597
128	133	138	143	148	278	283	288	293	298	428	433	438	443	448	578	583	588	593	598
129	134	139	144	149	279	284	289	294	299	429	434	439	444	449	579	584	589	594	599
130	135	140	145	150	280	285	290	295	300	430	435	440	445	450	580	585	590	595	600

Name ______

Address ______

City ______ State ______ Zip ______

BOOKS

80 Microcomputing
Peterborough NH 03458

Please send me the following 80 Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Shipping and handling charges:
$1.50 1st book, $1.00 each additional book
(UPS, use street address)
$10.00 each book overseas airmail

Shipping/handling ______

Total ______

Enclosed $ ______ Check M.O.
Bill: AE MC VISA

Card # ______

Exp date ______ Interbank # ______

Signature ______

Name ______

Address ______

City ______ State ______ Zip ______

2-82

Please allow 4–6 weeks for delivery No C.O.D. orders accepted.

SUBSCRIPTION

80 microcomputing

subscribers save $10 off the newsstand price.

☐ New subscription ☐ Renewal
☐ 1 year—$24.97

Enclosed $ ______ ☐ Check ☐ M.O.
Bill: ☐ MC ☐ Visa ☐ AE ☐ me

Card # ______ Exp. date ______
Signature ______ Interbank # ______
Name ______
Address ______
City ______ State ______ Zip ______

Canada—$27.97, 1 year only, US funds.
Foreign Surface—$44.97 1 year only, US funds.
Foreign Air—Please inquire.

322B89

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 T.M.

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 T.M.

Peterborough NH 03458

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 30 DALTON MA 01226

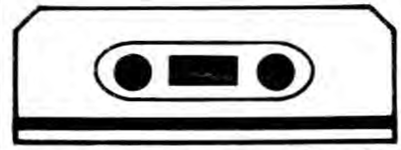
POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

P. O. BOX 306
DALTON, MA 01226

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

LOAD 80 LOADS EASY



LOAD 80

ELIMINATE HOURS OF TYPING AND AGGRAVATION

LOAD 80 cassettes for BACK ISSUES of 80 Microcomputing now available

In the April, 1981 issue of 80 Microcomputing, we introduced LOAD 80 to save you the time and trouble of typing our programs yourself. LOAD 80 cassette tapes contain dumps of the major program listings in 80 Microcomputing.

If you have not yet ordered any LOAD 80 tapes and wish you had, don't worry. We are now offering a "back issue" cassette program. You can order any LOAD 80 cassette from April, 1981 to date for $9.95. Back issues of 80 Microcomputing are also available with the LOAD 80 cassettes for $3.50. With the complete documentation found in the companion magazine issue, you should have no difficulty loading any of 80's major programs.

Please send me the following:

	month	year		month	year
Tapes	______	______	**Magazines**	______	______
	______	______		______	______
	______	______		______	______
	______	______		______	______

ITEM	COST	QTY.	TOTAL
LOAD 80 Tapes add $1.50 per tape for postage and handling	$9.95		
80 Microcomputing add $1.00 per magazine for postage and handling	$3.50		
		TOTAL	

☐ Check Enclosed Bill my: ☐ MC ☐ AE ☐ VISA

NAME ____________________
ADDRESS ____________________
CITY ____________ STATE ________ ZIP ________
CARD # ____________________ EXP. DATE ________
SIGNATURE ________________ INTERBANK # ________

Please allow 4-6 weeks for delivery
LOAD 80 • 80 Pine Street • Peterborough NH 03458
Attn: Debra Boudrieau

ATTENTION UNIVERSITIES AND MEDICAL RESEARCH FACILITIES

DO YOU NEED DETAILED STATISTICAL ANALYSIS?

StatPac is a new 3-part program written and designed for research, survey and questionnaire processing, data entry, statistical analysis, management and market analysis. StatPac can be easily tailored to fit your needs because you create the data file format. All phases of research analysis can be processed, including:

- frequency analysis
- crosstabs & chi-square
- correlation and linear regression
- multiple linear regression
- descriptive statistics
- t-test
- analysis of variance
- management reports

StatPac eliminates the need for expensive programming or renting time on large mainframe computers because it is the **SPSS** for the TRS-80's.

Total Package, including an extensive step-by-step user's manual is $285.00. Manual may be purchased separately for $30.00 (may be applied to the purchase price of the disks). MasterCard and Visa accepted.

Write or call for a **FREE** 16 page brochure (612) 866-9022.

Walonick Associates, Inc.
5624 Girard Ave. So.
Minneapolis, MN 55419

"Absolutely the best statistical analysis package I've seen for a microcomputer. . . ." Eugene Allred, President of Consolidated Data Services, Minneapolis, MN.

✓363

AT LAST!

Mass production prices for high quality software. Buy direct and save 50%. Also available for CPM and HDOS.

DATA BASE MANAGER Mod I & III $69. $149 (48K) Mod-II $199
Maintain a data base and produce reports, all without user programming. Define file parameters and report formats on-line. Key random access, fast multi-key sort, field arithmetics, audit log, label. No time-consuming overlays. 500 happy users in one year. Mod-II and 48K versions have over 50 enhancements, including 40 fields maximum. "IDM-M2 is great!" - 80-US.

A/R Mod-I $69 Mod-II $149 Mod-III $69
Handles invoices, statements, aging, sales analysis, credit checking, forms input, and order entry. Unlike other accounts receivable programs, ours can be used by doctors, store managers, etc.

WORD PROCESSOR $49
Centers, justifies, indents, and numbers pages. Mod-I version features upper/lower case without hardware modification! File merge option available.

MAILING LIST Mod I & III $59. $79 (48K) Mod-II $99
The best! Compare and be selective. Includes forms input, 5-digit selection code, zip code extension, sort on any field, and multiple labels. Who else offers a report writer and merges with word processor?

INVENTORY Mod I & III $89. $109 (48K) Mod-II $149
Fast key random access. Reports include order info, performance summary, EOQ and user-specified reports. Many people have converted to our system! "Next to impossible to damage the file."

GL. A/R. A/P. PAYROLL Mod-II $129 each
Integrated accounting package. 100+ page manual. As opposed to Osborne's slow binary search and 64 column screen, we use fast ISAM and 80 columns. Dual disk and TRSDOS required.

L216 $59
A cassette package of 10 business programs for Level II 16 K systems. Includes word processor and data base manager. Poker game $19.

Most programs are on-line, interactive, random-access, bug-free, documented, and delivered on disks. Mod-I programs require 32K TRSDOS. We're #1 in business software—don't let our low price fool you! Ask for our free 20-page catalog if you're still not convinced. Compiled versions are available.

MICRO ARCHITECT, INC. ✓54
96 Dothan St., Arlington, MA 02174

80 NEWS

edited by John P. Mello Jr.

Danger to civil rights?

Report warns computers may threaten constitutional rights

Civil rights in the future could be threatened by a bloodless adversary—the computer.

That's the opinion of the Congressional Office of Technology Assessment in a 116-page report released late last year.

"Extensive data collection and possibly surveillance by government and private organizations could, in fact, suppress or 'chill' freedoms of speech, assembly, and even religion by implicit threats contained in such collection or surveillance," the report said.

It maintained even innocuous data could bludgeon civil liberties.

"[O]ne does not usually attempt to keep secret the titles of books borrowed from a public library," the report noted. "However, an accurate profile of an individual's interests and attitudes could be provided by a complete dossier on that person's reading habits."

It said new definitions of public and private may be needed in the coming age.

"Information technology blurs the line between public and private activity," the Congressional researchers stated. "A nonelectric mail cover requires approval by the Postal Inspection Service but not a search warrant because only the outside of an envelope is examined. In an electronic mail system, however, no distinction may exist between the 'outside' (or address) and the 'inside' (or contents) of a message. Therefore, it may be difficult to distinguish a mail cover from a wiretap, which would require a warrant issued by a court. . . ."

The same may hold true for surveillance. The report said: "[T]he observation of shopping habits by following a person from store to store is surveillance. However, the use of an electronic funds transfer system to gather the same type of information would be far more intrusive, since much more data, some of it of a highly personal nature, could be collected in secret."

Even the Sixth Amendment—the right to be tried by a jury of one's peers—could come under attack.

"A small industry has grown up around the use of computerized dossiers of potential jurors along with models predicting juror behavior," the report noted. "At this time the technology is very expensive and its value controversial. . . .

"However, future computer technology will make this application cheap, and far more personal data about potential jurors will be available, legally or illegally.

"Furthermore, there is sufficiently sound social scientific basis underlying this type of use to suggest that predictive techniques will be likely to improve ineffectiveness. If so, the entire concept of an 'impartial' jury as required by the Sixth Amendment may be challenged."

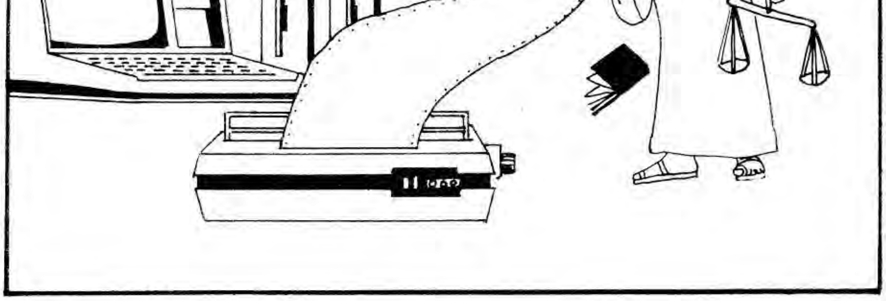
These predictive techniques will bring other constitutional amendments into question, the report contended.

It said: "As social scientists improve this predictive capability, important questions of Fifth and Fourteenth Amendment rights will be raised. Essentially, individuals may be denied rights, privileges and benefits not on past performance, but on a prediction of future tendencies."

In the past, the courts have intervened on behalf of the individual in these kinds of cases, but the report suggested even judges are having difficulty when advanced technology is involved.

The Congressional researchers cited a U.S. Supreme Court case where a person's electronic bank records were determined to be the bank's property, not the person's.

continued



Color computer owners,

32K PLUS DISKS*

$298.00

Yes, that's right - for as little as $298.00 you can add 32K of dynamic RAM, and a disk interface, to your TRS-80 Color Computer! If you just want the extra memory it's only $199.00, and you can add the disk interface later for $99.00.

Just plug the *Color Computer Interface (CCI)*, from Exatron, into your expansion socket and "Hey Presto!" - an extra 32K of memory. No modifications are needed to your computer, so you don't void your Radio Shack warranty, and Exatron give both a 30 day money-back guarantee and full 1 year repair warranty on their interface.

The *CCI* also contains a 2K machine-language monitor, with which you can examine (and change) memory, set break-points, set memory to a constant and block-move memory.

So what about the *CCI Disk Card*? Well as we said it's only an extra $99.00, but you'll probably want Exatron's *CCDOS* which is only $29.95 – unless you want to write your own operating system. The *CCI Disk Card* uses normal TRS-80 Model I type disk drives, and *CCDOS* will even load Model I TRSDOS disks into your color computer – so you can adapt existing TRS-80 BASIC programs.

As a further plus, with the optional *ROM Backup* adaptor, you can dump game cartridges to cassette or disk. Once the ROM cartridge is on cassette, or disk, you can reload, examine and modify the software. The *ROM Backup* adaptor is only $19.95.

For more information, or to place an order, phone Exatron on their Hot Line *800-538 8559* (inside California 408-737 7111), or clip the coupon.

excellence in electronics

exatron

DEALER ENQUIRIES INVITED

Exatron, ✓187
181 Commercial Street,
Sunnyvale, CA 94086

☐ Please send a 32K Color Computer Interface for $199.00

☐ Please send a CCI Disk Card for $99.00

☐ Please include CCDOS and manual for $29.95

☐ Also include a ROM Backup adaptor for $19.95

Please add $5.00 for shipping to all orders, and 6 percent sales tax in California.

Name ....

Address ....

City ....

State .... Zip ....

Charge my:

☐ MasterCard Interbank Code ....

☐ Visa Expiration Date ....

Card ....

☐ Check enclosed for ....

☐ Ship COD ($2.00 extra)

Signature ....

*32K RAM plus Disk Interface TRS-80 is a trademark of Tandy

CIVIL RIGHTS *continued*

In the 1976 case, *United States v. Miller*, the Court said the use of a bank account was a voluntary action.

"Yet," the report observed, "it is questionable whether future participation in a computerized society can be construed to be voluntary if the alternative is to forego

all services necessary to live comfortably as a member of that society.

"Extensions of such reasoning could leave only a hollow shell of Fourth Amendment protection for personal records, while eroding any substantive effective barriers against government intrusion."

The report added computers are bound to have a long-term effect on the right to privacy.

It said: "The nature of societal values attached to privacy in the United States may change if larger and more ubiquitous information systems gradually remove the ability of individuals to hide their private activities. . . .

"It has been pointed out that the possession by large organizations of personal data on individuals enhances the power, real or perceived, of the organization over the person. These and similar effects may increase the suspicion some citizens have of large organizations—business, labor or government—and thus erode social cooperation and a personal sense of well-being.

"The self-image held by humans of their uniqueness, distinguished by their ability to think, may be threatened by the association with machines that increasingly demonstrate apparent characteristics of intelligence. . . .

"One effect of such a perception may be to increase the uncritical reliance on computers."■

PC helps handicapped

A $10,000 prize has been awarded the designer of a Pocket Computer system which enables deaf people to communicate on the telephone.

The $10,000 was first prize in the "Personal Computing to Aid the Handicapped" contest sponsored by Johns Hopkins University's Applied Physics Laboratory.

The Pocket Computer application was designed by City University of New York professor of hearing and speech Dr. Harry Levitt. He vied with more than 900 programmers and inventors for the prize.

Weighing three pounds and running on batteries, Levitt's teletype emulator consists of a message-handling protocol and a cassette port adaption which enables coupling of the PC to a modem. This allows the computer to send and receive messages from similarly adapted kin. The messages may be reproduced on the computer's display screen or on a line printer.

With modifications, the computer's output can be translated to Baudot code enabling it to talk to the standard teletypwriters in the homes of many deaf people.

The $3,000 second prize went to a device allowing severely handicapped persons to operate a computer through eye movements. Psychology professor Mark B. Friedman headed the team from Carnegie-Mellon University in Pittsburgh that developed "Eyetracker," which uses sonar and microcomputer technology.

The $1500 third prize was won by Robin L. Hight, a software technician with Surface Systems Inc. of St. Louis, MO, who designed a lip-reading instructional program for the Apple II.

Other devices competing in the contest included a head-movement controlled wheelchair, braille word processor and text-to-speech synthesizer.■

Making cash and trouble

Computer-based information services eventually will be the top money maker in the data processing market, according to a report released by Congress's Office of Technology Assessment.

The 166-page report—*"Computer-Based National Information Systems: Technology and Public Policy Issues"*—noted:

"The computer industry has traditionally been concerned with selling hardware. . . However, current trends in both pricing and structure of the market are driving the emphasis toward providing computer-based information services, such as bibliographic and data base searches, electronic publishing, electronic banking, and the like."

The Congressional researchers added: "Eventually, information services of all kinds will dominate the data-processing market in terms of dollar volume."

But some weeds are invading the clover.

"As information becomes a more valuable commercial commodity," the report said, "increasing tensions are arising between those who wish to sell it through new information services, and those who recommend that the government take steps to prevent the social inequity that would possibly result from the increasing cost of access to information and the means to use it."

"Some observers have suggested," the researchers wrote, "that the advent of information technology will widen the gulf between the haves and the have-nots in society. This view is based on relative differences in what might be called 'information literacy,' the ability to use information technology to cope with everyday life."

This "information gap" could create social barriers for the have-nots.

In the workplace, information illiteracy could result in unemployment. The report observed, "If projections about an evolving information society and economy are correct. . . information literacy could become an essential requirement for entering the labor market."

Vis-a-vis government, have-nots without information knowledge may be unable to exercise basic rights like voting, be at a serious disadvantage in legal proceedings, and be barred access to services like welfare, health care, and educational benefits.

"To the extent that information illiteracy is unevenly distributed among cultural or economic groups," the report said, "the consequences would fall disproportionately on the poor and disadvantaged, in general.

"The gulf between the haves and have-nots in society might not only be increased but, due to such barriers as a more limited access to jobs, societal efforts to bridge that gulf would be frustrated."■

TRS-80™ DISCOUNT - BUY DIRECT

WE SELL THE FULL LINE OF **TRS-80** AT WHOLESALE PRICES

MODEL II

26-4002 64K I Drive **$3288**
Ask About Hard Drives

MODEL III

26-1062 16K **$849**
26-1066 48K with
2 Drives, RS232 .... **$2069**

COLOR COMPUTER

26-3001 4K **$318**
26-3002 16K Ext. Basic **$488**
26-3003 32K Ext. Basic **$578**

POCKET COMPUTER

26-3501 Pocket Computer **$188**

TRS-80 COLOR COMPUTER
DISK DRIVES

26-3022 Disk Drive #1 .... **$498**
26-3023 Disk Drive #2, 3, 4 .... **$338**

LEARN TO
"DO IT YOURSELF"
AND SAVE

THOUSANDS OF SATISFIED CUSTOMERS

WRITE FOR YOUR
FREE CATALOG

ORDER TOLL FREE

1-800-841-0860

•LARGE INVENTORY
•FAST DELIVERY

TM - TANDY CORPORATION
FREE COPY OF WARRANTY UPON REQUEST

MICRO MANAGEMENT SYSTEMS, INC.
DEPT. NO. 4
115 C. SECOND AVE. S.W.
CAIRO, GA. 31728
GA. 912-377-7120

TRS-80™ DISCOUNT SOFTWARE TRS-80™

MODEL III SOFTWARE

26-1563 Scripsit I & III	**$79.00**
26-1566 Visicalc I	**$83.00**
26-1567 Visicalc III	**$83.00**
26-1552 General Ledger	**$90.00**
26-1553 Inventory Ctl.	**$90.00**
26-1556 Payroll	**$180.00**
26-1555 Accts. Receivable	**$135.00**
26-1554 Accts. Payable	**$135.00**
26-1558 Bus. Mail List	**$90.00**
26-2201 Fortran I	**$90.00**
26-1562 Profile	**$72.00**
26-1592 Profile III Plus	**$180.00**
26-1565 Microfiles	**$90.00**
26-1569 Visicalc	**$180.00**
26-1590 Super Scripsit	**$180.00**
26-1591 Scripsit Dictionary	**$135.00**
26-1582 Time Manager	**$90.00**
26-1580 Project Manager	**$90.00**
26-1581 Personnel Manager	**$90.00**
26-1568 Medical Off. Syst.	**$270.00**
26-2203 Cobol	**$180.00**

COLOR COMPUTER SOFTWARE

MODEL II SOFTWARE

26-4515 Profile Plus	**$270.00**
26-4511 Visicalc II	**$265.00**
26-4512 Profile II	**$162.00**
26-4501 General Ledger	**$180.00**
26-4502 Inventory Mgmt.	**$180.00**
26-4503 Payroll	**$360.00**
26-4504 Accts. Receivable	**$270.00**
26-4505 Accts. Payable	**$270.00**
26-4507 Mail List II	**$108.00**
26-4701 Fortran	**$270.00**
26-4703 Cobol	**$270.00**
26-4705 Compiler BASIC	**$180.00**
26-4540 Statistical Analysis	**$90.00**
26-4508 Medical Off. System	**$675.00**
26-4509 Manuf. Inventory	**$675.00**
26-4513 Job Costing	**$135.00**
26-4520 Time Accounting	**$450.00**
26-4621 Personnel Search	**$270.00**
26-4560 Westlaw	**$225.00**
26-4531 Scripsit 2.0	**$360.00**
26-4534 Scripsit Dictionary	**$180.00**

POCKET COMPUTER SOFTWARE

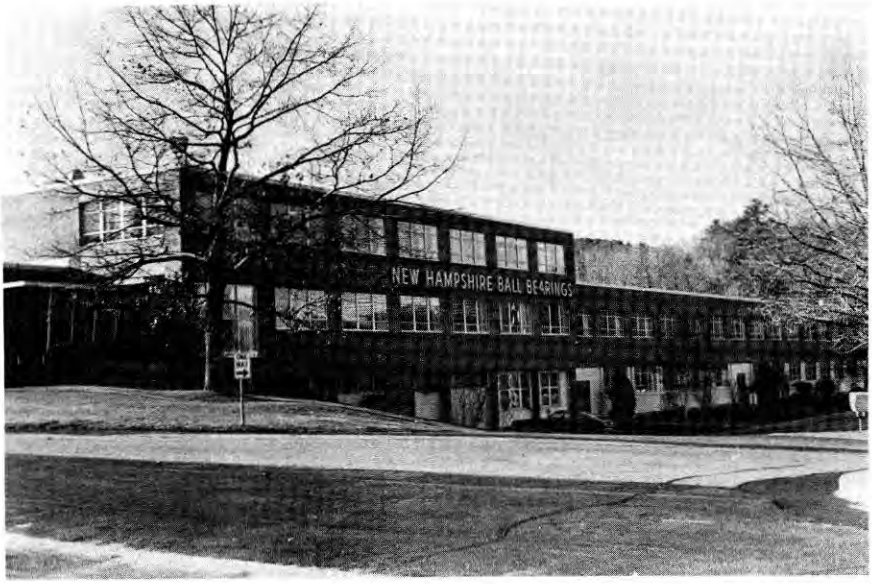
✓72

✓See List of Advertisers on page 290

Tandy micro works for ball-bearing leader

by Kerry Leichtman
80 Micro Staff

Amidst the smell of oil and noise from clanging machinery sits a 4K TRS-80



Viola Bochu (above) mans TRS-80 Model I at New Hampshire Ball Bearings (below). Photos by Paul Babich, Wayne Green Inc.

New Hampshire Ball Bearings, reputed to be one of the world's largest manufacturers of ball bearings, produces bearings for everything from roller skates to national defense systems. Approximately 1200 people shape, cut, finish, refinish, assemble and test the firm's product using huge and staggeringly expensive machinery.

Amidst the smell of oil and noise from clanging machinery sits a 4K TRS-80 Model I used to print labels.

Before the microcomputer, each label had to be typeset individually. Five labels took one hour to produce. The TRS-80 has been putting out 8,000 a day since its installation in January 1981.

When it came time to buy a new $16,000 printing press, Jim Sargent, then in the engineering department and now machine maintenance supervisor, suggested a microcomputer. He shopped around and decided on the TRS-80 interfaced to a DMTP-6 paper roll printer, manufactured by Practical Automation Inc., Shelton, CN.

Sargent installed an infrared detector to position the labels, modified some of the printer's circuitry, built a printer interface and wrote the Basic program.

Data is entered into the computer, displayed on the screen and sent to the printer. The printer prints a line when its 20-character buffer is filled, or when the print command is given. When the buffer is full, the computer receives a busy signal which stops it from sending more data until the line is printed.

The program allows a maximum of six 20-character lines; completes the 20-character requirement with spaces as needed; computes the number of labels to print per job; checks the label load operation; makes sure the proper number of labels has been printed; and includes an edit function. Only two commands are used: Advance Paper and Print.

The TRS-80 is popular with the many people who operate it. Sargent said, "I've had a number of people come in and want to take the thing home."■

Does your **CP/M**** or **TRS-80*** Word Processor need help?

Aspen Software has the finest document proofreading tools available.

GRAMMATIK™

Beyond Spelling Checking

Spelling checkers are useful, but they are not enough! Grammatik can find many errors that a spelling checker can't. It detects many errors commonly found in text entered on computers, such as doubled words ("the the"), inconsistent capitalization ("STicky shift key"), incorrect punctuation, and others. That's not all! Grammatik also checks your document for good writing style using a dictionary of over 500 misused phrases as defined in many writer's style manuals. It marks and classifies the problems it finds in the document file for easy correction with your word processor, and provides suggestions for correcting the problems. The phrase dictionary can be easily extended to include checking for esoteric jargon or your own pet peeves. Grammatik also collects other stylistic information that can be used to revise the document to improve its readability such as average sentence and word length. It can produce a profile with the number of times each unique word in the document was used, helpful for identifying overworked vocabulary. Grammatik is not only a valuable proofreading tool, it is also a useful learning tool. You will notice significant improvements in your own writing style after using Grammatik for only a short time. Grammatik is fast, easy to use, and works with all popular TRS-80 and CP/M word processors. Model I/III version requires a minimum 32K, 1 drive system. Model II TRSDOS* version requires 64K and 1 drive. CP/M version requires CP/M release 2.2 and 48K.

PROOFREADER™

The Aspen Software Company Spelling Checker.

Don't buy a spelling checker until you've considered Proofreader. Recently, several ads for other spelling checkers have compared themselves to "others". They weren't comparing themselves to Proofreader! Proofreader has all the features you need for checking your documents for spelling errors and typos. Proofreader looks up every word in its 38,000 word dictionary, and does not increase its "vocabulary" by using less accurate root word analysis like some others do. You won't need to spend as much time adding new words as you would with a smaller dictionary. Proofreader is easy to use -- you can start checking your documents immediately. Proofreader is fast -- it can check even your largest document (20 pages or more) in less than 5 minutes! Unknown words are listed on the screen, and can be saved in a file for later manipulation. The Proof-Edit feature (optional on the Model I/III version, included with Model II and CP/M) allows you to interactively correct the unknown words in context. New words can easily be added to the dictionary, and expansion is limited only by disk capacity. Proofreader works with all TRS-80 operating systems and word processors, so if you change systems, you won't need a new spelling checker. On the TRS-80 Model I/III, only 32K RAM and 1 disk drive are needed. Proofreader also works with all popular CP/M word processors. Add up the facts and the low price, and you will conclude that Proofreader is the best value available in spelling checkers.

Aspen Software programs are professional quality, reliable software tools developed for the TRS-80 and CP/M by a Ph.D. in Computer Science. All software is protected by Aspen Software's low cost upgrade privilege for new versions. Other tools include:

— SOFT-SCREEN™, a powerful, state of the art full screen text editor. Over a year in development, Soft-Screen is compatible with all TRS-80 programming languages, including BASIC, FORTRAN, MACRO, Ratfor, and COBOL. Commands are easy to learn, yet versatile and complete to satisfy the most experienced user. Soft-Screen is also available for P&T Model II CP/M.

— SOFT-TEXT™, Aspen Software's text formatter. When used with Soft-Screen, provides a powerful word processing system. Full featured, including automatic pagination, even and odd page headings, underlining, index generation, footnotes, support for advanced printer capabilities, and much more. Model I/III version supports serial printers at full speed. Soft-Text offers a real alternative for Model II TRSDOS users. Please write or call for more details.

— RATFOR, a structured language preprocessor for Fortran developed at Bell Labs. Aspen Software Ratfor is one of the best versions available, and the only one with a pretty printer option. Totally compatible with Microsoft F80. Includes several extensions, including "case", "string", and conditional compilation. User's manual contains all information needed to learn and write Ratfor programs. Requires FORTRAN.

— PP-RATFOR, a pretty printer. Automatically formats and indents Aspen Software Ratfor source programs. An essential program development tool.

	Model I	Model II (64k,1d)	Model III	CP/M (2.2,48k)	Manual only
Proofreader	$54.00(32k1d)	$119.00	$64.00(32k1d)	$129.00	$8.00
Proof-Edit	$30.00	incl.	$30.00	incl.	incl.
Grammatik	$59.00(32k1d)	$99.00	$59.00(32k1d)	$149.00	$8.00
Soft-Screen	$69.00(48k1d)	$99.00	$75.00(48k1d)	$99.00(P&T)	$15.00
Soft-Text	$69.00(48k1d)	$99.00	$75.00(48k1d)	$99.00	$15.00
both	$129.00	$179.00	$139.00	$179.00	$25.00
Ratfor	$49.00(48k2d)	$99.00	$59.00(48k1d)	$99.00	
PP-Ratfor	$30.00(48k2d)	$49.00	$34.00	$49.00	
both	$74.00(48k2d)	$139.00	$84.00	$139.00	$15.00

IMPORTANT: Specify computer model, operating system, memory size, and number of drives when ordering! For CP/M, currently only 8" single density CP/M versions available. Please inquire about other CP/M disk formats. All TRS-80 versions available. Manual only orders can be applied to final purchase. CP/M prices are introductory.

Orders sent postpaid by first class mail. Terms: Cash, check, money order, VISA, or Master Card. NM residents add 4% tax.

Proofreader, Grammatik, Soft-Screen, and Soft-Text are trademarks of Aspen Software.

**trademark of Digital Research *trademark of Tandy Corp.

VISA MasterCard

ASPEN SOFTWARE COMPANY™

P.O. Box 339-M Tijeras, NM 87059 (505) 281-1634

449

See List of Advertisers on page 290

That mainframe attitude

Firm gives macro service to micro owners

They've brought a mainframe attitude to microcomputer software and it is paying off in big bucks.

Two data processing professionals—George Rosenbergen and Howard Wolowitz—put their mainframe know-how together in 1979 to form The Small Computer Company. In three years, their firm's gross revenue mushroomed from zero to a quarter of a million dollars.

Small Computer's origins lie in a program called "Electric File Clerk."

Rosenbergen and Wolowitz—formerly with Group Health Insurance of New York, a large health insurance business similar to Blue Cross—had worked with medical billing programs and realized the potential for the programs if adapted for microcomputers. They knew physicians would jump at the opportunity to install their own systems...if they were simple enough.

They decided to write their programs for the TRS-80 Model II. Wolowitz felt it had more computing power than some IBM computers he had worked with, and its price tag was a fraction of the one for a mainframe.

Wolowitz developed a set of modules to perform repeated functions rather than rewriting the code over and over again. After five months, Wolowitz completed "Electric File Clerk," which he and Rosenbergen marketed successfully to numerous medical groups and private practices on the East Coast.

Howard Wolowitz of the Small Computer Company working in the firm's program development shop.

The pair realized Electric File Clerk was more than a clever way to do medical billings and they began to look for a way to market it to the world.

Wolowitz attended the opening of the Manhattan Radio Shack computer center. A Radio Shack salesman, Bill Prady, spotted Wolowitz demonstrating Electric File Clerk and called over two division managers. They brought a copy to Fort Worth.

Within weeks a contract had been signed with Tandy which marketed Electric File Clerk through 8,000 Radio Shack stores as "Profile II." During its first year on the market it sold 15,000 copies.

Small Computer was on its way. During 1980 and 1981, Radio Shack awarded it nine contracts, including one for a Model III version of its Profile II. By the end of 1982, the firm expects 40,000 customers to buy its software through Radio Shack.

The company has developed a series of application packages built on Profile II. Those include packages for office management, jewelry management, accounts receivable, accounts payable and personal financial management. In early 1982, the firm expects to release dental and garment center inventory control packages.

Coming attractions from Small Computer include:

- Profile Proform, which formats a Model II disk twice as fast as Radio Shack's format program;
- Profile Extended Sort, a fast sort for hard disks;
- Mass Utility Programs, which deletes, purges and prints all records meeting specified criteria, rather than going through the data base on a record-by-record basis;
- Archive, which goes through a data base extracting records meeting specified criteria and puts them in an archival storage file; and
- 3741 Emulator, which creates data in standard IBM disk format allowing the Model II to function as an input device for IBM computers.

Small Computer also began to acquire a reputation as a company treating microcomputerists with the respect usually reserved to large equipment users. Soon, the firm began providing customized programming services. Through word of mouth, Small Computer was tailoring software for more than 300 customers by

Continued on page 302

SMALL FIRM*continued*

September 1981. Its customers include Citibank, Estee Lauder, U.S. Coast Guard, U.S. Lease Corp., Oppenheimer and other companies in the New York City area.

Wolowitz said of the custom software end of the business:

"When a businessman buys a micro, it's like buying a portable phono, rather than a hi-fi—it's all he can afford. But there's usually no place to turn for professional software, so he turns to the part-timer or freelancer.

"He should have an opportunity to go to a professional corporation, to have software developed with the attitude and the care that goes into big computers.

"There's an unspoken attitude that people using micros don't have data that is as important as the data that's running on the big computers. It is *just* as important and the programs need to be written just as well.

"That's our goal: to bring the professionalism of the big computer to the small computer."■

From left to right: Bill Prady, Howard Wolowitz and George Rosenbergen of the Small Computer Company.

Reveille for the Color Computer

Vermont College marches to the beep of a different drummer

by Kerry Leichtman
80 Micro Staff

A pair of Color Computers have become the keystones of the computer science program at a 1600-student university in Vermont.

The Color Computer's power and versatility convinced the university to buy the machines, said Ron Lessard, who teaches electrical engineering at Norwich, a private school with a military bent offering science, engineering and business degrees, and a full liberal arts curriculum.

"When I saw what it could do with graphics," Lessard observed, "I said, 'That's it. This is the computer for engineers.' "

He explained when he, Greg Shadel, who teaches mechanical engineering, and their department head viewed the Color Computer, they were so impressed they not only bought two for the school but each personally bought a unit.

Asked about the Color Computer's potential for teaching micro-technology, Lessard noted: "I was quite impressed by it. Radio Shack has left all the flexibility and options open. It's a good machine."

Until recently Norwich offered computer sciences only as a minor. Now, coinciding with the acquisition of two TRS-80 Color Computers, Norwich is offering a full major in computer science engineering.

Before the Color Computer, Ron and his colleagues taught computer studies using a Motorola D-2 kit. The Motorola is a basic set-up. Its advantages are in its primitive simplicity. In teaching computer fundamentals, the Color Computer is too sophisticated. By using the D-2 kits, the students can learn more of what is going on at each instruction.

Third year students continue to use the D-2's as they continue to learn such computer functions as transfers between input and output and how and why a bus is where it is for a particular transfer. They learn in both machine and assembly languages.

In their fourth year, the students are ready for the Color Computers. One of the senior projects is driving an IBM/Wang terminal typewriter with the Color Computer. The student is designing and building the interface hardware to do the task. Another senior is converting the Color Computer into a digital oscilloscope.

And another group of students is using the Color Computer to monitor lumber drying. They are hooking up sensors to measure the moisture and temperature of the lumber and the kiln. The temperature reading is measured in analog voltage and then converted to a digital signal. The digital signal is then sent to the computer. The moisture reading is obtained by measuring the electrical resistance of the lumber. Both measurements have to be taken simultaneously and computed.

The Color Computers are new at Norwich and are being phased into the program. The aim is not to use them to replace the D-2 kits, but to use them as an illustration of what a microcomputer is capable of. At the same time, Lessard and the rest of the engineering department are receiving an education themselves. "I'm learning that this machine has a lot of potential. We're going to go much farther with it than I originally planned. We were talking this morning about designing our own network of Color Computers. Our primary objective is teaching. We could teach how to design a network . . . Part of the beauty in this machine is its simplicity of design and the options they've left open to the user. If the people who write software realize the potential of this machine, it could really take off."■

PULSE TRAIN

Pirate busters Want members.

The Association for Software Protection announced it is accepting membership applications for 1982. The organization, aimed at software black marketeers and pirates, started last June. Services it plans to offer include a bi-monthly newsletter, an information library on international software laws and customs, research on software protection subjects, education for end-users, referrals of end-users, and conferences and seminars. Associate memberships cost $150 a year; participatory memberships, $350 a year. Applications and additional information may be obtained by contacting Robin Robinson, President, Association for Software Protection, 10143 Leona Ave., Tujunga, CA 91042, or Adrienne Webb, 3336 Bradshaw Rd., Suite 340, Sacramento, CA 95827.

Tandy's sales Increase

Tandy Corporation's consolidated sales in November 1981, $176,321,000, were 16 percent higher than the firm's sales in November 1980, $151,811,000. Slightly higher gains for the month were recorded by Tandy's U.S. Radio Shack division: $141,828,000 in November 1981, $121,264,000 in November 1980—a 17 percent increase.

TRS-80 Discount coupons

A Florida engineering software firm is offering coupons that will knock 10 percent off the price of any TRS-80 computer system. McClintock Corp. of Miami offers the coupons to anyone buying $1000 of the company's software.

Electronics Firms jumping To Bayou state

A recent report released by the Louisiana Department of Commerce shows more than birds have been heading south over the past decade. Since 1970, the number of electronics firms in the state has increased 72 percent, and people employed in the electronics industry 84 percent. According to the study, cost savings to incoming firms are numerous, but "the primary resource that is most important is its human resources—men and women capable and willing to work if given job opportunities."

FCC looking At Baby Bell

The Federal Communications Commission is studying a plan submitted in December by the American Telephone and Telegraph Co. to set up an unregulated subsidiary of the phone company. The proposed company—for the moment dubbed XYZ Inc., but known in the industry as "Baby Bell"—would initially offer customers "advanced communications services" allowing computers to talk with each other over long distance lines. The service also allows customers to manage their own data communication networks and to process, store and transmit data through the new firm's facilities. The phone company hopes to get FCC approval in time to start the new venture by March 31. The new firm will be given a $3 million cash advance from AT&T; physical assets worth $56 million; and 850 employees.

Money-saving Service at CompuServe

Refundle Bundle, a clearinghouse for consumer information about making and saving money through manufacturer's refund offers, is now an information provider on the CompuServe Information Service. The clearinghouse offers hints and tips on what to save from packages and how to save it, definitions of terms involved in refunding and descriptions of the various kinds of refund offers. An interactive section allows readers to answer specific questions.

Pocket computers May take off soon

Pocket computers may soon start eating into the market share held by programmable calculators, according to a report in the *Wall Street Journal*. The *Journal* said the introduction of a new pocket product by Panasonic could mark the beginning of the end for advanced calculators. The most important next step for the pockets, the report noted, was plug-in programs. Tandy and Casio plan to offer those types of programs by the middle of this year. Quoting Egil Juliussen, a partner in a Dallas consulting company called Future Computing, the report said, "hand-held computers will slow the growth of programmable calculators beginning next year and stop it within the next two or three."

Yet another Beeper show

Northeast Expositions Inc. of Chestnut Hill, MA, which sponsors the national computer shows, announced it will produce a series of new shows, Electronica, in the top 10 markets starting this fall. The shows will be expositions of computers, electronic games, ham radio equipment, TVRO, do-it-yourself electronics, CB, cameras and home entertainment products—projection tv, stereo equipment, compact audio and video equipment.

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

FIELD PROVEN!!

10 MEGABYTES and MORE for the TRS-80* Model II plus SHARED ACCESS to HARD DISK DRIVE

"NEW MODEL II DRIVES AVAILABLE IN JANUARY—COMPETITIVE PRICES."

Hard/Soft Disk System (HSDS) Software allows access as single drive. You can have that 10 Megabyte continuous file - that 50,000 name maillist or inventory! Or a directory with 1000 entries! All completely compatible with TRSDOS 2.0 BASIC. You can mix floppy and hard disk drives. Includes special utilities including HPURGE, DCS Directory Catalog System, HZAP Hard Disk Superzap, and many special formatting options. Three to eight times faster than floppy! RACET quality

HARD DISK DRIVE & CONTROLLER $5995. Second User $595. HSDS Software $400. (Note: HSDS now also available for CORVUS drives!!)

KFS-80 (1-drive 32K Min - Mod II 64K) **Mod I and III $100, Mod II $175**
Assembly language ISAM facility! Interfaces with BASIC. Very fast access to records in large files. Access/insert times 1-4 seconds. B-Tree directory structure eliminates record search. No file reorganization required. Sector buffering reduces media and drive wear while increasing performance. Can be used for primary and secondary file database applications. Records are kept in sorted order on disk by alpha key. Deferred write option. Comprehensive file maintenance and debugging utilities are included. **Attention applications programmers -** Save time in implementing your custom database application software.

MAILLIST (1-drive 32K Min - Mod II 64K) **Mod I and III $75, Mod II $150**
This ISAM-based maillist minimizes disk access times. Four keys — no separate sorting. Supports 9-digit zip code and 3-digit state code. Up to 30 attributes. Mask and query selection. Record access/update time 1-4 seconds!!

DISCAT (32K 1-drive Min) **Mod I and III $50**
This comprehensive Diskette Cataloguing/Indexing utility allows the user to keep track of thousands of programs in a categorized library. Machine language program works with all TRSDOS and NEWDOS versions. Files include program names and extensions, program length, diskette numbers, front and back, and diskette free space. RS232 drivers and other features.

LPSPOOL (32K 1-drive Min) **Mod I $75**
LPSPOOL — Add multi-tasking to permit concurrent printing while running your application program. The spooler and despooler obtain print jobs from queues maintained by the system as print files are generated. LPSPOOL supports both parallel and serial printers.

BASIC LINK FACILITY 'BLINK' (Mod I Min 32K 1-disk) **Mod I $25; Mod II $50; Mod III $30**
Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be **merged** by statement number. The statement number where the chained program execution is to begin may be specified!

INFINITE BASIC (Mod I & Mod III Tape or Disk) **Mod I $50; Mod III $60**
Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **Mod I & III $30**
Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, −, *, /. Binary search of sorted and unsorted arrays. Hash codes.

COMPROC (Mod I & Mod III — Disk only) **Mod I $20; Mod III $30**
Command Processor. Auto your disk to perform any sequence of instructions that you can give from the keyboard. DIR, FREE, pause, wait for user input, BASIC, No. of FILES and MEM SIZE, RUN program, respond to input statements, BREAK, return to DOS, etc. Includes lowercase driver software, debounce and screenprint!

GSF (Mod I & III Tape or Disk - Specify Memory Size) **Mod I $25; Mod II $50; Mod III $30**
Generalized Subroutine Facilities. The STANDARD against which all other sorts are compared! And then compare prices! Machine language — fast and powerful! Multi-key multi-variable and multi-key character string. Zero and move arrays. Mod II includes USR PEEKS and POKES. Includes sample programs.

DSM (Mod I Min 32K 2-drive system. Mod II 64K 1-drive. Mod III Min 32K 1-drive) **Mod I $75; Mod II $150; Mod III $90**
Disk Sort/Merge for RANDOM files. All machine language stand-alone package for sorting speed. Establish sort specification in simple BASIC command File. Execute from DOS. Only operator action to sort is to change diskettes when requested! Handles multiple diskette files! Super fast sort times — improved disk I/O times make this the fastest Disk Sort/Merge available on the TRS.

UTILITY PACKAGE (Mod II 64K) **$150**
Important enhancements to the Mod II. The file recovery capabilities alone will pay for the package in even one application! Fully documented in 124 page manual! XHIT, XGAT, XCOPY and SUPERZAP are used to reconstruct or recover data from bad diskettes! XCOPY provides multi-file copies, 'wild-card' mask select, absolute sector mode and other features. SUPERZAP allows examine/change any sector on diskette including track-0, and absolute disk backup/copy with I/O recovery. DCS builds consolidated directories from multiple diskettes into a single display or listing sorted by disk name or file name plus more. Change Disk ID with DISKID. XCREATE preallocates files and sets 'LOF' to end to speed disk accesses. DEBUG!! adds single step, trace, subroutine calling, program looping, dynamic disassembly and more!!

BASIC CROSS REFERENCE UTILITY (Mod II 64K) **$50**
SEEK and FIND functions for Variables, Line Numbers, Strings, Keywords. 'All' options available for line numbers and variables. Load from BASIC — Call with 'CTRL'R. Output to screen or printer!

DEVELOPMENT PACKAGE (Mod II 64K) **$125**
Includes RACET machine language SUPERZAP, Apparat Disassembler, and Model II interface to the Microsoft 'Editor Assembler Plus' software package including uploading services and patches for Disk I/O. Purchase price includes complete copy of Editor Assembler + and documentation for Mod I. Assemble directly into memory, MACRO facility, save all or portions of source to disk, dynamic debug facility (ZBUG), extended editor commands.

DEALER inquiries invited. System Houses - check for special pricing on hard drives and software.

CIRCLE READER REQUEST FOR FREE 24-PAGE CATALOG
*TRS-80 IS A TRADEMARK OF TANDY CORPORATION

CHECK VISA M/C C.O.D. PURCHASE ORDER
TELEPHONE ORDERS ACCEPTED (714) 997-4950

RACET COMPUTES ✓41

1330 N. GLASSELL, SUITE M, ORANGE, CA 92667

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

RELOAD 80

It's official: The disk version of LOAD80 will be available starting with the March 1982 issue of *80 Microcomputing*. The disk will cost $19.97 for single issues or $199.97 for a twelve month subscription. At these prices, you can have those great programs from your favorite mag on disk for just $1 a program!

The disk will be a double-sided, single-density floppy formatted for TRSDOS 2.3 operation (there will *not* be an operating system on the disk). Single disk-drive owners will need a single drive copy utility to use the LOAD80 disk. Model III users will have to use their TRSDOS Convert utility to use the programs. Some of the EDTASM source code files may not run on the Model III without modification. Since double-sided disks are more susceptible to damage and deterioration, we suggest you make a backup of your disk as soon as you receive it.

LOAD80 will continue to be offered on cassette tape, both current and back issues. However, production difficulties make it impossible to offer back issues of LOAD80 on disk. It is an easy matter for the dedicated computerist, however, to put the tape-based LOAD80 program on disk. Many of our readers are building libraries of programs from *80 Micro*—making a subscription worth many times its price!

* * * * *

Magnetic Media

The December 17 issue of *Infoworld* carried a guest editorial calling for industry publications carrying software listings to consider offering the software on magnetic media. The author of the editorial thought readers might respond with enthusiasm. *80 Microcomputing* readers have already proved such a program was needed and have indeed responded to the LOAD80 project with enthusiasm. As the microcomputer industry is learning, *80 Micro* is a leader, giving its readers what they want and need to progress in this rapidly evolving field.

* * * * *

New Manager

With the March issue of *80 Microcomputing*, Art Huston will take over as project manager of LOAD80. Art comes to *80 Micro* from our sister company, Instant Software, where he manned the customer service department and worked as the head of IS data processing. If you have any problems with LOAD80 tapes or disks, contact Art at (603) 924-9471, Ext. 226. In future issues of the magazine, Art will pass along loading hints for tapes and disks and try to keep you informed about LOAD80 matters.

* * * * *

Reminder

Here is a reminder about LOAD80 tapes. Since Model I TRS-80s read cassette data at the rate of 500 baud (bits per second) while the Model III reads tapes at 500 *or* 1500 baud, LOAD80 tapes are recorded at 500 baud so they may be run on both models. Therefore, if you own a Model III, be sure to select the low cassette baud rate before trying to load the tape. To select the low baud rate, under Disk Basic, type: 'POKE 16193,0' from the Basic Ready prompt.■

February LOAD80 Directory

Program	Title	Page	Comments
1	GSETTR	70	None
2	PUNCTUAT	82	None
3	TESTMAKR	104	None
4	PLANETS	112	None
5	CAPCHEM	116	None
6	BOGGLE	166	None
7	GETTEAM	196	None
8	KINGSCAT	232	None
9	SALECAST	268	None
10	SNAPSHOT	272	None

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIV

PRICE BREAKTHROUGH

Super Sale on DISK DRIVES

was ~~$299.95~~ • now only **$199.95** complete!!
40-track, double/single density

for

TRS 80*
Model I and III
(with power supply and case)

ORDER NOW • SUPPLY IS LIMITED

Verbatim Diskettes (box of 10)	**$30.00**
2 Drive Cable	**$25.00**
4 Drive Cable	**$35.00**
Model III 48K - 2 Drives	**$1,799.95**

Please charge to my ___Visa ___MC

Card #__________ Exp. Date _____

Enclosed is a check for $ __________
(Please allow 2 weeks for personal checks.)

☐ Model I Drive 0 1 2 3
☐ Model III Drive 2 3
☐ 2 Drive Cable ☐ 4 Drive Cable

Name __________

Address __________

City __________ State _____ Zip _____

Signature __________

FOB Framingham Mass. residents add 5% sales tax

SOFTWARE SUPPORT

ONE STALKER LANE, FRAMINGHAM, MA 01701 ✓188
(617) 872-9090

Not responsible for typographical errors.

*TANDY CORPORATION

✓See List of Advertisers on page 290

See List of Advertisers on page 290

80 REVIEWS

edited by Michael E. Nadeau

"The CAL field is ready for many critical, analytical and creative minds."

Learning with Computers
Alfred Bork
Digital Press
Bedford, MA
Hardcover, 286 pp.
$25

by Dr. Timothy B. Jay

Alfred Bork is a pioneer in computer assisted learning (CAL). He has made significant contributions to the field through materials and strategies developed for physics classes at University of California-Irvine's Educational Technology Center, his work for the National Science Foundation and by offering courses on computer learning. His opinions, evaluations and projections for the future of CAL cannot be ignored. Thus his book merits serious attention.

Bork has assembled his papers, some co-authored, over the past ten years into seven chapters on learning with computers. The key word here is "with". He views the computer as a tool that educators can use to enhance learning. The reader learns how to use computers to manage classroom records, provide testing application, instruct, remediate, and motivate students. The text is not about a computer's architecture or acquiring a programming language.

The reader realizes current programming languages such as Basic, Fortran or Cobol as well as authoring languages like Pilot or Coursewriter are inadequate because they do not provide adequate graphic capabilities or flexible dialog interaction. Bork prefers the rapidly developing approaches in languages such as Pascal and Ada.

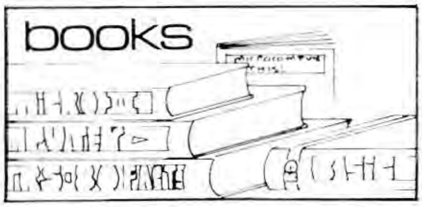
His philosophy of education and instruction is readily apparent. Materials should be prepared by teams of teachers, programmers and designers. Material preparation takes time, effort, money, an understanding of software/hardware capabilities, and good review and evaluation. The approach should include a dialog interaction between student and program while using graphics, personalization, problem solving and immediate feedback to motivate students and enhance education. Review, transportability, distribution and marketing considerations for computer material is also discussed in detail.

The text provides a realistic view of the place of computers in education. Bork's ideas represent a state-of-the-art assessment of all basic philosophy and method for instructors using CAL technology.

Several issues will bother you about this book. The price is steep for what amounts to a collection of reprinted articles. While the material is very readable, interesting and exciting the social scientist or educational material evaluator will find no data supporting claims about the efficacy of graphics, dialogs or the computer learning environment. However the lack of assessment and evaluation is typical of the lag between instruction and critical analysis. There is a critical need for task analyses of learning procedures and statistical tests of the value of CAL materials and procedures. I agree with Bork's projections, but such speculation must eventually be supported by proof.

The physics course examples may be difficult to understand for the non-scientific reader and the redundant examples and demonstrations will bother the cautious reader.

Two years ago I was captured by Bork's innovative and enthusiastic approach to education at his NSF Chautauqua course. I hope educators will enjoy a similar fate by reading *Learning with Computers*. The CAL field is ready for many critical, analytic and creative minds. ■



Basic for Home Computers, $5.95
Basic 2nd Edition, $8.95
Data File Programming in Basic, $9.95
Introduction to 8080/8085
Assembly Language Programming, $8.95
Using CP/M, $8.95

The Wiley Self-Teaching Guide Series
John Wiley & Sons, Inc.
New York, NY
Softcover, 320 + pp.

8080/Z80 Assembly Language—
Techniques for Improved Programming
Alan R. Miller
John Wiley & Sons
New York, NY
Softcover, 318 pp.
$9.95

by Dennis Thurlow

These five books are representative of the Wiley computer series.

Each book covers the fundamentals of its topic, using some standard version for reference. The chapters are logically arranged for ease of learning; knowledge builds on knowledge, quizzes follow each chapter, and each paragraph has fill-the-blank self-check questions following it. The beginning of each chapter outlines what you should learn from it.

These books are good self-teachers, but at the beginning of each, many concepts are covered all at once, and you can get confused unless you are a careful reader. Also, if you do not know how these computers are used daily, you might not grasp the significance of many examples.

Most personal computers today contain Basic because of Bob Albrecht, founder of the People's Computer Society. Using Basic, he proved that anyone can learn to compute. This simple book is a continuation of that effort.

This first book is so well illustrated that you do not need a computer to learn the language. If you use the book as a textbook, one computer for a class of 30 students is plenty. Any computer would do, as this volume teaches a simple version of Microsoft Basic that ought to run on any machine.

Basic Second Edition

This volume reviews the entire command set, and focuses on three things: debugging programs and using Basic to solve real world problems. And, it expands on earlier concepts, such as doubly subscripted arrays. Some commands introduced at this point do not work for all systems, so look at the manual for your machine to find what to substitute.

The real-world applications section encourages lazy business programming. Why teach someone that they must program so that a last-name field must have a certain number of characters when there are many ways around that constraint?

The book does not teach how to use the commands in programming, but a chapter on files at the end answers questions about what Basic words like Open and Close mean.

Data File Programming in Basic

This book deals with data files, the workhorses of business programming. It covers disk and tape files, both random and sequential, and shows you how to write utilities to convert from one set of files to another (the real problem in many business applications).

The book uses Microsoft Basic as the reference and dedicates a separate chapter to North Star files. The book references this chapter whenever references are necessary.

Using CP/M

CP/M is an operating system for 8080, 8085 and Z80 systems that has become so widely used it is considered a standard for business software. Most systems that use CP/M also have a Basic.

CP/M is revised frequently, so there are many versions of it in existence. The book uses the Digital Research Version 2.0 as the reference and notes the differences from earlier versions.

If you use this book as a self-teaching guide make sure you read the introduction and first chapter very carefully. You should have a computer to learn how to use CP/M. CP/M is not a language, but a system that manipulates files, programs and other types of data.

The book does not tell you how files are physically stored on disk. (It should because it differs dramatically from machine to machine.) On the whole the book adequately explains the fundamental use of CP/M.

Introduction to 8080/8085 Assembly Language Programming

This is a standard book that teaches number systems, hex and register layout. The self tests make Assembly-language programming easy for the do-it-yourselfer.

You do not need a computer to learn simple Assembly-language programming, but be aware that you are learning the Intel abbreviations of the instruction set. The book should teach machine code in octal instead of hex. The 8080/8085 series was developed from octal sets and lends itself to octal much more readily than hex, therefore much of the good published software for the 8080 is in octal.

If you already know the 8080 or Z80 and want to learn about the 8085, skip this book. The book briefly notes one difference: two instructions that send and receive an interrupt mask. The appendix shows an assembler implementation on CP/M which is useful to the CP/M user.

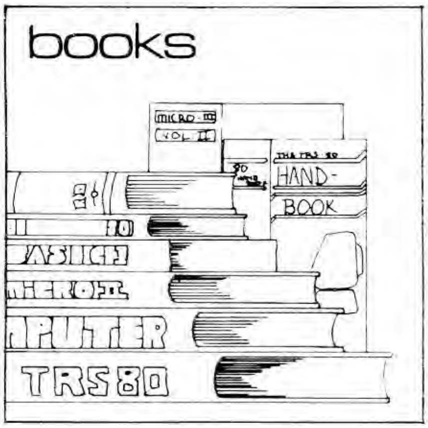
As an approach to learning, the series is good. I did not review other books in the series on Fortran, Cobol and JCL. I would not use the series as my only learning source, however.

I recommend the first Basic book (or its TRS-80 or Atari equivalent) for teaching a beginning class on a low budget. I would definitely use the CP/M book if I wanted to learn CP/M—there is no other simple CP/M book on the market. I recommend Assembly-language programming book only if you have a CP/M-based assembler■

"I recommend the first Basic book . . .
for teaching a beginning
class on a low budget.
I would definitely use
the CP/M book
if I wanted to learn CP/M."

8080/Z80 Assembly Language

This book is not a part of the self-teaching series, but it is the best of the Wiley books that I have seen.



The examples are in Z80 op codes because most assemblers use them. The examples are applicable to the 8080 chip if you limit the instructions to those compatible to both. The appendix gives the 8080 instruction mnemonics and a cross assembler listing. The book discusses the octal base of the 8080 instructions. This makes the instructions easier to remember if you must program without a chart. Use the information in this discussion to save memory when you decode or encode commands in a utility you are writing.

The book briefly explains gates and uses them to represent Boolean logic. The book discusses magnetic and paper tape I/O routines in detail, and it also discusses memory mapped I/O, a topic usually neglected in such books.

The book briefly explains macros once it lays the groundwork of simple assemblers. More could be said about macros, but their inclusion is commendable.

Three sections of the book make it worthy to include in my reference library, even though I have been programming in Assembly language for many years. There is a 50-plus page chapter on how to develop a monitor for your particular 8080 or Z80 system. There is a chapter explaining how to link your own routines to CP/M. An appendix details the undocumented Z80 instructions, how to add them to existing assemblers, and how to use them. These three items alone would be worth the price of the book.

Bravo, Alan Miller!■

Programs for Beginners on the TRS-80
by Fred Blechman
Hayden Book Co., Inc.
Rochelle Park, NJ
Softcover, 150 pp.
$8.95

by Douglas Stewart

I started with Level I and moved to Level II after two months. I do not think anyone will disagree that Level II is needed to write serious programs and that 16K is required to write programs of some sophistication. The best way to get started, though, is with Level I. The original instruction book was written for Level I and is much easier to follow by going through it in the proper version of Basic.

This book fits in after you have gone through the Level I beginners' manual. You can use it before or after conversion. It contains 21 programs all run with 4K of memory and all but two can be used in either Level I or Level II. The programs apply to both Model I and Model III. (Model III conversion information is not included in the book, but the author will provide the instruction when sent an SASE.) No printers, disks or interfaces are required.

Accompanying Discussions

The author admits that the programs are simple. There is no question that they have little practical value regardless of whether your TRS-80 is used for business or pleasure. The value of this book is the accompanying discussion that goes with each program. Each program begins by the author telling you what you will learn. Several subjects are covered by each program. A description of the program is given next. Then a detailed explanation of each line of the program. This is the purpose of the book. It is here where you can learn and get ideas on how to write your own programs.

The listings in the book are reproduced from a dot-matrix printer. This method has advantages because it is usually error-free. I do think that the listings in these types of books would be more readable if solid character printers were used.

There is a very convenient index in the back of the book listing different Basic commands and in what program they are demonstrated.

This is an excellent book for the beginner.■

The Basic Handbook
An Encyclopedia of the
Basic Computer Language
David A. Lien
Compusoft Publishing
San Diego, CA
Softcover, 480 pp.
$19.95

by Alyson Grupp

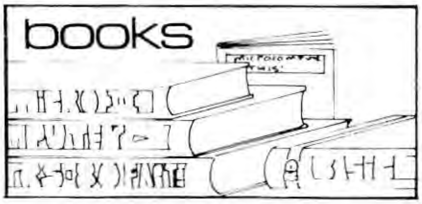
Which version of Basic did you learn first? Is it the same version you use now? What are the differences? How do you find substitutes for the words you can no longer use?

Perhaps you only use one dialect of Basic, but adapt programs written for other systems to work on yours. How on earth do you figure out how to replace those commands that just won't work on your computer?

David Lien's *Basic Handbook*, a standard reference for both the neophyte and experienced Basic programmer, has been an excellent solution to these and other problems for the past three years. Those familiar with the work will be glad to hear it has been thoroughly updated. Programmers who have never used it can begin with a far more complete version.

New features include special sections on dialects of Basic. There are more cross-references and information throughout the text. The vocabulary has been expanded from about 250 words to over 500. Many more computers are included. Ever hear of the Acorn Atom? The ABC-80? Well, they are here... along with 76 other computers.

Extensive information is given on each Basic word. It is easy to flip through the book and quickly find a particular word. Each word is explained clearly and concisely. The *Handbook* also notes which systems support a particular command.



Short test programs provided with each word quickly demonstrate if the word is in your computer's vocabulary. Included are hints on usage, alternate spellings and suggestions about what to do if your computer can not handle a word. Variations in usage are listed to describe how a word can be used differently. Last, but by no means least, are the cross-references. These are invaluable when converting programs from one dialect to another. The book refers you to several words to perform the same function, making it easy to find a usable substitute.

Special sections are provided on four computers: the Acorn Atom, the Atari, the Tektronix, and the TRS-80 Color Computer. These sections note Basic words unique to each vocabulary, special features of each dialect and have liberal cross-references to the main body of the text.

The Index and Scorecard section is a key element of the Handbook. All 500 plus Basic words described are listed, with space for marking the results of each word's test program. There is lots of space here and throughout the book for notes— a nice touch!

The book came off the press just as I was beginning to convert a program written in Level II to run on Extended Color Basic. I do not know how I would have managed without it. Here in one place was everything I needed to convert the program. The special section on the TRS-80 Color Computer helped me make sure I was not missing any Extended Color Basic words to do the job. Cross-references provided the words not supported by Extended Color Basic with a minimum of searching and frustration.

The Basic Handbook is one of those books you will refer to again and again. It is an excellent book. Whatever your programming interests, the *Handbook* will be a useful addition to your library.■

"The Basic Handbook *is one of those books you will refer to again and again."*

✓See List of Advertisers on page 290

UPI-3, UPI-4 Serial Printer Interface
Speedway Electronics
Noblesville, IN
$139.95, Model III
$149.95, Model I

by Jerry W. O'Dell

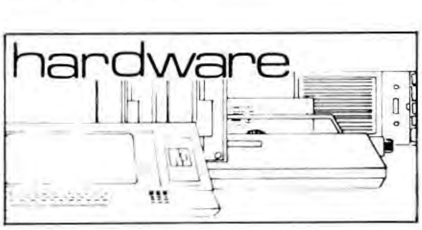
If older teletypes such as the Model 33 Teletype become available cheaply in large numbers, they could be a bonanza for the computer hobbyist, for the Model 33 makes a decent printer.

However, you can't just hook a Model 33 up to your TRS-80. There are two difficulties: Most microcomputers have parallel printer ports. All eight bits of the ASCII code are sent out to the printer at one time. The parallel method is faster, and perhaps more accurate, but it needs at least 10 wires for full handshaking. The teletypes require data in serial form. The ASCII code is sent out one bit at a time over one wire.

Most microcomputers designed for Centronics printers send out a carriage return at the end of a line, but no line feed. In the Model 33, however, carriage return and line feed are separate characters, and must be sent separately. You must somehow insert line feeds if you want proper operation; that takes more logic design than I feel comfortable with.

Previous units which purported to solve the line feed problem used a software driver that remained in the computer's memory at all times. Software drivers take up memory, and slow things down in Basic. You always forget to load them when you most need them. What is needed is a little unit that plugs into the TRS-80 printer port, and makes the TRS-80 think that it has a parallel printer attached.

The Speedway Electronic Serial Printer Interface does just that job. There are two models, the UPI-3, for the TRS-80 with an expansion interface, and the UPI-4 which runs directly from the Model I TRS-80 without the interface.



The unit plugs into my Model III, and it puts out either RS-232 or 20 ma current loop signals. I used the 20 ma current loop which is included in the RS-232 connector. I also used an opto-isolator to electrically separate the two units.

The UPI-3 does exactly what it is advertised to do: It converts the parallel data to serial and puts in a line feed where needed at the end of the line. Mine worked the first time I plugged it in, and has worked ever since. In addition, you can make the UPI-3 put in a delay to give the Model 33 Teletype time to complete the carriage return. Teletypes take a certain amount of time to return to the left margin, and if you are not careful, they will print out characters while they are returning. You can adjust this time delay from very, very long to just right.

In addition, you can select a number of other characteristics of serial data transmission. You can select, with a neat little DIP switch (see Photo 1), handshaking, line feed after carriage return, nulls after carriage return, odd-even parity, number of bits per word, number of stop bits, and parity on or off. The switches in the figure are set up for the Model 33 Teletype. In addition, you can adjust the baud rate with a potentiometer. Mine came set for 110 baud, and I didn't change it, but I'll bet that with a counter you could make it work just fine at 300 baud.

The unit is very nicely built. Photo 2 gives an internal view. The heart of the unit is a GI AY-5013A UART (the big chip in the photo). To implement all the other features, notice that there are 16 other ICs; the UPI-3 is a rather complicated unit. The potentiometer nearest the outside the box is the one to adjust the baud rate; the one on the inside controls the delay during carriage return. Timing in this unit is controlled by a 556 timer; crystal control is preferred, but the unit seems stable without it. Construction seems excellent. The unit is built on a good glass-epoxy board. The parts and workmanship are first-rate.

The UPI-3 performs as advertised and can be depended upon to keep doing it. There are a few minor quibbles: The box is rather cheap (but serviceable), and the cable to the TRS-80 is too short for my taste. But in all other respects it is an excellent unit.■

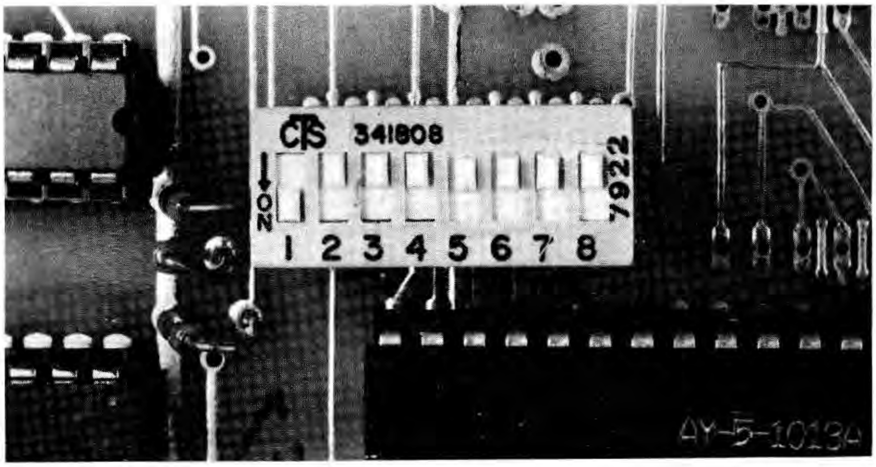


Photo 1

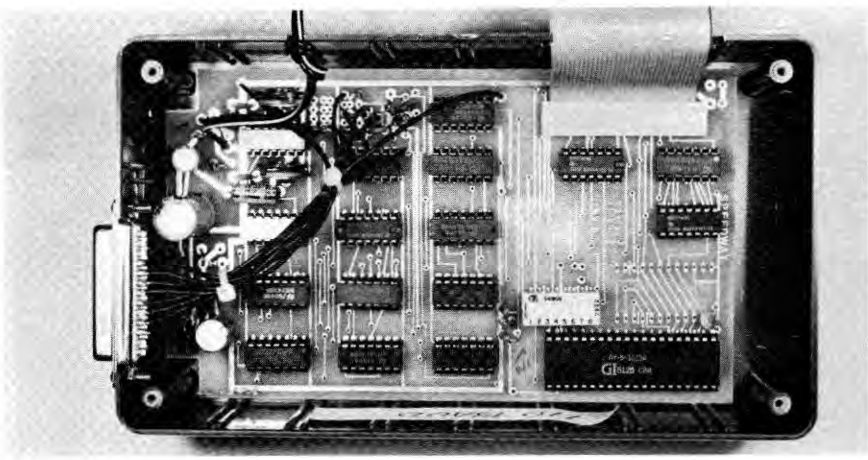


Photo 2

NEWDOS/80 Version 2.0
The Support Keeps Coming.

Apparat's newest disk operating system for the TRS-80®, NEWDOS/80 Version 2.0, has added many new enhancements and features to make your Model I or III computer more powerful. We've kept one thing the same. Our support.
Version 2.0 is our second upgrade of our original NEWDOS for the TRS-80. Each version builds and improves on the capabilities of the preceeding versions. Just as important, Apparat's commitment to supporting our products makes a good product even better. By providing our customers with zaps on an ongoing basis, we're continually making NEWDOS/80 Version 2.0 a more powerful tool.

Version 2.0...
High Performance DOS

NEWDOS/80 Version 2.0 builds even more performance into NEWDOS/80. The versatility and sophistication of Version 2.0 includes features like:

- Double density support on the Model I
- Enhanced compatability between Model I and III
- Triples directory size
- Dynamically merge in basic (also allows merging of non ASC II format files)
- Selective variable clearing
- Can display basic listings page by page
- Automatic repeat function key
- Routing for peripheral handling
- Enhanced disassembler
- Command chaining
- Superzap to scan files
- Fast sort function in basic

These new features, added to the existing capabilities of NEWDOS/80, makes it one of the most powerful additions you can make to your system. And Apparat's commitment to support assures that you've purchased a superior product, both today and tomorrow. At just $149.00 it could be the best investment you will make for your TRS-80.

If you're thinking about upgrading your system, call Apparat today. Dealer inquiries welcome.

303-741-1778

VISA

Apparat, Inc.

4401 So. Tamarac Parkway, Denver, CO 80237 (303) 741-1778

"ON GOING SUPPORT FOR MICROCOMPUTERS"

AD-III Audio Amplifier
Thomas Engineering/TCS Systems
Lake City, FL
$19.95

by Fred Blechman K6UGT

I used T-Beep to provide sound with my Model I. However, when I hooked up to my new Model III, T-Beep locked on after CSAVE, requiring a keyboard entry of OUT255,0 to turn it off—a real bother. Then I saw a news release offering an internal audio amplifier for the Model III.

The price included the amplifier board, speaker, all wiring, instructions and a 90-day replacement warranty. The TCS Systems Model AD-III one-watt audio amplifier claims to have high quality sound, and is designed with a switching system for deactivating the amplifier during data tape transfer. Installation was described as "six quick connections to the CPU board, mounting the amplifier and speaker and you are up and running." I was not thrilled about wiring to the CPU board, however.

The AD-III arrived completely assembled and prewired. It even had tinned ends on the six wires for soldering to the CPU board! The step-by-step installation instructions included several perspective-view sheet drawings of the Model III disassembly, including screw locations. A photocopy of the CPU board shows exactly where to solder each color-coded wires.

The assembled amplifier board includes an LM386 integrated circuit amplifier, five-volt relay, volume control and four capacitors. The two-inch diameter squareframe speaker is already wired to the amplifier board and the six wires requiring connection to the CPU board are harnessed for easy handling—a first-class job.

Once the Model III case top, containing the CRT and video electronics, is removed and disconnected, mount the AD-III amplifier board and speaker to the lower grille using the existing holes. The screws, nuts and spacers are provided. Solder the six connections to the component side of the CPU board, reconnect the video and test. CLOAD and program with sound and Run. Then adjust the amplifier volume control. If you do not have a program with sound, use the test program listing, written in Basic, provided with the AD-III. It uses PEEKs and POKEs for machine-language execution speed and produces tones over several octaves.

To perform a test not in the documentation, enter FOR X = 1 TO 5:OUT255,1:OUT 255,0:NEXT.

When you press Enter you should hear a short buzz. For a longer buzz change the 5 to a larger number. You can use this one-liner anywhere in a Basic program by giving it a line number wherever you want a buzz. You are, however, stuck with this one sound unless you use additional programming. To get varying tones, or to play music, refer to the book *Advanced TRS-80 Level II Basic* (Radio Shack Cat. #62-2072 $9.95).

I am delighted with my AD-III. The price was right, the relay disables the amplifier during CSAVE, and installation was smooth and fast. Installation of the AD-III voids your Radio Shack warranty, so wait until your 90 days is up.■

C. Itoh Daisy Wheel Printer
C. Itoh Electronics, Inc.
New York City, NY
$1,895

by Patrick Morgan

The C. Itoh company appears to be the parent company marketing a line of daisy-wheel printers under at least four brands: C. Itoh, Starwriter, Tokyo Electric Company (TEC) and Vista. Apparently all four are identical. They are available in either 25 or 45 characters-per-second versions. Both speeds are available in either a RS-232 serial port or a Centronics-type parallel port. The carriage length is 15 inches with a tractor feed available for an additional $300.

Print quality is excellent. It uses Diablo-compatible 96-character print wheels which are readily available and inexpensive ($6–$8 each). Wheels are available in many fonts of both elite and pica sizes. Ribbons are in a cartridge which moves along with the carriage. The ribbons come in either regular fabric or carbon film for a sharp impression.

The front panel has switches for page set, page advance and off line. It has LEDs to indicate power, on line and paper out. Behind the back cover are 10 DIP switches to set form length (for top-of-form control) and serial or line mode. A toggle switch sets the power on default values of 10 or 12 characters per inch. Other DIP switches can prevent line feeds or any carriage movement or turn off the print hammer. A self-test feature continuously prints out the entire character set.

"The manual. . . is typical of Japanese translations."

There are two modes of operation: Line and Serial mode. The Line mode types bidirectionally and prints only on receipt of a carriage return (ODH). No special features are available in this mode. The Serial mode prints each character as it is received. Special features are possible by sending a ESC code (1BH) followed by one or more characters. Most common is underlining, superscripts and subscripts. The letter spacing can be set at 1–120 characters per inch. Vertical line spacing ranges between 1/48 inch to one inch. It can jump to tabs in either forward or reverse.

The carriage can be moved under software control anywhere on the paper with a resolution of 1/120 by 1/48 inch. With the right software a 10-inch high graph of several hundred points could be made in one pass from left to right.

I did notice a few problems. The manual is not too bad but is typical of Japanese translations. It is necessary to re-read some sections several times to understand. However, the necessary information is there.

The platten is 15 inches long and will print a line until receiving a carriage return (ODH). There is no way to make a line wrap around. If you are using 8-1/2-inch paper and LLIST a Basic program, part of your data will print on the rubber platten. Another DIP switch to set line length would be helpful.

There is only a nine-byte buffer in the Serial mode and only a one-line buffer in the Line mode. Admittedly this is all that is needed but I would rather see a 1K or 2K buffer so the CPU could dump a few lines and go on without waiting for the printer.

My last complaint may seem trivial but it is bothersome. The carriage return is slow. It takes a full second to return 15 inches. This is only noticeable in the Serial mode. Overall, this is an excellent printer.■

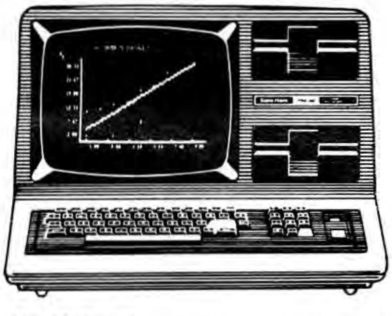
AMERICAN Small Business COMPUTERS

118 SO. MILL ST
PRYOR, OK 74361
(918) 825-4844

ALL LISTED PRICES ARE CASH DISCOUNTED. WE ACCEPT MC, VISA, AND COD (REQUIRES CERTIFIED CHECK, CASHIER'S CHECK OR CASH). FOR CHARGE ORDERS, PLEASE ADD 3%. FOR COD SERVICE, PLEASE ADD 3%. OKLAHOMA RESIDENTS PLEASE ADD 4% SALES TAX.

DON'T PAY MORE!!

TRS-80 MODEL III



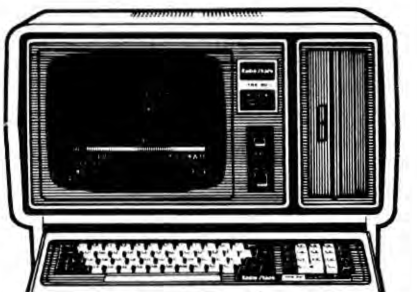
TRS-80 16K Model III..**$849.00**
TRS-80 48 K Model III..**$899.00**
TRS-80 48 K Model III
with one disk drive..**$1499.95**
TRS-80 48K Model III
with two drives..**$1895.00**

*Model III is also available with dual head and 80 track drives. Drives listed give same performance as standard Radio Shack units.

THERE'S NO SECRET

There's no secret as to why we can sell TRS-80 computer equipment for less. Low overhead — that's the reason — pure, simple and straight! As everyone knows, Radio Shack is a huge corporation...with huge expenses. We are able to buy the barebones computers from Radio Shack. To these computers, we add our own memory chips, disk drives, cables, etc. The result is a computer system which is equal to, or in many cases, superior to the one you would get from Radio Shack in both price and performance capabilities. There is one thing that you don't get from Radio Shack — their 90 day warranty. What you get in its place is the exclusive 120 days American Small Business Computer's Warranty. So...if you like the Radio Shack computer, but don't like the price...**CHECK US OUT!!!**

TRS-80 MODEL II



64K MODEL II WITH ONE DISK DRIVE

$3295.00

AW...WHAT THE HECK

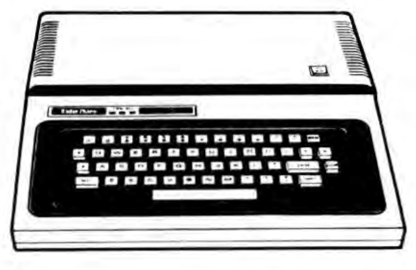
RAM Memory Chips for the TRS-80 *Apple* and others

It is the policy of American Business Computers to offer merchandise at the lowest price possible. Several 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.

$18.95 PER 16K SET

These chips are brand new "4116's". These 200 nanosecond chips are fully compatible with all TRS-80 products. 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.

COLOR COMPUTER



COLOR COMPUTER 4K LEVEL I......**$369.95**
COLOR COMPUTER 16K
EXTENDED BASIC......................**$499.95**
COLOR COMPUTER DISK
DRIVE 1....................................**$499.95**
COLOR COMPUTER DISK DRIVE
2, 3 or 4....................................**$329.95**
COLOR COMPUTER JOYSTICKS.......**$24.95**

* DEALER INQUIRIES INVITED

TRS-80 MODEL III SOFTWARE

DISCOUNT PRICES ON Model III Software

Profile	(26-1562)	**$69.95**
Profile III Plus	(26-1592)	**$175.00**
VisiCalc	(26-1566)	**$87.95**
VisiCalc III	(26-1569)	**$175.00**
Scripsit	(26-1563)	**$87.95**
Super Scripsit	(26-1590)	**$175.00**
Scripsit Dictionary	(26-1591)	**$131.00**

CC DISK DRIVE

COLOR COMPUTER DISK DRIVE UNIT

$499.95

CALL 918-825-4844

* TM Tandy Corp.

✓529

Aircraft Instrument Approach Simulator
J.C. Sprott
Madison, WI
Models I & III, 6K
$9.95 cassette

by Buzz Gorsky

As an instrument-rated commercial pilot and instrument flight instructor I can never resist looking at games and simulations involving flight. This program falls into the latter category; there is no way to look at this as a game. An individual unfamiliar with instrument flying will not know how to make the simulator work and will probably crash often.

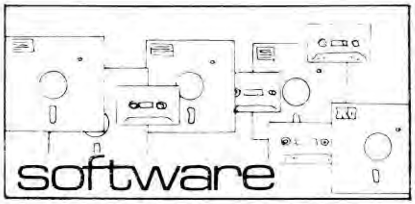
The program is written in a straightforward manner; liberal use of spaces in lines and the absence of long lines make the program easy to follow. The program comes with a description of aviation terms as well as a diagram of a typical approach and an explanation of the screen display.

In spite of these materials neophytes may find the screen somewhat mystifying. The description neglects the artificial horizon, but after "flying" the thing for a while that becomes obvious. Non-pilots may be confused by the localizer function since the simple statements about flying to or away from the needle in front course versus back course situations will not make much sense.

However, this program is lots of fun, somewhat of a challenge and a learning experience for a student or pilot who is familiar with instrument flying and wants some simulator practice. In spite of my extensive flying experience, I had one crash and one rough landing before getting an ILS approach right down the groove.

The program has some drawbacks. The pilot/user enters numbers from the keypad to direct the aircraft. Unfortunately these are often combined (such as climb right or descend left) and it is sometimes unclear how to get out of a turn while climbing or descending and still maintain the climb or descent. It is artificial to have to remember the numbers to keep the commands straight. The four arrows should have been used for nose up/down/-left/right and a small range of numbers should indicate power changes. Power can only be increased or decreased very slowly. This too is artificial.

The user specifies the field elevation and the landing heading as well as wind direction, speed and turbulence. I like programs that select all these things for me and indicate the field elevation on the screen during the approach. This is more realistic since even in a situation where one knows the wind characteristics on the ground (from weather information) and at



altitude (from cruising experience), one must always "feel out" wind conditions during an approach and experiment until the approach is set up. Knowing the exact wind throughout the approach takes away a good bit of the challenge.

Considering the price of the program and the amount of fun someone interested in flying is likely to have with the program, I recommend it. It works and so will you! ■

Addendum: Since this review was written, Mr. Sprott has incorporated some of the author's suggestions.

Paddle Pinball
Eric E. Quintana
Tandy/Radio Shack
Fort Worth, TX
$9.95 cassette, Models I & III

by Charles Gulick

Radio Shack scores high with this excellent game, the best feature of which is the option to change the playfield. In fact, you can create and save dozens of different layouts and load any of them once Paddle Pinball is aboard. The game also features lively sound effects, which operate on games of your own design as well as the game provided. The program is in machine language.

The programmed playfield is deliberately sparse, concentrated to the left and right to leave plenty of room for designing your own central area.

The play combines pinball-style action with a paddle which you guide across the bottom using the left/right arrows. You control the direction of the ball's rebound by bouncing it off different segments of the paddle. The shift key speeds paddle movement—in fact, it will often cause you to overshoot until you have had some practice.

When you have System-loaded the game you can enter up to three players. The field then appears and waits for your Enter to bring the first of three balls into play. The object is to amass points, of course, but also—if there is more than one player—to score credits. Each credit entitles you to an extra game. There is also a Music block, which starts with the number two in it. The number augments each time you hit a corner of the box, and you are rewarded with a tune consisting of that many notes.

Two short vertical bars at the top of the playfield are dubbed The Gap. When your ball goes into this gap, you get 3,000 points, and you light one letter of the word "music" at the bottom of the screen. When you have spelled the whole word, hitting the gap again will give you 26,000 points and an extra game. There are other scoring features including a chute which increases in value if you play well. The ball usually, but not always, goes out of play when it hits the bottom of the field.

To modify the playfield, you press the up arrow. This freezes the action. You may then change the position of each of the three bumpers using all four keyboard arrows. Next you can create bonus dots, placing them at will. For each bonus dot you are awarded 1,000 points, which are credited when the ball drops to the bottom of the screen. In the same manner, you can create random dots, which may or may not score and which add to the interest and action. Finally, you can move The Gap left or right. When the field is as you like it hit "A" to resume play.

You can save your newly designed playfield on cassette with a single command, reloading it whenever the game is aboard.

Taken altogether, Paddle Pinball is a clever and creative program, well worth its modest price if you like arcade-type games. It loads like programs should load. My copy (both sides) spun in readily at my standard setting (7.5 on an old CTR-41). My only complaint is that the tape is the same for both computers, and thus loads on Model III at the same weary 500 baud as on Model I. When you get used to 1,500 baud, that slowly flashing asterisk can be an irritant. I hope Tandy will soon provide cassette loads which take advantage of the faster capabilities of the Model III. ■

Starfighter
Adventure International
Longwood, Florida
Models I & III
$24.95 16K cassette
$29.95 32K disk

By R.J. Brown

Starfighter contains a 32-page *New Pilot Introduction Manual* (stamped Top Secret), published by the Solar Galactic Authority, Periodical Office/Landbase Central, and two cassettes.

The Trainee Simulator tape allows a novice pilot to explore the operating parameters of the SC-78503 Starfighter craft and to learn about the universe without the hazards of actual combat. The simulation stops if the pilot makes an error and the correct procedure is indicated.

Make no mistake about the Main Mission tape though, it is the real thing. The pilot departs Landbase Central as a paramilitary mercenary combatant, after being given a SC-78503 Starfigther craft free of charge by the Solar Galactic Authority. As in the past, there is no free lunch since the pilot is now responsible for proper maintenance, fueling and recharging of the SC-78503. These services are paid for by the pilot at four of the eight Landbases, where the work is done. A positive bank balance is required for this and is obtained by destroying certain designated hostile craft for cash bounties.

Two of the Landbases transfer the combat record to the pilot's personal bounty log, though it is possible that the pilot's arrival at Landbase may precede the combat record. Payment for services rendered are drawn against this log.

Hyperspace, Hypercharge, and Stellar Pumping

The theories of Hyperspace, Hypercharge and Steller Pumping are classified and cannot be related here, but are thoroughly covered in the *Introduction Manual*. Briefly, the pilot uses Hypercharge to get into Hyperspace to travel from now to elsewhere and elsewhen.

While in Hyperspace, as the pilot watches the stars slip by, a gravity field sensing system enables the SC-78503 to detect possible targets in normal space. After exiting Hyperspace, the controls are manually changed to combat mode and a wave or beam weapon selected, each having certain distance and spread characteristics.

Every SC-78503 is equipped with a beacon, similar in function to antiquated twentieth-century aircraft transponders, for identification purposes. This beacon may be turned on or off at pilot discretion. The pilot may request a beacon from any unknown object and may be requested to transmit a beacon by the same object.

The Solar Galactic Authority is currently in a state of war with the Petro Research Conglomerate. The PRC deploys certain hostile spacecraft, which must be destroyed to maintain the integrity of the SGA. A large number of SC-78503 Starfighter craft types are engaged in marauder activities along with Space Pirate craft. These crafts are to be considered valid targets by the pilot.

There are at least 12 major craft types operating, including intersellar mines, SGA Ball-turret Gunships and the dreaded PRC Deathcaster. The pilot must seek out, identify, and destroy all craft hostile to the SGA. Indentification is achieved by careful observation of visual profile or characteristic craft movement and evasive tactics. Performance data of all known craft is detailed in the *Introduction Manual*.

The SC-78503 Starfighter provides the standard real-time visual screen, a combat computer with target ranging and direction displays, targeting grid and target lock controls. Included on the companel are a craft velocity indicator, speed and direction controls and a digital beacon message readout. A unique feature of the targeting grid gives the pilot rear, as well as forward views of space, quite helpful for detecting hostile craft approaching from the rear.

Combat Efficiency

Pilot combat efficiency is evaluated and rated by Landbase Central and rewarded by monetary awards and rank promotion. If pilot fatigue develops during a mission, the pilot may return to Landbase, record the current mission on tape, deactivate the SC-78503 craft and turn off the computer console. After a suitable rest period has elapsed and nourishment requirements met, the pilot may return to reactivate the SC-78503, load the personal tape and continue the demise of the evil Petro Research Conglomerate.

The *Introduction Manual* contains much valuable information to ensure successful missions. Of particular interest to the novice are the sections on recommended operational procedures and target introduction.

Once a pilot reaches the rank of Star Lord, a secret code word will appear on the companel. Sending the code word to Adventure International will bring a special gift by return mail.

I do not care much for wandering about strange castles or dungeons, arcade games bore me. Star Trek is nice occasionally, but Starfighter is a challenge and does not get boring. This simulation seems realistic and the real-time action does not leave much time for even a sip of coffee. ■

Arcade 80
Jerry Yamaguchi
Datasoft, Inc.
Sepulveda, CA
Model I, Level II, 16K
$29.95 disk
$19.95 cassette

by Joel Benjamin

Arcade 80 is a collection of three game programs.

Astro Mines captured my interest most. You have a fixed amount of fuel and time (about one and a half minutes) to blow up as many mines as possible. This feat requires quick reflexes, a sense of strategic planning, and a high frustration tolerance level. It is difficult enough to tax even a practiced hand.

Falling Bricks is similar to the popular arcade game Invaders except that it is your irrevocable destiny to be beaned by a brick. The most you can hope to succeed in is to top your previous score. Although there is some element of strategy necessary, your predominant activity is to relentlessly and franticly pound the space bar which launches your missiles against those bricks.

Star Run does not work well or it does not work at all—it is hard to tell the difference. The graphics consist of moving Hs for stars and a screen that whites out when you have been hit. The perpendicular sights at the center of the screen did not seem to move in response to the controls nor did the movement of the stars seem to change. However, even if the controls could be made to work, this game would not capture my interest for more than a few milliseconds.

Astro Mines is challenging and interesting; Falling Bricks is challenging; and Star Run is hopeless. A speed option, a running high-score display, and more imaginative graphics would improve the first two. Children from ages five to twelve would probably enjoy playing two of these three programs, but if you have high demands when it comes to computer games think twice before purchasing Arcade 80. ■

See List of Advertisers on page 290

Asptch 3.2
Byte Miser Software
Rocky Mount, NC
$17, Models I & III

by Dan Zuckerman

Byte Miser software has developed a patch to Editor/Assembler that assembles programs directly to memory. It is particularly useful for writing machine-language subroutines to be called from Basic.

Asptch 3.2 loads on top of Editor/Assembler. The tape replaces over 300 different areas, while stealing less than 400 bytes from the text buffer. Instructions and a utility are provided to make a combined tape of Asptch and Editor/Assembler.

When Asptch is first executed, memory can be protected from the text buffer (just like Basic). This high memory is used for the machine-language program. Asptch already has keyboard debounce. This debounce routine can also be used from Basic.

Extra commands are allowed by replacing the Illegal Command error of the Editor/Assembler. When Asptch gets an unrecognized command (X, for instance), it clears the screen. Then it reminds you of reserved memory, the reentry point (non-destructive) for Asptch and Editor/Assembler, whether the program will be assembled to memory or tape, and how much room is left in the text buffer. A list of the extended commands is also given.

The Asptch memory size command (A) redefines protected memory. This is not a destructive command. It leaves the text buffer intact, unlike Basic. This is useful when you find that your program uses more memory than expected.

You can test your machine-language program. The Execute command jumps to the starting address of your program just like Run in Basic. Asptch can regain control in several ways. A return instruction without a corresponding call will return to Asptch. The machine-language program can jump to an Asptch or Basic reentry point. The reset button will put you in Basic. Once in Basic, you can use the System command to reenter Asptch. As long as your program does not disturb Asptch or its text buffer, you can regain control without losing a byte.

The Dump command switches the dump to memory/dump to tape flag. This determines whether the assembler will record tape or record in memory. A dump to memory still results in a Cassette Ready prompt. Answering this puts the object code in memory. At the same time, a few random tweeks come out the cassette port (nothing to worry about).

The calculator facility of Basic can be used with the Ready command. The System command jumps right to the System mode to load machine-language tapes or jump to a given address.

A unique and powerful feature of Asptch is the memory size command. This lets you enter Basic with Asptch, your Assembly-language source code, and your machine-language object code all protected in high memory. Using this feature machine-language subroutines can be developed for use with Basic. Only the Basic program has to be reloaded, not the assembler and its program.

The Convert, Display and Modify command puts Asptch in a monitor mode. You can display memory addresses and their contents in hex, decimal, and the printable character codes (ASCII). Using the up and down arrows, you can scroll through memory. Subcommands can modify locations and enter the Basic calculator mode.

Don't confuse this monitor with T-Bug, Z-Bug, or anything similar. It has a provision for breakpoints or looking at registers' contents. This monitor is not designed for heavy debugging. The author assumes you would rather modify your source code and reassemble than play with your object code.

An alternative technique would be to write your own Assembly-language breakpoint routine. While debugging, your program could periodically call the routine. It would display registers and wait for some action on your part. The Asptch documentation tells about its hex and decimal conversion routines. It would not be difficult for an adept Assembly-language programmer to simulate a monitor with breakpoints utilizing these routines.

Asptch adds a few more features to Editor/Assembler. After each source dump it provides a verify option. This gives you the opportunity to rewind the tape and let the computer compare the file to memory just like CLOAD? in Basic.

The Break key will edit from cassette load hang-ups. It is no longer necessary to use the reset button. Asptch is fully compatible with Radio Shack's lowercase modification. The documentation tells you how to disable Asptch's keyboard debounce so that you may use your own.

It should be noted that Asptch will not work with Editor/Assembler version 1.1. You can check which version you have by loading it and looking at the opening screen.

Competition

Asptch's major competition (on the Model I) is Microsoft's Editor/Assembler Plus. Asptch's advantages are lower cost (assuming you already own Editor/Assembler 1.2), source tape verify, Basic reentry with Asptch, an Assembly-language program saved in protected memory, and a larger text buffer. ■

KWICOS
KWIK Software
Bolivar, MO
Model I, Level II 16K
$24

by Fred Blechman

KWICOS is a cassette operating system that converts your 500-baud Model I to a high-speed system for loading and saving Basic or System programs.

KWICOS is a System program using about 1,500 bytes of low user RAM (memory locations 17129 and up). It allows you to use your regular high-memory utilities (lowercase drivers, printer drivers, sound drivers, renumber programs) as usual. KWICOS provides disk-like loading and saving commands in addition to your regular Basic commands. Keyboard debounce, operating speed and the speed of loading and saving programs are controlled with KWICOS commands. KWICOS features include program loading and saving up to 4½ times faster than Level II, with unique screen graphic symbols to indicate lock-on and transfer status; an active Break key during program transfer; a Merge command allowing you to append Basic programs; program identification with a filename and comments of up to 32 characters; optional password protection; verification of any file saved in KWICOS format; display of Basic program length; display of start, end and entry addresses for System programs; display of filename and parameters of all KWICOS programs on a tape; controllable keyboard debounce; and slowed program execution and scrolling of listing or tabulations for any program in memory.

For all its versatility, KWICOS is simple to load and use. KWICOS does not need reserved memory space. Type System and press Enter, then type KWICOS and press Enter. Loading time is about 40 seconds. A good load is signified by a screen prompt: Key Data-Rate (2-3-4-5-6). If you have the Radio Shack XRX cassette input modification, the KWICOS documentation explains how to defeat the modification. With XRX active you cannot use the 3, 4, 5 or 6 KWICOS data rates.

The data-rate numbers indicate the relative speed of program transfer compared to standard Level II 500 baud. Since program leader lengths run several seconds even on short programs, only longer programs approach these multiples in total transfer time. For example, I found data-rate 6 to be about 280 characters per second, or about 4½ times faster than the normal Level II program transfer rate (62 characters per second). This is equivalent to 2,250 baud. The significance is in long programs, where a 15K program takes about four minutes to load in Level II format, but only 54 seconds in KWICOS at data-rate 6.

When you enter the desired data-rate, the program returns to Basic with a standard Ready prompt. Use CLOAD and System normally (except for low-memory System programs, which overlay and destroy KWICOS). The program entry pointers are set by KWICOS to load a Basic program above KWICOS. KWICOS uses about 1,500 bytes, so programs that do not leave that much memory when run cannot be used with KWICOS.

Using KWICOS for loading and saving programs is easy. After KWICOS is in position, load a Basic program. Type Save "Filename" and press Enter. The recorder will start and you will see a graphic rectangle near the upper right corner of your screen. In a few moments, as the program leader is generated, this rectangle alternates with another shape. This is replaced with a checkerboard pattern as the program tranfers from computer to tape. The screen now shows <B> followed by the filename and the total number of program bytes. The <B> indicates a Basic program has been saved. The familiar Ready prompt follows as KWICOS returns you to Level II Basic.

To verify the tape, rewind it and type LOAD? "filename." The tape starts and a graphic bar appears at the upper right side of the screen. As the program leader is captured, the bar appears to vibrate. The bar changes to the alternating checkerboard pattern as each byte on the tape is compared with the program in memory.

LOAD? without a filename will verify the next KWICOS-format Basic program on a cassette with the program in memory. The command Load loads the next KWICOS-format Basic program on the tape into computer memory—like CLOAD, but much faster, and with the graphic symbols showing synchronization and reading of the tape. You can also Load"Filename" to load a specific program from a cassette.

You can add another KWICOS-format Basic program to the one in memory with Merge (for the next program on tape) or Merge"Filename" (for a specific program) in memory; KWICOS appends the second program to the first by moving the appropriate program pointers with no regard to line numbers.

Load 500-baud System programs normally, but to save them in KWICOS format you need the starting, ending and entry (or transfer) addresses in hexadecimal notation. Unless you can identify these addresses in hexadecimal, you won't be able to use KWICOS to save and load System tapes! This may not be a disadvantage: System programs are usually short, so loading time is not much of a consideration.

Enclosing any part of the filename in asterisks provides password protection. The password must be specified to load the program; the password will show only a series of asterisks on the screen—keep a record of your passwords! Insert comments, such as the recorder digital counter number, after a filename by preceding the comment with an equal sign. Filename, password and comment must not exceed 32 characters.

Computer execution speed is controlled by a combination of a POKE and LSET commands. POKE 16467,1 provides a very slight slowdown and minimal keyboard debounce. POKE 16467,255 results in very slow keyboard and program execution. A program listing scrolls only one program line per second! This can be very valuable during analysis of graphics or with the TRON (trace on) command to see exactly how the program operates. Any number from 1 (fast) to 255 (slow) can be used with this POKE, followed by LSET. The RSET command returns computer speed to normal until the next LSET is encountered (in the program or from the keyboard).

Four KWICOS error codes identify user syntax error, checksum error on reading tape, unmatched byte on verification, or insufficient memory for Load or Merge.

I found KWICOS easy to use, with no bugs. The user must identify KWICOS cassette programs with the data-rate number; the data-rate selected on program initialization must match the KWICOS recording, and the data-rate cannot be changed without reloading KWICOS. The preliminary documentation I received with KWICOS was well written, but left out some essential details. This is being corrected, and some more program features are being added.

If you do not have a high-speed tape device and do not expect to get disk drives soon, KWICOS will save you time and frustration, and provide keyboard debounce and slowdown for program development and debugging.■

Temple of Apshai
Automated Simulations
Mountain View, CA
$24.95 cassette
$29.95 disk

by John Warren

Like its brothers in the noted Donjonquest series, Temple of Apshai is a computerized off-shoot of fantasy war-gaming. Dungeon and Dragon players will feel at home as they bargain with the Innkeeper for supplies and set forth into the four-level abandoned temple.

The game consists of two Basic programs, Innkeeper and Dunjonmaster, and four data bases which represent levels within the temple. The cassette documentation contains details on transferring the data bases to disk.

Innkeeper designs a character and equips him or her for the upcoming battles. All characters have six attributes: intelligence, intuition, ego, strength, constitution and dexterity. The values assigned to each govern a player's fighting effectiveness. For example, a fighter with a high value for dexterity but a low one for strength could hit opponents more often but would do less damage with each blow.

"The Innkeeper's invective is energetic and insulting."

Innkeeper also provides a number of silver pieces for purchase of equipment for the adventure. The program then states a price list and haggles with the player until a mutually agreeable price is reached. The Innkeeper's invective is energetic and insulting.

When the adventurer leaves the inn and enters the temple, Innkeeper loads Dunjonmaster which reads the appropriate data base and displays the first segment on the screen. Even when the programs and data have been loaded onto disk, this is a lengthy process. However, it provides an opportunity for the player to raid the icebox and visit the restroom—there will be no further rest until the game is over!

Dunjonmaster works in real time; the monsters do not wait for the player's response before striking. Once I was seized with a coughing fit while my character was alone in a chamber. By the time I had recovered, he was dead—a victim of a giant mosquito. Fortunately, players can select the monsters' response speed; this is particularly valuable for beginners who may have difficulty remembering the direction codes.

A series of one-letter codes directs the computer. Most are self-explanatory—A for attack, L for turn left, O for open door and so forth. The character's status appears on the right margin of the screen. Health and fatigue level are continually updated as well as how much is carried and the number of arrows and magic arrows available. The nature of the attacking monster is displayed under the player's status. This is relatively important since different strategems are more often successful with particular monsters.

The graphics are unexciting, but acceptable. All the monsters are identical—sort of a graphic cross—while the character is a graphic "V" with the point indicating the direction of travel. The initial drawing of a room when the character enters is painfully slow. Thirty to forty seconds may not seem like much, but when you are on the track of fame and fortune, it is an eternity. Perhaps a machine-language routine called through a USR routine might be more effective.

At a time when most documentation gives the impression of having been written by an illiterate chimp, the Temple of Apshai manual is a welcome change. It is almost as much fun to read as the game is to play.

Aside from detailed playing instructions, the manual contains two short stories that give background for the adventure and describe the treasures and monsters that abound within the walls. This attention to detail is also shown in the appendices that describe the more than 200 rooms within the temple.

However, these particular descriptions are of little use while immersed in the game. Anyone who attempts to catch up on reading while playing will eventually look up to find "Thou art slain" on the screen. Eternal vigilance is the price of survival.

A major drawback to Temple of Apshai is that the cassette-based program cannot remember a character. Players are required to write down the characteristics, weapons, armor and treasure so they can describe their character to the Innkeeper. NEWDOS Plus users can print this information with the screen printer option, but others might as well begin an extensive notebook. The most effective technique seems to be to have a partner keep track of treasures and magic healing potions as well as read the descriptions and make a much-needed map. If you lack such help, shift@ freezes action.

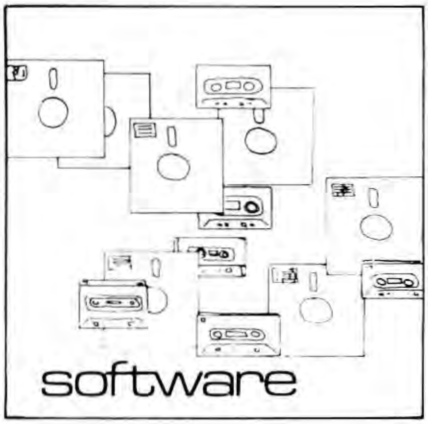
Temple of Apshai brings the excitement of role-playing games to the world of home computers. Buckle on your sword, don your armor and type Run.■

Crush, Crumble and Chomp
Epyx
Automated Simulations
Mountain View, CA
$29.95

by Bert Latamore

"If (your monster's) belly is full and you continue to feed it, it will give you the message 'burp'."

I do not expect much in computer games. Compared to the strategy board and fantasy role games I am familiar with, most of them are very simplistic. Some, like MED Systems' Labyrinth, have very complex mazes which can be interesting. Several companies have put out some good arcade games.



The problem with these games is there is no strategy.

Therefore, I have been particularly pleased with the new Epyx entry, Crush, Crumble and Chomp. In several ways, it is far ahead of the other games I have tried.

A Change in Direction

This game is a major change of direction for Epyx, whose other games, like Temple of Apshai, are attempts at putting fantasy role games on a computer. This is a slightly tongue-in-cheek take-off on the grade B monster movies in which you get to play one of the famous movie monsters —Godzilla, Mechismo, the Glob, etc., and attack one of four major cities: New York, San Francisco, Washington, or Tokyo.

Crush, Crumble and Chomp is a carefully designed game. The graphics on the TRS-80 version are as good as or better than anything I have seen before on the machine.

The documentation is also well done. The booklet is explicit and arranged for easy reference. It includes maps of the four cities which are very handy when you are playing the game. Commands for the various monsters are on separate, unbound cards for easy use.

The writing goes beyond easy use. It provides a gentle humor, just enough to keep you entertained while wading through the rules without making a farce of the game before you play it. That is a hard balance to achieve, but Epyx has succeeded.

Snacking on Human Units

That humor is carried through into the game itself. Your monster has to eat, as the instructions point out, so one of the things you are constantly trying to do is catch some of the "human units" for snacks. When your monster does get something to eat, the display flashes

"chomp, chomp, chomp." If its belly is full and you continue to feed it, it will give you the message "burp."

The real victory of Crush, Crumble and Chomp—the thing that makes it different from the competition—is that it is a true strategy game. You must make decisions from the very start of the game, indeed, from before the start of the game, and your success will in large part be determined by those decisions. You can play one game one way and the next so completely different it may not seem like the same game at all.

Your decisions start when you set the parameters of the game. You decide what monster you want. Each has its own capabilities and weaknesses; each demands, therefore, a different strategy. You do not fight with Mantra in anything like the same way you do with Mechismo. To give the player full control, the disk version of the game allows him to design his own monster if he prefers.

Once you have decided on a monster, you pick the city you will play with. Actually, to an extent, the two choices are interrelated. This is especially true with Kraken who cannot move onto land and who therefore, is best in San Francisco and Tokyo where the board map includes large amounts of water.

Your third decision determines how your performance will be scored. You predetermine your primary mission—to destroy buildings, eat, kill combat units, or survive.

Minor Nuisance to Major Catastrophe

You can make these decisions randomly, of course, and your first few times through you probably will just try them out. As you become more familiar with the game, you will begin to develop strategies, and you will make your choices on the basis of these strategies. It is then that you will begin to climb from "minor nuisance" to "major catastrophe."

Epyx does give you some help in forming your strategies. On the back of each monster's card you will find some notes on the monster's origins and on the basic strategic approach appropriate to each.

The degree of difficulty of the game is another issue which Epyx has considered carefully. Games must not be so easy that the average player conquers them immediately. On the other hand, they cannot be so hard that they discourage most people from trying.

Some games solve this problem by providing the player with a choice of levels of difficulty. Crush, Crumble and Chomp solves both this problem and question of giving the player enough variety to keep him interested game after game with its variety of choices of monster versus city versus objective. Some of the choices are very difficult to succeed at.

This is not to say that Crush, Crumble and Chomp is a perfect game. From the technical standpoint, it runs very slowly. Movement is taken in turns with your monster moving first and then the other units—crowds, police cars, military units, etc.—moving one at a time. When the computer has several of these on the screen, each turn can seem to take forever.

As a game player there is one major change I would like to see in the game. I would like the opportunity to fight against the monster. This would let me deal with an entirely different strategic situation, although it would, of course, defeat the role-playing aspect of the game.

That, however, is not a criticism of the game. I consider Crush, Crumble and Chomp to be a must for the serious microcomputer gamer's collection, and I firmly hope it is an indication of better things to come both from Epyx and their competitors. ■

Murder in the Mansion and Other Computer Adventures in Pocket Basic for the TRS-80

Jim Cole
ARCsoft Publishers
Woodsboro, MD
$5.98
by Bert Latamore
***Desktop Computing* staff**

The pocket computer isn't meant to be a game machine. It has a limited memory and text display area and no graphics. Given this, Jim Cole may well have attempted the impossible in this book of 10 game programs. It is, therefore, hard for me to decide whether to congratulate Cole on his technical achievement or criticize him for his basic failure in game design.

Murder

For they do all perform exactly as promised in the sample runs. The games are all based on pseudo-random number generation. In each, the player is asked to put in a number at the start of the game. This request is, wherever possible, phrased to fit into the game: i.e., "How many assistants do you need?" to help solve the murder, or, "How many men on deck?" in your spaceship. The program uses this number in a formula to generate pseudo-randomness to vary the games. Cole gives two methods for doing this in an appendix.

In a way it is amazing Cole managed to get so much variety out of such a simple idea. The title game is a mystery which asks you to discover the place of the murder, weapon and, finally, the murderer. If you guess wrong the first time, it gives you a transparent clue.

Safari

Lost Safari is a rescue of a hunting party in which the player guesses what the party was hunting, what African country they are in and how many days it will take to reach them.

Saturn Death is a space opera in which you must deal with asteroids and an "alien death probe." You are asked how many shots you need to deal with the problem. Your choice determines what happens. You may miss, damage or destroy the target. And the asteroid can hit and destroy you, the probe can fire back and damage or destroy you. This is undoubtedly the showiest of the games.

Treasure and Scrambled Eggs

Buried Treasure is a treasure hunt, similar to Lost Safari. The Black Pearl, Barrel of Apples and Wood Chuck Chuck are thinly disguised number-guessing games that quickly approach maximum boredom.

Memory Test scores you on how well you remember a list of words and a secret number. Scrambled Eggs presents you

> *"Jim Cole may well have attempted the impossible in this book of 10 game programs."*

with mixed-up letters you must sort out to spell words. Fractured Descriptions, the last program, is a party game which scrambles similes to inform you "Bob is smart as a song" or "Joyce is tall as an apple."

"The book includes a sample run of each program."

The book includes a sample run of each program. It also has a list of abbreviations usable in Pocket Basic (For example Pa. for Pause, B. for Beep, I. for Input.)

Lacking True Randomness

Unfortunately, the programs, while elegant as exercises in Pocket Basic, are totally unsatisfactory as games. Most are supposed to be strategy games. In fact they are simply number-guessing games. Since they lack true randomness, you will soon discover choosing the same number twice gives you the same response twice.

Memory Test and Scrambled Eggs are better, but they are too limited. The former has very few strings of words, and the "secret number" is always a multiple of 99. The word game can only handle 15 words, and since you must program them in, you know the answers before you start. Fractured Descriptions is cute, but it might be more fun as a word game without the machine doing the work for you.

I do not mean this as a criticism of Cole. I think he did what he could with the machine he had. It is not his fault that this machine cannot support a game. The only criticism I have of his work is the typeface chosen for the book—which appears to be directly off a computer printer—is sometimes hard to read.

I can criticize the concept, however. I hope if Cole writes more game programs he works with a more capable machine.

Whether you should buy the book depends on your interest. If you are looking for an evening's fun with a new game, forget it. On the other hand, if you want to study some finely thought-out programs—if you've just bought a Pocket Computer, for instance, and you are learning to program it—then this book might be just the thing for you.■

Computer Downs
Ernest H. Fellows
San Antonio, TX
$30 cassette

by Joel Benjamin

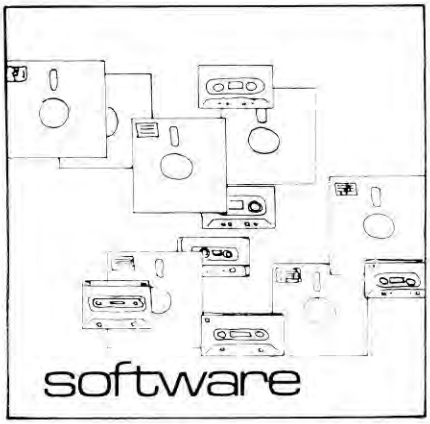
There are many lessons to be learned at the race track. After many agonies of defeat, Ernest Fellows and John Cater put together a TRS-80 16K, Level II program to take a lot of the guessing out of picking a winner.

The documentation is well organized. It asks for quite a bit of information about each horse. This is to your advantage, since the accuracy of the program's predictions depends upon the number of relevant factors considered.

The program is easy to use. If you make a mistake inputting the left arrow erases it. The more critical answers use error trapping; a message appears stating there has been an incorrect answer input and you have another chance to answer the same question. Most program requests for information may be bypassed by pressing Enter. The screen formatting is well organized. Above the input prompts, a table continually displays the previously input information and a calculated speed rating for each horse. To tell the computer to start its evaluation, enter " " when the name of the next horse is requested.

Table 1 outlines the data requested by the program.

Although the program is designed well, runs smoothly and seems to be free of bugs, there are a few features which it lacks that would enhance its usefulness. A printout option would be useful. If the results were somewhat spurious, you could carefully check the printout of your input data for errors. Then, if the program had an editing option, you could return to the input mode and change the incorrect entry and recompute the results. Besides giving you the ability to correct input errors, an editing option would enhance your flexibility. You could give different weights to different factors by changing the input data. For example, if you thought that a particular horse had a smaller chance of winning in today's race because he is being ridden by a different jockey than he is used to, you could tell the program that he placed in his last race instead of having won.



"There are many factors to consider in choosing horses. Computer Downs uses most of the obvious ones."

There are many factors to consider in choosing horses. Computer Downs uses most of the obvious ones. Several factors it does not consider are: track condition, track surface, the track surfaces of a horse's previous races, and whether the jockey is comparable to the jockey who rode the horse in the best of his last three races.

When balancing the strong points with these negatives Computer Downs is still an excellent program. It has continually proven its accuracy at the track. It is important to note that this as well as any other horse race handicapping program is an aid. It does not profess to remove all risk from your decision. But, every bit of relevant information helps to give you a better idea of which horse is most likely to win.■

✓See List of Advertisers on page 290

80 APPLICATIONS

by Dennis Kitsz

"Imagine putting RAM where Basic resides so the entire Color Computer memory might be made up of RAM."

It annoys everyone. The Technical Reference Handbook explains it. There has never been a solution for it. Irritated by one glitch too many, I decided to clean up those streaks on my Model I video display—you know, the ones most visible when you draw graphics.

This month, we will revisit computer jargon and apply some dejargonized concepts to create a sweet, glitch-free video image—for less than two dollars. Also, a passel of updates will end the column.

The first part of this column is mostly for newcomers to TRS-80 computers. You veterans can skip to the video glitch modification about midway through. You may want to tag along though, because this first section will cover microcomputer terms, a description of the expansion connectors of the Models I/III and Color, some digital logic, and a few other topics that might be news.

About Conventions

You often read phrases like "when WR* goes low, flip-flop U44 will deselect buffer U17, causing the output to tri-state and releasing control of the RAM for DMA." This is code, jargon, techspeak, whatever. If you're candid, you'll admit that you use it, too. The classical musician's techspeak sounds like this: "Gentlemen, at L the beat comes in on 4, please, not a forte staccato but sforzando as it ritards up to M, where it's subito sotto voce. L, please . . . and 3." The sports announcer provides greater examples: "Squatiero, who's warming up in the bullpen, has a career average of 204, is 4 for 5 today, with 2 RBI's. Pitch is high outside, 3 and 2 the count." Or how about: "Use caution at ASA 12; even with flash, stopped to f/1.2, the automatic feature will set 1/125 exposure, resulting in underexposure or reciprocity failure."

Mechanics, homemakers, dentists, artists, doctors, hookers, librarians, lawyers, cooks, pushers, teachers—all professionals create a jargon to facilitate conceptually complex communication between familiars. Threatening though it may be to a novice, techspeak can be valuable. I leave the hard analysis of those thoughts to Eric Maloney—his column "As the Word Turns," in *Kilobaud Microcomputing* alone is worth the subscription price. My purpose this month is neither to castigate nor commend the jargophiles, but only to explain some terms and conventions, particularly those of TRS-80 style microcomputing.

No one has written an excellent glossary of microcomputer terms yet; the only serviceable one is "Microcomputer Lexicon," published by Sybex (2020 Milvia Street, Berkeley, California 94704; $2.95). It answers many questions, but often fails to distinguish between an acronym and a name based in reality (Pascal). It is a good pocket reference. It defines but fails to explain.

For TRS-80 talk, you're on your own. (The Sybex Lexicon even omits the TRS-80 and Tandy/Radio Shack from its listings.) For starters, here are some general terms and their TRS-80 specifics.

CPU, Instructions, Architecture, Instruction Set, Clock

The central processing unit, or *CPU*, is a general-purpose digital device which reacts to complex electrical conditions with predictable and repeatable results. These electrical conditions, called *machine instructions*, are organized in specific patterns. The entire collection of patterns (known as the *instruction set*) and the CPU's overall reaction to them are called the device's *architecture*. The instructions are processed by the CPU according to a timetable set by the steady ticking of a master *clock*. Microcomputer clocks run from just under one million clock cycles per second (1 Megahertz, or *MHz*) to over six million cycles per second (6 MHz).

Data, Address Space, Memory, Read, Write

A computer is designed to work with information; for simplicity, the patterns representing information are produced in an electrically identical way to machine instructions. These are called *data*. The area in which all these kinds of information—instructions and data—are found is called the *address space*. The most significant part of this address space is consumed by electrical circuits which store instructions and data: the *memory*. In order to find and use the information in memory, the CPU must read from memory (obtain information) and *write* to memory (store information).

Address, Address Bus, Data Bus, Memory Map, Serial Data

The computer finds the memory to read from or write to by identifying the memory's location, called its *address*. For reasons of electronic simplicity, the CPU responds only to electrical on/off patterns—digital information. Therefore, to *address* a memory location, the computer uses a set of parallel, simultaneous electronic signals called an *address bus*. For the same reasons, instructions or data are sent via a different set of parallel electronic pathways called a *data bus*. How the memories (and other devices) are arranged in the address space is called the *memory map*.

In some cases, data is sent along a single path (not a bus), with the information in a line rather than in parallel; this is called *serial* data.

Read/Write, Memory RAM, Access, Access Time nS

Several types of memory have been created to meet different needs. The first need is for storage and retrieval, that is, memory which can be both written to and read from. This is correctly called *read/write memory*, but more generally known as *random-access memory*, abbreviated RAM. The "random access" part of the term means that any address may be used by the CPU at random to read from or write to (*access*) memory, rather than in some fixed sequence. The time it takes memory to be electronically ready after being sent its address from the CPU is called *access time*. Access time is written in billionths of a second, or nanoseconds (*nS*).

Read-Only Memory, ROM, Programs, PROM, EPROM, EAROM

Another form of memory cannot be changed (written to); such memory normally stores all the permanent instructions and data needed by the CPU to get the computer system up and running. Since this memory can only be read, it is called *read-only memory*, or ROM.

RAM (or R/WM) and ROM are used together by the CPU. They contain organized sets of instructions and data—computer *programs*. Thus, the memory is *programed*. How the memory is programmed suggests that there are different kinds of ROM. ROM which is created at the factory by "masking" internal connections in order to create instruction and data patterns is called mask-programmed read-only memory, or simply ROM. ROMs which can be permanently programmed by the end-user are called programmable read-only memory, or *PROM*. ROMs used for development purposes which must occasionally be erased and used again are called erasable, programmable, read-only memory, *EPROM*. Finally, ROMs which can be changed slowly by the computer's signals are termed electronically alterable read-only memory, *EAROM*.

Static and Dynamic-
Memory, Cell, Refresh

There are two major kinds of read/write memory (RAM). *Static memory* retains instructions and data so long as power is applied to the computer; each memory *cell* contains several electronic parts to accomplish this. *Dynamic memory* relies not only on continuous power, but also on an external electrical jolt called *refresh* to keep instructions and data in place; each memory cell is a single electronic part.

Chip, Integrated Circuit,
Pinout, Digital Logic, TTL, MOS

In speaking of a CPU or a memory, what is generally meant is a single gadget containing hundreds or thousands of electronic parts arranged on a single *chip* of silicon. The silicon chip is packaged in a ceramic or plastic case (a *package*) with between 8 and 64 external connections; it is called an *integrated circuit*. The size and capability of any integrated circuit is determined by what it is supposed to do, whether the parts fit into it, how many external connections it must have (its *pinout*), how much power it will use, and how much heat it will give off.

Integrated circuits used for *digital logic* (as in computers) come in many electrical "families." The most commonly used is direct transistor-output-to-transistor-input logic, or *TTL*, which is powered by a standard 5-volt power source. A special version of this TTL family, low-power Schottky TTL (LS TTL) is the present-day choice for computer construction. Other digital logic types are low power metal-oxide semiconductors (*MOS*, including CMOS, NMOS, PMOS, and HMOS families).

TTL static memory is simple to use, but is hard to make memories of many cells because it would have high power consumption and radiate a great deal of heat. Consequently, static memory can store less information per integrated circuit. MOS dynamic memory is harder to use because it must be refreshed, but demands very little power. It can store a large amount of information per package.

Bit, Word, K, Byte, Nybble

What is the "amount" of information stored by memory? The smallest piece of digital information is the on or off condition—the binary digit, or *bit*. Memory storage is determined by how many bits are available simultaneously (in parallel—the *word*), and how many different address locations these parallel bit groups make up. Since all the digital work is in binary (base 2) format, the number of different addresses is always a power of 2; for convenience, blocks of 1,024 addresses are abbreviated *K*, a pretty close approximation of 1,000. The format is written in bits, address-by-word, such as 8,192 × 1 (8K × 1) or 1,024 × 4 (1K × 4).

The size of the CPU address space is determined by the total number of bits in the CPU address bus. Thus, a 16-bit address bus runs from 2-to-the-power-0 to 2-to-the-power-15 (zero always counts in digital information!), or 65536 unique numbered addresses. The size of the CPU instruction or data "chunk" is also determined by the number of bits in the CPU data bus. An 8-bit data bus runs from 2-to-the-0 to 2-to-the-7th, making available 256 different instructions or data values. This is also called the CPU *word*. A word eight bits in size is called a *byte*; a word four bits in size is (I'm not kidding) called a *nybble*.

I/O, Memory Mapping, Ports, Port Addressing

To be useful, a computer has input/output (*I/O*) devices. These range from simple (a cassette motor switch) to complex (a disk drive). Such devices may be connected to the computer so they occupy a numbered address location, making them look exactly like memory to the CPU. Since they are given space in the memory map, the technique is called *memory-mapped I/O*. Some CPUs contain separate electrical connections for I/O called *ports*, and this method is called *port addressing*.

TRS-80 Specifics

With the descriptions above, that "same old stuff" about TRS-80s should make a little more sense: Try these:

The TRS-80 Model I contains a Z80 CPU designed by Zilog, Inc., which has a 16-bit address bus and an 8-bit data bus making it capable of addressing 64K of memory. It can execute more than 1,000 one- and two-byte machine instructions. The memory map places Level I Basic ROM in the first 4K or Level II Basic ROM in the first 12K; the two languages cannot coexist. The map has an unused area slightly less than 2K long, ending with memory-mapped disk and printer I/O, memory-mapped dual-cassette selection. Following this is a keyboard mapped in 2K of address space and duplicated in the next 2K, video memory mapped in the following 2K, and ending with 16K of dynamic RAM. Cassette I/O and certain video functions are port addressed, as well as a serial (type RS-232) device. The dynamic RAM consists of 8 integrated circuits, each 16K × 1 bits; video RAM is 7 integrated circuits, each 1K × 1 bits. System expansion allows an additional 32K of RAM. The basic computer contains 80 integrated circuits. The clock runs at 1.774 MHz, with memory access time of 350 nS.

The TRS-80 Model III differs in that the unused memory area contains ROM, disk I/O is port addressed, and the clock runs at 2.02752 MHz with memory access time of 300 nS. It is mostly compatible with the Model I. The computer contains 108 integrated circuits. Video RAM is 2 integrated circuits, each 1K × 4 bits.

The TRS-80 Color Computer contains a 6809E CPU made by Motorola, Inc., which has a 16-bit address bus and an 8-bit data bus capable of addressing 64K of memory. A special address management integrated circuit permits a theoretical expansion to 96K bytes of memory. It is capable of executing several hundred machine instructions. RAM is mapped in the first 32K address space, including RAM used for special CPU functions and the video memory. Extended Color Basic occupies 8K of space, followed by Color Basic in 8K. These two languages can coexist. The remaining 16K can be occupied by dynamic RAM or a ROM or EPROM cartridge, and the final 256 bytes are used for memory-mapped I/O, including color video control, and cassette, keyboard, sound, printer, RS-232 and joystick I/O. Dynamic RAM consists of 8 integrated circuits, each either 4K × 1 (4K system), 16K × 1 (16K system), or 64K × 1 (32K system, half the cells unused) bits. The clock speed is 0.889 MHz (software changeable to 1.778 MHz), with a memory access time of 250 nS.

Edge Connectors

Now we can use the computer jargon presented earlier. All TRS-80 computers can be expanded using the "computer bus" (containing address bus and data bus, plus other signals), which is available on its card edge (Model I/III) or card edge connector (Color). The pinouts of the respective computer connectors are shown in Figs. 1, 2 and 3.

Look at all three diagrams, and find the common items. First, eight connections carry the parallel data, marked D0 through D7 (XDB0 and XDB7 on the Model III—those X's will be explained later). Sixteen connections carry parallel address bits, A0 through A15 (only XA0 through XA7 on the Model III). Several ground (GND) lines

1	RAS*	2	SYSRES*
3	CAS*	4	A10
5	A12	6	A13
7	A15	8	GND
9	A11	10	A14
11	A8	12	OUT*
13	WR*	14	INTAK*
15	RD*	16	MUX
17	A9	18	D4
19	IN*	20	D7
21	INT*	22	D1
23	TEST*	24	D6
25	A0	26	D3
27	A1	28	D5
29	GND	30	D0
31	A4	32	D2
33	WAIT*	34	A3
35	A5	36	A7
37	GND	38	A6
39	GND	40	A2

Fig. 1. Model I Expansion Connector.

1	XDB0	2	GND
3	XDB1	4	GND
5	XDB2	6	GND
7	XDB3	8	GND
9	XDB4	10	GND
11	XDB5	12	GND
13	XDB6	14	GND
15	XDB7	16	GND
17	XA0	18	GND
19	XA1	20	GND
21	XA2	22	GND
23	XA3	24	GND
25	XA4	26	GND
27	XA5	28	GND
29	XA6	30	GND
31	XA7	32	GND
33	XIN*	34	GND
35	XOUT*	36	GND
37	XRESET*	38	GND
39	IOBUSINT*	40	GND
41	IOBUSWAIT*	42	GND
43	EXTIOSEL*	44	GND
45	N.C.	46	GND
47	XM1*	48	GND
49	XIORQ*	50	GND

Fig. 2. Model III Expansion Connector.

1	− 12V	2	+ 12V
3	HALT*	4	NMI*
5	RESET*	6	E
7	Q*	8	CART*
9	+ 5V	10	D0
11	D1	12	D2
13	D3	14	D4
15	D5	16	D6
17	D7	18	R/W*
19	A0	20	A1
21	A2	22	A3
23	A4	24	A5
25	A6	26	A7
27	A8	28	A9
29	A10	30	A11
31	A12	32	CTS*
33	GND	34	GND
35	SND	36	SCS*
37	A13	38	A14
39	A15	40	SLENB

Fig. 3. Color Computer Expansion Connector.

are connected together (common). The Color Computer provides some power on the edge connector (+5, +12, −12 volts), and one pin on the Model III is unconnected (N.C.). That covers more than half the expansion bus in one quick paragraph.

More is revealed from the description of computer terms. Memory must be read from and written to by the CPU, and there is also port-addressed I/O. These are handled by read (RD), write (WR), input (IN), and output (OUT) connections. Notice, though, that on the Model I these connections are starred. That indicator tells an important story: These signals do their work—are *active*—in the "zero" state. That is, when the RD* line goes from 1 to 0, the CPU wants to read memory; when WR* goes from 1 to 0, the CPU wants to write to memory; and so forth. This is called an *active low* signal. Sometimes the signal is written with a bar over the name to show it is active low.

In any case, there are slight differences in the three computers, but the principle is the same. The Model I has four separate lines (RD*, WR*, IN*, and OUT*), all active low; the Model III only has two (XIN* and XOUT*); and the Color Computer has a

1	RAS*
2	SYSRES*
3	CAS*
14	INTAK*
16	MUX
21	INT*
23	TEST*
33	WAIT*

Fig. 4. Model I Expansion Connector, unique signals.

37 XRESET*
39 IOBUSINT*
41 IOBUSWAIT*
43 EXTIOSEL*
47 XM1*
49 XIORQ*

Fig. 5. Model III Expansion Connector, unique signals.

3 HALT*
4 NMI*
5 RESET*
6 E
7 Q*
8 CART*
32 CTS*
35 SND
36 SCS*
40 SLENB

Fig. 6. Color Computer Expansion Connector, unique signals.

composite signal indicating read when high, and write when low (R/W*).

By eliminating data, address, read/write, in/out, and ground these from the expansion diagrams, only a few unique signals remain; these are shown in Figs. 4, 5 and 6.

Here are the most straightforward of those signals and their uses.

Unique Model I Signals

On the Model I, a low level applied to INT* (interrupt input) causes the processor to enter an interrupt-service subroutine, a program which takes priority over the program in progress. Upon doing so, it issues a low signal, INTAK* (interrupt acknowledged) so the device which asked for the interrupt can know it was received. This process is generally called *handshaking.*

When a low level is connected to WAIT*, the processor suspends normal operation until the low level is removed. This is a handy input for slow memory or other devices which operate slower than the CPU. It is not implemented in most TRS-80 peripherals.

Another little-used input is TEST*. A zero level on this input causes the CPU and all its control lines (address, data, read, write, and so on) to be set to an "invisible" state, neither one nor zero, called *tri-state.* Although mostly used to examine the circuitry during initial testing of the computer, the TEST* line can also be used to scrap the entire CPU/control circuitry and substitute a newer, faster Z80 CPU, better memory control, or even an entirely new memory map. Uses of this control signal are very sophisticated, and include allowing the CP/M disk operating system to run on the Model I.

Finally, SYSRES* is an output which goes from 1 to 0 when the system is reset, as it would be during power-up. This valuable signal can reset all peripheral devices to a known start-up condition before the computer needs them.

That leaves only RAS*, CAS*, and MUX. Those will be explained later.

Unique Model III Signals

The Model III differs from the Model I in the way external devices *(peripherals)* are dealt with. Unlike the Model I, this computer was designed as a complete unit; no expansion interface is needed to complete the system. Thus, all peripherals are considered clearly "external" and are isolated from the internal computer pathway by swinging electronic doorways called *bidirectional buffers.* When these buffers are told to send information in or out, they are enabled or selected.

That explains the X's in the Model III expansion connector: "eXternal." The CPU issues a signal inside the computer called ENEXTIO*—ENable EXternal Input/Output. When that signal enables external data XD0 through XD7, external port addresses XA0 through XA7, and peripheral input and output XIN* and XOUT* for use by the CPU. Upon ENEXTIO*, the system reset signal (XRESET*) is also allowed to pass through to external devices.

Likewise, with ENEXTIO* activated, the CPU can receive input from the outside world. IOBUSINT* (I/O BUS INTerrupt) and IOBUSWAIT* (I/O BUS WAIT) work like the Model I INT* and WAIT* lines, described earlier.

Three connections remain, EXTIOSEL*, XM1*, and XIORQ*. EXTIOSEL* (EXTernal I/O SELect) is an active low input to the Model III from a peripheral device; when the peripheral device places a 0 on this line, it turns around *(switches)* the buffers, aiming them at the CPU so it can read information from that peripheral device. XM1* (eXternal Machine cycle 1) informs interested devices that the CPU is about to obtain its next machine instruction from memory. XIORQ* (eXternal I/O ReQuest) indicates that a request for input/output information (rather than a request for memory read/write) is being issued by the CPU.

Unique Color Computer Signals

The Color Computer is quite a different animal from the Model I/II/III family because it uses a CPU with quite different architecture; its special features are reflected in the expansion connector.

If a low signal is applied to RESET*, the computer is brought to a power-up or "OK" condition like the Model I's reset button. HALT* is similar to the Model I TEST* signal, in that the CPU is put in a tri-state condition, giving control over to some external device. NMI* (Non-Maskable Interrupt) is an interrupt signal like that on the Model I, except that it cannot be ignored by the processor—it has *higher priority.*

The remaining signals are quite different. E and Q* are the CPU's master clock pulses. The 6809E CPU requires two clocks of identical frequency, but of different *phase*; that is, the Q* clock's "tick" occurs before the E clock's "tick." The reason for placing these pulses on the computer bus will be discussed later.

CART* (CARTridge in place) is an input signal to the computer telling it that a cartridge has been inserted in the expansion slot, and CTS* (CarTridge Select) is a computer signal to enable (select) the ROMs in that cartridge. SND* (SouND) is an input to the computer with which the ROM cartridge can feed information directly to the sound output circuitry. SCS* (Spare Condition Select) is a spare internal computer enabling signal by which a combination of software and hardware can access a user-defined external device.

Finally, there is a powerful input signal, SLEND* (SeLect ENable/Disable). Internally, the CPU and support ICs select RAM, ROM, and peripheral control circuitry. By placing a low level on SLEND*, that selection task is given to the peripheral device. What good is that? The memory map can be redefined. Imagine putting RAM where Basic resides, for example, so the entire Color Computer memory might be made up of RAM.

Now back to some mysterious signals— RAS*, CAS*, and MUX (Model I), E and Q* (Color Computer). Since computers operate at such high speeds, it is always important "when," and these signals can supply that information. Specifically, that "when" can apply to dynamic memories.

Dynamic memories are very peculiar; I'd like to describe a little about how they work. The most popular dynamic memories today are the "16K's"—16Kx1 bits. That is, 16,384 bits of information are stored in a single integrated circuit, and eight of these circuits together can store 16K bytes. Only one bit of data is stored at each address, but there are 16K worth of addresses (2-to-the-14th power). Now what might the pinout of a 16Kx1 memory chip be? There are data input and output lines, a read/write line, three power lines

(+12, +5, −5 volts), and ground... that's seven. But, since this is a 16-pin integrated circuit, there are only 9 lines left. How can 14 address lines fit into the nine remaining on the memory circuit?

The answer is an imaginative technique called *multiplexing*. To "multiplex" is to create a traffic pattern—at a traffic light, one line of cars moves while the other waits, and vice versa when the light changes. The memory chip is multiplexed: It receives seven memory address bits, the traffic light changes, and the remaining seven address bits are fed to it.

That explains RAS*, CAS*, and MUX. RAS* (Row Address Strobe) informs the memory circuit that the first group of address bits is on the way. There is a pause while the memory chip goes about receiving these address bits, and then a change in the MUX (MUltipleX) signal informs special multiplexer chips that the second set of address bits must be sent to the memory. CAS* (Column Address Strobe) then heralds the arrival of the remaining half of the memory address bits. At that point, the memory is ready to provide or receive CPU data. Altogether there are only seven address lines, plus RAS* and CAS*.

By the way, the names of these signals also indicate that the memory is set up in a grid of rows and columns. Also, notice that the word "strobe" suggests a quick flash of light; quite so. It's less than 100 billionths of a second each for the operation of RAS*, CAS* and MUX, making this memory a lightning-fast circuit.

You have no doubt noticed that the Color Computer provides only two signals, E and Q*. Proper combinations of E and Q*, using digital logic circuits, can precisely simulate the actions of RAS*, CAS* and MUX.

Aside to Newcomers

I would like to believe that the column so far has brought you from a state of wide-eyed wonder to being ready for certification as a Professional Engineer. The general concepts have been turned into TRS-80-specific jargon. However, it might be too much for one sitting, so take a break and come back to it. More is coming (for Model I owners at least) as the video destreakifier modification is explained and installed.

Video and Streaks

A master clock frequency provides all the pulses for every electronic event in the TRS-80. A series of integrated circuits divide this 10.6442 MHz master frequency by 672 to create 15,840 Hz, slightly above but close enough to the standard 15,750

✓See List of Advertisers on page 290

Hz television horizontal sweep rate. This is divided by 264 to provide the 60.0 Hz vertical frame rate. In other words, the television screen consists of two "frames" of lines, each 262-½ horizontal lines per frame. The second frame is displayed after the first, offset vertically on the screen, and interlaced to give the appearance of a screen containing 525 lines. This pair of frames is repeated 30 times per second. You can see these lines more easily on your computer monitor than on an ordinary television, but they are there in all cases. In order to synchronize the computer to the display, then, 15,840 Hz horizontal and 60 Hz vertical signals are needed.

As you know, one area of the computer is called "video memory." This memory is counted through, byte-by-byte in binary by the same divider chain that creates horizontal and vertical sync. Each second, the entire video memory is scanned through 60 times. Each byte of video memory is read into a holding circuit called a latch. When ready to be displayed, the contents of the latch are fed to a shift register; this circuit takes the parallel group of dots which make up part of a letter, and shifts them a dot at a time out to the video display, in synchronization with the horizontal scanning rate of the monitor.

Sometimes no letters should be displayed—the top, bottom, left and right borders of the screen. From the video divider chain, then, another piece of information is created to blank out the display in the borders. The signal is conveniently called BLANK* (meaning the screen is blanked out when the signal is low).

That covers the bulk of the display process, though one thing is still missing. How does the CPU get to use the video memory to change the display, or otherwise read or write to it (such as Basic's PEEK and POKE)? VID*, another internal computer signal, does the job. When VID* changes from 1 to 0, it tells the video display circuitry that the CPU needs the video memory. The display circuitry clears out its display latch, and gives control of the memory to the CPU.

So there's the reason for the streaks displayed during graphics. The "dots" are cleared out of the latch, and the CPU takes over. What you are seeing is a combination of the time the CPU needs to use video memory, plus some additional blank space because the dots were removed from the holding latch. Why clear the dots from the latch? Because unless you do, by the time the video display circuitry gets back to the display process, it's too late to finish displaying the current character on the screen. So instead of having bits and pieces of characters floating randomly about the screen, the TRS-80 designers decided to clear the whole latch in favor of regular black streaks.

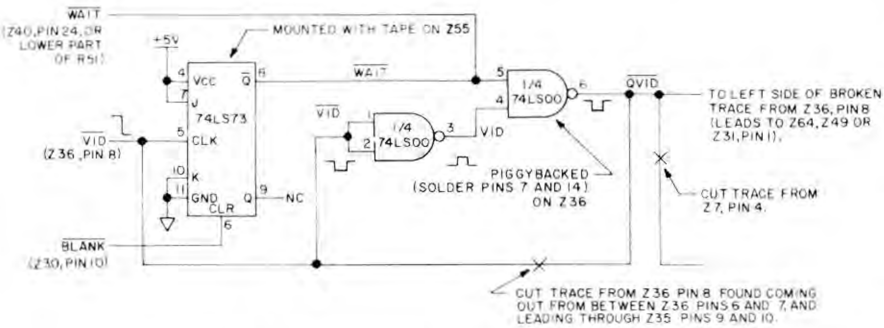


Fig. 7. Destreakifier Modification Schematic.

Destreakifying

How do you cure the streak problem? Simply prevent the CPU from accessing video memory when the electron beam is within the display bounds. (My use of the word "simply," let me add, is a retrospective one.)

The information is available—BLANK* is high when the screen is used. How to

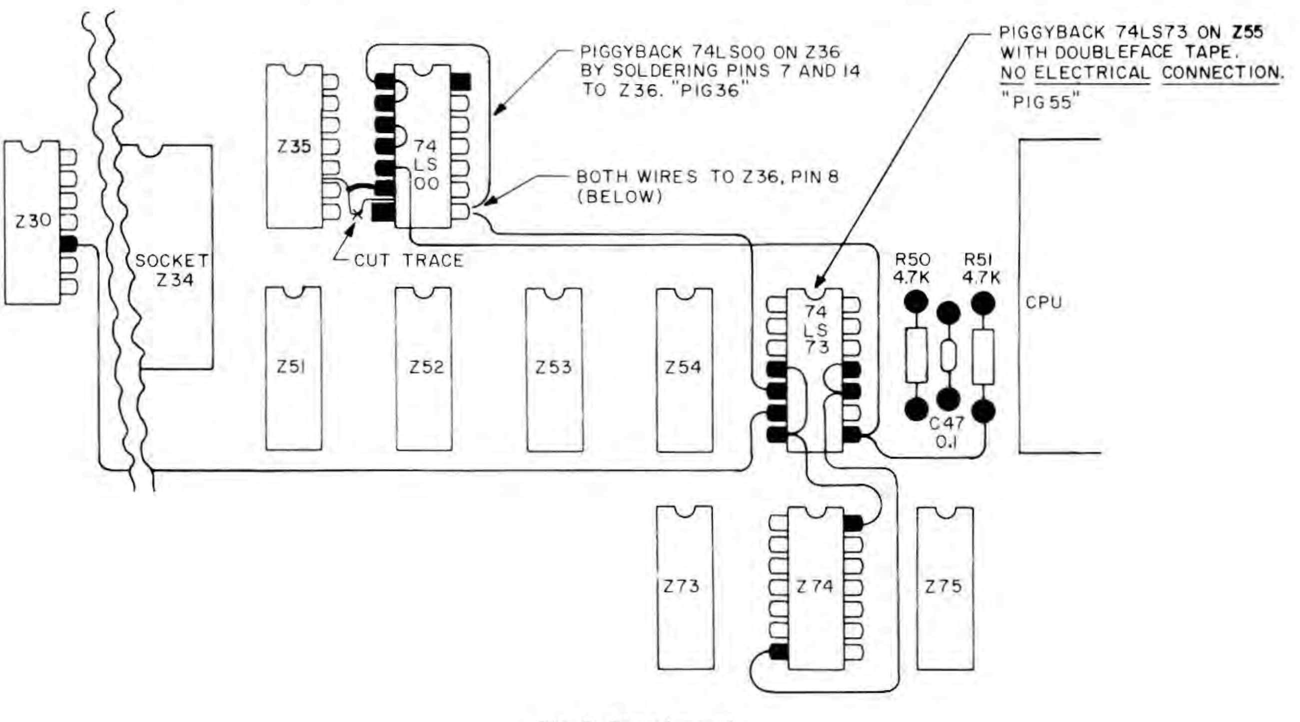


Fig. 8. Parts Layout.

tell the CPU to wait when BLANK* is high? By telling it to WAIT*. Recall the description of the Model I's WAIT* line: Hold it low and the CPU suspends operation until you release it. Though the simplest method to deglitch the screen, it has one drawback—*whenever* the screen is being displayed, the CPU will wait. That will slow the CPU's operation by 50 percent, hardly a desirable situation.

Instead, the CPU should only suspend operation when the screen is displayed and it wants access to the screen. How to know when it wants access to the screen? VID* will be low. So when BLANK* is high and VID* is low, the processor must WAIT*.

Almost good enough. But it's necessary to keep some form of VID* low—latched—so that the video memory won't be freed as soon as VID* resumes high a short time later. It must continue to latch until the border of the screen is reached, then release the CPU from its WAIT* condition.

Installing the Mod

The installation of the destreakifier will be described first. Fig. 7 is a complete

"How do you cure the streak problem?"

schematic of the modification, and Fig. 8 is the actual parts layout. Two integrated circuits are needed, one 74LS73 and one 74LS00 (Radio Shack parts #276-1801 and #276-1803, totaling $1.48). You will also need a piece of double-face tape, a razor blade or X-acto knife, a soldering iron, a few feet of fine insulated wire, and solder.

The computer must be opened; please refer to last month's Applications for instructions on opening your TRS-80. Find the integrated circuits marked Z36 and Z55. Take the 74LS73, and bend all its pins parallel with its body. Apply a piece of double-face tape to the bottom. Key the 74LS73 in the same direction as Z55 (with the notch or dot matching on both), and affix it to Z55. The piggybacked 74LS73 will now be called PIG55. Take the 74LS00, and bend all pins parallel with its body except pins #7 and #14 (the bottom left and top right pins). Key the 74LS00 in the same direction as Z36, place the 74LS00 on Z36, and solder pins 7 and 14 of both together. The piggybacked 74LS00 will be called PIG36.

Locate the trace leading from Z36 pin 8. It can be found snaking out between Z36

See List of Advertisers on page 290

pins 6 and 7, bending down and up again as it slides between Z35 pins 9 and 10 (see Fig. 8). With a sharp razor blade or X-acto knife cut this trace, and scrape off some of the green solder mask to expose the bare metal.

Solder together PIG36 pins 1 and 2. Solder together pins 3 and 4. Run a wire from PIG36 pin 5 to PIG55 pin 8, and solder. Solder a short wire to PIG36 pin 6, and the other end to the left side of the cut trace (refer to Fig. 8). Finally, solder a wire to PIG36 pin 1, and lead it around to Z36, soldering it to pin 8; don't accidentally solder to PIG36 pin 8.

Solder a second wire to Z36 pin 8, and run this to the PIG55, pin 5. Solder together PIG55 pins 10 and 11, and run a wire from here to Z74 pin 7. Connect a short wire from PIG55 pin 4 to pin 7, and another from there to Z74 pin 14. Now locate R51. Solder a wire to its lower side (refer to Fig. 8), and solder the other end to PIG55 pin 8, which already has a wire attached. Solder a long wire to PIG55 pin 6, and run this 'way over to Z30 pin 10.

Finally, locate Z7, turn the circuit board over, and cut the trace leading from Z7 pin 4. This completes the destreakifier modification.

The 74LS73 is a flip-flop, or "toggling" circuit. It can also be used as a latch, which is what is done here. Fig. 9 shows the pinout of this circuit so you can refer to it when examining how the installed modification compares with this theory; likewise, Fig. 11 is the pinout of the 74LS00.

The 74LS73 has several inputs. "J" and "K" are input data of some kind, clear (CLR) sets the output to a known state, and clock (CLK), when it drops from 1 to 0, causes a change in those outputs. Fig. 10 is the truth table of the 74LS73; a truth table is a chart of all the known input and output conditions of a digital logic integrated circuit.

I will skip the reasons I chose the 74LS73 and the 74LS00 and work backwards. Notice that CLR, when it is low, causes the two outputs to be low and high, respectively. This is the known condition to which the 74LS73 is cleared.

The next four conditions show what happens to the outputs when CLK falls from 1 to 0. With both J and K low, the outputs remain at their previous state, whatever that was. With J high and K low, the outputs become high and low, and vice versa. And finally, with both high, the outputs will toggle from their previous state (again, whatever that was) to their inverse. If they were low and high, they will toggle to high and low. When the CLK signal returns to high, no change appears at the outputs.

Fig. 12 shows the truth table for the 74LS00. It is a simple Not-AND (NAND) gate. You can see that the output of this circuit is always 1 (high) unless both inputs are high; then the output drops low.

BLANK* will be attached to CLR, and VID* will be connected to CLK. J will be tied high, and K will be tied low. Q* will be sent to the computer's WAIT* line. Now look at Fig. 13. Since J and K are permanently set, three conditions can be dropped from the truth table for this application.

If CLR is held low, Q* will be high. That means: When BLANK* (the input to CLR) is low, WAIT* (the output from Q*) will be high. That is, the CPU will continue its normal operation unaffected.

If CLR goes high (BLANK* is high), then the display is being drawn; two conditions may then exist. If VID* (the input to CLK) drops low, then video memory is needed by the CPU. Since BLANK* is high, the screen is in use and the CPU must be told to wait. Thus, WAIT* (connected to Q*) drops low which suspends the CPU operation. If BLANK* is still high when VID* goes back high, it will be latched there until BLANK* resumes low, when the display is back in the border area.

So far, so good. The CPU is being told to wait when the video display circuitry is using the screen. However, VID* will trigger memory access (and corresponding streaks) anyway unless something is done about it. Cutting the trace from Z36 pin 8 takes care of that, but how can the CPU get access to the memory at all?

By combining VID* when it is low (memory needed) and WAIT* when it is high

TRUTH TABLE
74LS73

INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	↓	L	L	Qo	$\bar{Q}$o
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	TOGGLE	
H	H	X	X	Qo	$\bar{Q}$o

Fig. 10. Truth Table of the 74LS73.

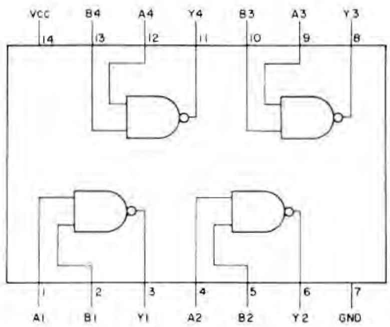


Fig. 11. Pinout of 74LS00 Circuit.

TRUTH TABLE
74LS00

INPUTS		OUTPUT
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

Fig. 12. Truth Table for 74LS00.

TRUTH TABLE
74LS73
WITH J AND K PERMANENTLY SET H AND L

INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	↓	H	L	H	L
H	H	X	X	Qo	$\bar{Q}$o
$\overline{\text{BLANK}}$	$\overline{\text{VID}}$				$\overline{\text{WAIT}}$

MEANING: WHEN $\overline{\text{BLANK}}$ IS LOW (AT CLR), $\overline{\text{WAIT}}$ (AT $\bar{Q}$) IS HIGH.

MEANING: WHEN $\overline{\text{BLANK}}$ IS HIGH (AT CLR), AND $\overline{\text{VID}}$ (AT CLK) FALLS, $\overline{\text{WAIT}}$ (AT $\bar{Q}$) IS LOW.

MEANING: WHEN $\overline{\text{BLANK}}$ IS HIGH (AT CLR), AND $\overline{\text{VID}}$ (AT CLK) IS EITHER LOW OR HIGH, $\overline{\text{WAIT}}$ (AT $\bar{Q}$) STAYS IN ITS PREVIOUS STATE.

Fig. 13.

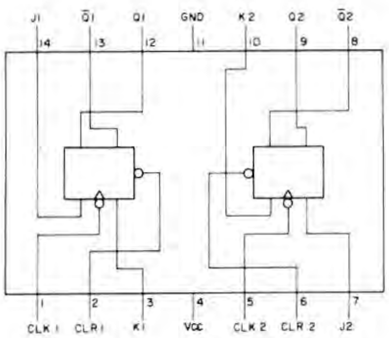


Fig. 9. Pinout of 74LS73 Circuit.

See List of Advertisers on page 290

(normal CPU operation), a new signal I've called QVID* has been created. It signals memory access only during normal CPU operation. Therefore, feeding QVID* to the former VID* input of memory fills the second need.

Finally, the signal that cleared the dots from the video latch (called VCLR* in the technical manual) is disabled by cutting the signal from Z7 free.

Fig. 14 is a composite truth table showing all the possible input conditions to the 74LS73 and the 74LS00, how they interact, and the final result at QVID*.

How does this modification affect performance? Obviously, the streaks are gone. And program speed is virtually unaffected by the occurrence of WAIT*. On video-intensive programs (such as games with high-speed graphics) processor delay is hardly noticeable—less than 10 percent slower in the worst case I was able to test. Only one situation might be abnormal, and that is where the video access is an inherent part of a crucial timing loop; the maximum delay (assuming the CPU attempts to access video at the very beginning of a horizontal scan) is 31.7 microseconds.

Final Note to Newcomers

I hope that this journey through TRS-80 microcomputer jargon was useful to you, and demonstrated that, like most specialized language, it is meant to simplify not obfuscate communication. To an engineer I might have described my entire column this way:

"Hello, Mike? Open up your tech manual to the video schematic. Pick up BLANK and VID*. If you clear an 'LS73 with BLANK*, and clock it with VID*, you can force wait-states on the processor; just hold J high and K low. Then if you cut loose VID* from the video memory multiplexor, and NAND WAIT* and NOT VID* through an 'LS00, you've got an off-screen valid memory access. I called it QVID*. Free up VCLR* from the video latch, and the screen is de-glitched. Check it out and give me a call.*

"Oh, Mike, you still there? Now get this. The best part is that there is a free 'LS73 and two free NAND gates already on board the TRS-80. They could have built it with a glitch-free screen, right from the start. For free! Later, Mike. . . "

Lots of Updates

This month's updates cover a lot of material that has crossed my desk. Please glance through these updates to see if I answer any of your questions before you write. . .

● I have been using a preliminary release of a version of Flex, the standard disk operating system for 68XX family computers; it's up and running on the Color Computer (!). It is sector-based yet fast, and contains a text editor, debugging monitor, and assembler, as well as two dozen utilities. Watch for it, and plenty of support from the Flex community. It's sort of the CP/M for the 68XX family, and even in this preliminary version it runs circles around CCDOS (Exatron) and the preliminary Color TRSDOS (Radio Shack).

● In December Applications (page 95), there is a problem with the Motorola 7 × 9 character generator suggested as an alternative device for the Color Computer lowercase project. The MCM66700 generators have not been meeting "typical" specifications listed in the Motorola data sheet. Instead, they have been closer to 450 nS access time, and only half of my samples even met that time. Motorola engineers say they intend to downgrade the specs rather than upgrade the chip (not a big seller). In place of these character generators, I have created a group of character sets in EPROMs. In case you want to begin creating the Color Computer lowercase project, the pinout of these circuits follows:

Pin	Function	Pin	Function
1	ASCII bit 3	24	Vc, +5 volts
2	ASCII bit 2	23	ASCII bit 4
3	ASCII bit 1	22	ASCII bit 5
4	ASCII bit 0	21	Vpp, to Vcc (+5 v)
5	Row Select 3	20	Chip Select (low)
6	Row Select 2	19	ASCII bit 6
7	Row Select 1	18	Output Enable (low)
8	Row Select 0	17	Dot Column 0 (not used)
9	Dot Column 1	16	Dot Column 7
10	Dot Column 2	15	Dot Column 6
11	Dot Column 3	14	Dot Column 5
12	Ground	13	Dot Column 4

These character generators are available from MSB Electronics (Drawer 766, Barre, VT 05641) for $15.00, plus shipping.

● Regarding February 1981 Applications: The PC boards and complete units for the Memory Sidecar are now sold by MSB Electronics instead of The Peripheral People.

● Using the high-speed CPU mode on the Color Computer is not always possible. Among the culprits are the peripheral interface chips, which you can replace with 68A21 if the installed ones are too slow. Also, your CPU may be too slow; some of the plastic 6809E processors can't keep up. Replace it with a 68A09E if you really want higher speed. Since the CPU alternates between high and low speed in normal operation (more on that some day), the only real speed gains will be when running some aspects of Basic.

● Regarding December Applications (page 83), for those who wanted to install 32K RAM in their Color Computers and noticed some differences inside the computer, here's the news. The bulk of Color Computers up until November were the "D" boards, which match my description. The newer "E" boards contain five new jumpers marked 16K/32K. These need not be changed to accomplish the installation I described. However (a big however, please), it seems (which means I haven't grabbed one to look inside) that the cost of 64K dynamic memories has come down far enough to give Radio Shack a profit at their memory upgrade retail price. Industry prices for these 64K memories are now under $20 for a set of 8 (that's *64K* memory

COMPOSITE TRUTH TABLE
OF
DESTREAKIFIER MODIFICATION

74LS73			74LS00					
FLIPFLOP INPUTS		FLIPFLOP OUTPUT	NAND GATE INPUTS #1		NAND GATE OUTPUT	NAND GATE INPUTS #2		NAND GATE OUTPUT
CLR	CLK	$\overline{Q}$	A1	B1	Y1	A2	B2	Y2
$=\overline{BLANK}$	$=\overline{VID}$	$=\overline{WAIT}$	$=B1$ $=\overline{VID}$	$=A1$ $=\overline{VID}$	=NOT (A1 AND B1)	$=\overline{WAIT}$	=Y1	$=\overline{QVID}$
L	L	H	L	L	H	H	H	L
L	H	H	H	H	L	H	L	H
H	↓	L	↓	↓	↑	L	↑	H
H	L	L	L	L	H	L	H	H
H	H	$\overline{Q}$o (L)	H	H	L	$\overline{Q}$o (L)	L	H

$\overline{\text{WAIT}}$ WHEN $\overline{\text{BLANK}}$ IS HIGH (BEAM ON SCREEN) AND $\overline{\text{VID}}$ IS LOW (ACCESS REQUESTED BY CPU).

LOCK PREVIOUS STATE AFTER $\overline{\text{VID}}$ HAS FALLEN. FOLLOWING THE LOGIC, $\overline{\text{WAIT}}$ MUST ALWAYS LOCK LOW.

CPU ACCESS TO VIDEO MEMORY PERMITTED HERE—ONLY WHEN $\overline{\text{BLANK}}$ IS LOW (IN BORDERS OF SCREEN) AND $\overline{\text{VID}}$ IS LOW (ACCESS TO VIDEO IS REQUESTED BY CPU).

Fig. 14. Composite Truth Table.

I'm talking about now!), in thousand or more quantities. I'll verify these changes and report the details in a future update.

● In August Applications (page 58), I talked about my sudden rash of computer problems. These were traced to lightning (which blew up my modem the following week). As a result, my first-hand findings on lightning and other power supply surge protection will be covered in this column in the future. For those who want a jump on this, a high quality TII lightning surge protector is available from Datadyne, 450 Seventh Ave., New York, New York 10001. It is $14.95. MOV overvoltage protectors are also available from Datadyne.

● October Applications (page 62) presented several versions of a machine language monitor in Basic. A reader reports that the Extended Color Basic version ends prematurely, but since I have not been able to simulate this problem with my machine, I'd like readers' help. Here's the question: ". . . it generated a 'run end' in the middle of line 13. I did what page 150 of the Extended Basic manual suggested (defined all string data in line 1 and added + " "), and the program then performed satisfactorily. This seems a limiting bug in Basic." I would appreciate a post card from readers who have encountered this problem, showing the program lines in question.

● In September Applications (page 48), the captions for Photos 5 and 6 were reversed. The upper photo depicts transistors, the lower photo shows diodes.

● In December Applications (page 84), the IN wire to the video buffer amp should be connected to the left side of R16 when looking from the front of the computer. If you find that your monitor screen washes out with this connection, then attach the IN lead to the input of the video modulator module (the small metal cage at the back of the computer). The input of the video modulator box is the back rightmost of the four wires connected to it, nearest the plastic support post. Several readers reported that the video signal was too strong and washed out the screen colors.

". . . these 64K memories are now under $20 for a set of 8. . ."

● In May Applications (page 51), the LEDs in the Micro Front Panel schematic were shown with their polarity in reverse. The Micro Front Panel is now available assembled from MSB Electronics for $45.

● Here is a final set of clarifications (I hope) for the high-resolution graphics board (The Detailer, July). Thanks to Norris March, PE, of Pompano Beach, Florida, for questioning these items:

1. The ICs shown below Z27 a/b were unmarked; these should be Z27 c/d, part of the same integrated circuit.

2. In Fig. 4, Z15d should be marked Z1d, part of a 74LS04; the pins are 13 and 12, not 13 and 11 as shown.

3. In Fig. 1, Z15d is correctly connected in the countdown chain as shown. The reference to 1.774 MHz in the text should be changed to 0.887 MHz.

4. In Fig. 2, a drafting error left the bar off the MWR signal leading to the input to Z21 pin 3.

5. In Fig. 4, mixing of the TRS-80 and Detailer outputs can be done through coupling resistors or via an op amp, as shown below:

6. In Fig. 1, syncing is correctly accomplished by Z33. The reference to Z31 in the text is a typographical error.

7. In Fig. 1, Z15c is shown as an inverter. To configure the 74LS00 as an inverter, input pins 10 and 9 are connected together.

8. In Fig. 2, only one set of resistors R18 to R26 is needed to feed all 6 memory chips. The abbreviated format of the schematic made this unclear.

9. In Fig. 1, the connection to Z11 pin 5 is not drawn clearly. It is connected to the 5-volt supply.

10. It was not obvious that a second TRS-80 power supply is necessary to power The Detailer; do not use the power supply for the keyboard unit to power both it and the high-resolution board.

11. In Fig. 2, WR* to Z31A should be labeled CWR*. It is connected to the computer WR* line, and MWR* is derived from it. ■

● *Next month: Make your own Color Computer ROM packs.*

✓ See List of Advertisers on page 290

SOFTWARE FOR YOUR 16K TRS-80 COLOR MODEL I, III, ATARI 400/800, APPLE II

the programmer's program™

Do you know all the innovative ways of using BASIC commands? The staff at the Programmer's Institute have spent 8 months designing a set of programs explaining everything the manuals omit. And much more.

The Programmer's Program leads you step by step through the fundamentals of programming your computer. Then you explore the finer points . . . multiple loops, queues, stacks, game programming, sorts, trees, booloean logic, and memory savers. All these methods make your programming tasks easier and more efficient.

Difficult programs are built before your very eyes, and designed so you learn at your own speed!

A must for every owner of a microcomputer ($40).

appletree™ MAGATARI™ TRC™

On cassette or diskette*, our magazines are designed explicitly for your computer. Included every month are 6-10 ready-to-load programs ranging from games, home entertainment, and personal finance, to more of our unique "teaching" programs. We will also keep you informed on all the latest products.

Our February issue will include:

1. GOLF!
2. Sea War with graphics.
3. Schedule 1040 Long-Form.
4. Math Test with graphics.
5. Decision Maker - helps make decisions
6. Self-Programming Programs.
7. Converting Programs.

. . . and as always you get our 100% commitment to excellence and service.

(Back issues are available!)

Later issues will include backgammon, forecasting, stock market, and many more. The price per subscription to TRC, APPLETREE, or MAGATARI is $50 per year, $30 per ½ year, and $10 for a trial issue.

Don't miss out on our February issue.

Our complete home financial package with check maintenance, income/net worth statements, graphing, budgeting, mailing list, and appointment/bill calendars is available for only $50 or ABSOLUTELY FREE with any $50 order. Purchase both above packages and your net cost is only $75. ORDER TODAY!

THE PROGRAMMER'S INSTITUTE

A Futurehouse Company
P.O. Box 3191 Dept. B
Chapel Hill, N.C. 27514
(919) 489-2198
MC & Visa Welcome

name ____________
address ____________
city/st ____________

☐ Programmer's Program ☐ Year Subscription ☐ ½ Year ☐ Both ☐ Trial Issue
Computer Model ____________ ☐ Cassette ☐ Diskette

TRC and the programmer's program are trademarks of Futurehouse.

*All software available on cassette for the TRS-80 Model I, III, Color Ext. Basic, Atari 400/800. On diskette for the Model III, Apple II. (For diskette add $1 per month).

✓241

ACTION GAMES

The fastest growing producer of computer games for your 6809 has the products you have waited for!!

NEW! ARCADE GAMES FOR THE COLOR COMPUTER

CAVE HUNTER **COLOR BERSERK**

Fast paced action • Super Hi-Res Graphics
Dynamite sound effects • Runs in 16K of memory
These games will astonish you with their Detail and Quality.
They set a standard for others to follow.

— ADVENTURES —

Calixto Island • The Black Sanctum

Highly acclaimed by reviewers • Challenging situations
Fast, efficient machine language • Runs in 16K of memory
Save game in progress

Adventures on 5¼ TSC FLEX disc (specify 6800 or 6809)	ea	$24.95
Both adventures on single disc		$39.95
Adventures for color computer	ea	$19.95
Color Berserk for color computer	ea	$24.95
Cave Hunter for color computer	ea	$24.95

Shipped prepaid in continental U.S. California residents please add 6% tax

VISA **— MORE COMING SOON —** master charge

MARK DATA PRODUCTS

23802 Barquilla, Mission Viejo, CA 92691 • (714) 768-1551

TRS 80 IS A TRADEMARK OF TANDY CORP.

✓261

80 microcomputing™ bookshelf

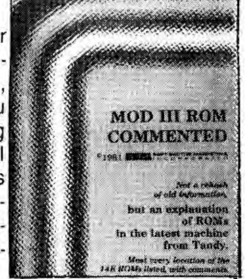
LEVEL II SYSTEM REFERENCE CARDS

At last! No more flipping through the pages of the Basic manual! No more working through the maze of machine language instructions! These cards completely summarize the Basic and Assembler manuals! FEATURES INCLUDE:
memory map, eyeball graphics, math instructions, basic commands, store instructions, basic functions, load instructions, basic statements, move instructions, special keys, exchange instructions, print using examples, shift instructions, basic special characters, compare instructions, basic and assembler messages and codes, branch instructions, basic facts, data alteration instructions, reserved words, I/O instructions, ROM routines, complete character chart with graphics and space-compression codes, hex-dec chart, control code cross-reference, assembler instructions commands and operators, screen line layout, editor commands & subcommands, condition code easy access
Plus—"magic graphics number—a mystery until you learn how to use it!
Designed as a fold-up, accordion-style card, fits in your pocket. Panels organized for optimum speed for reference.
Basic only FC1001 $2.95
Basic and Assembler FC1002 $4.95

NEW!

DOS RANDOM ACCESS & BASIC FILE HANDLING—By H.J. Muller. This book was written for the nonprogrammer. It is ideal for the businessman or professional who needs to solve and write special programs for in-house business problems, or the hobbyist who wants to go beyond the cassette recorder and into disk storage and file manipulation. It is written as a self-instruction tutorial and will provide anyone with some Level II experience with the ability to write special programs for inventories, mailing list, work scheduling, record keeping, research project data manipulation, etc. 150 pages. BK1236 $24.50

NEW! FOR THE MODEL III



MOD II ROM COMMENTED—Soft-Sector marketing, 1981. This book is not an instruction course on machine language, but rather an information source that you can use time and time again for writing your own program or patching old Mod I machine language programs. It contains an explanation of ROMs in the latest machine from Tandy, with must every location of the 14K ROMs listed, with comments. BK1235 $22.50.

NOW AVAILABLE FOR THE MODEL III

Level II System Reference Card
(as described above)
BASIC FC 1004 $3.95
BASIC & Assembler FC1003 $5.95

Prices do not include shipping and handling charges. Please include $1.50 for first book, $1.00 each additional book and $10.00 per book foreign airmail. Sorry no COD orders. Use the order card in this magazine to order. Allow 4-6 weeks for delivery.

80 Microcomputing Bookshelf • Peterborough, NH 03458

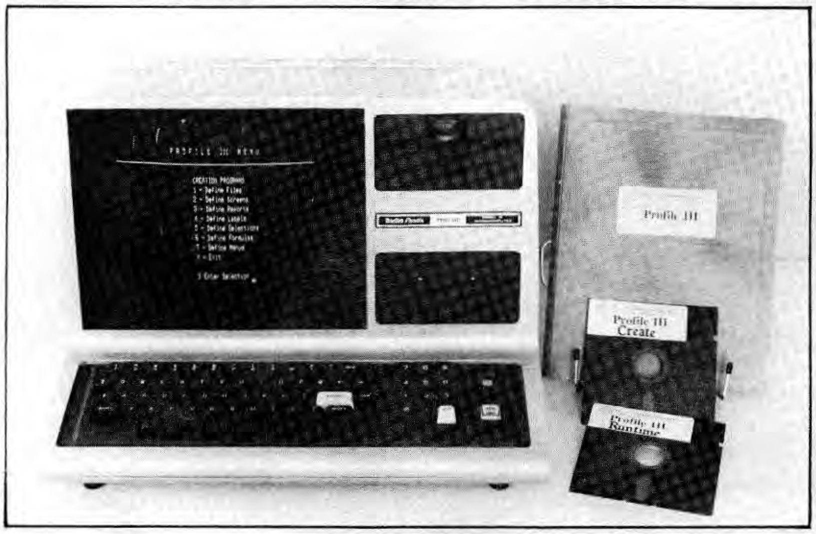
master charge **1-800-258-5473** VISA

See List of Advertisers on page 290

NEW PRODUCTS

Edited by Janet Fiderio

Featuring—Profile III + An Integrated System for Applications Generation, Financial Modeling, and Word Processing.



Profile III +

Profile III + A Data Base Management System

Profile III + is a data base management system for the Model III.

This system allows non-programmers to create sophisticated applications—such as a complete personal search system—quickly. Profile III + will interface with Visicalc and the forthcoming Radio Shack SuperScripsit, providing a powerful, integrated set of management information tools.

Designed to be user-friendly, Profile is completely menu driven. User options are shown at the bottom of the screen.

For additional information on Profile III + features contact The Small Computer Company Inc., 40 W. Ridgewood Ave., Ridgewood, NJ 07450, (201) 445-5643. Profile III + is priced at $300.

Reader Service ✓ 335

For the Pocket Computer

You and Your Dollars: All About Interest is a set of computer programs written for the pocket computer. These programs are aimed at taking the guesswork out of decisions involving today's interest rates.

This book is priced at $19; with cassette copies of the 17 programs, $33. For further information contact XCEL, 13763 Polk St., Sylmar, CA 91342.

Reader Service ✓ 344

TRS-80 More than Basic

The monitoring program presented in *TRS-80 More than Basic* turns the Model I or III into a development system by either loading object code from cassette, disk, or by ROM replacement.

A detailed description of the hardware features of both the Model I and III is provided, as well as the hardware allowing the inclusion of a single-stepping feature. The book shows how the monitor executes valid commands and flags errors. Over 26 commands are available and the user is given total documentation.

Priced at $10.95, the book is available from Howard W. Sams Inc., 4300 W. 62nd St., Indianapolis, IN 46268, (317) 298-5400.

Reader Service ✓ 329.

InfoStar

InfoStar is a CP/M-based report generator/file processor program for the Model II. It is intended as an application development tool for both novice and experienced users making it possible to produce business reports and tailor preexisting applications programs. The program is designed to grow with users as they implement more complex reporting systems.

InfoStar features a print enhancement package permitting you to draw upon print controls found in WordStar for letter-quality text.

Contact Micropro International Corporation, 1299 4th St., San Rafael CA 94901, (415) 499-0919, for additional information. InfoStar retails for under $1,000.

Reader Service ✓ 173

Tax Simulation And Planning Program

Shortax uses only 23 items of data to compute up to 18 values for 36 different combinations of taxpayers and taxable years.

The program computes the federal income tax using the tax rate schedules, the 50 percent maximum tax method or the income averaging method. For corporations, the program computes the alternative tax for long-term capital gains. If applicable it also computes the add-on minimum tax and the alternative minimum tax.

Shortax runs on CP/M and is available for the Model II. It is priced at $500, annual

✓See List of Advertisers on page 290

updates are available for $150. For more detailed information on additional Shortax features contact Syntax Corporation 4500 W. 72nd Terrace, Prairie Village, KS 66208, (913) 362-9667.

Reader Service ✓174

CP/M-80 board kit For the Model III

The CP/M 80 board kit provides the hardware, CP/M version 2.2 software, and the capability to display 1920 characters on any Model III CRT.

The kit allows you to utilize the system for business and word processing applications. The hardware needed to run CP/M includes a CP/M adapter and an 80 column video board.

A Model III with 48K is required. The package sells for $849 and is available from Microcomputer Technology Inc., 3304 W. MacArthur Blvd., Santa Ana, CA 92704, (714) 979-9925.

Reader Service ✓165

Basic Operating System

WOBOS I is a new operating system for the Models I and III for disk and cassette systems alike. WOBOS is a pre-structured program in Basic with 23 direct-access utilities including Append, Core Dump, single line renumber, Sound Generator and Device I/O control. It also features an easy to use graphics synthesizer. It is intended to be used as "stock" for all Basic applications.

The product is available on cassette for $35 from Western Operations, 6604 SW 173 Avenue, Beaverton, OR 97007, (503) 649-4526.

Reader Service ✓163

Typesetting for the TRS-80

The MSB Composer 100 is a software-based system which allows typesetting on the Models I and III. The system features spooled output of up to 64 disk files, code conversion and direct online hookup to the typesetter. Keyboarding and formatting of typeset output is accomplished through the use of Scripsit with access to typeset codes via simple MSB mnemonics.

MSB is currently available for the VIP and Linotron 202 from Merganthaler and for the Compugraphic 8600. Minimum requirements for the TRS-80 are 48K, one disk drive, and an RS-232 port.

The on-line program is priced at $1600, the off-line at $300. This system is produced by MSB Consulting, 170-15 Highland Ave., Jamaica Estates, NY 11432, (212) 526-0825.

Reader Service ✓340

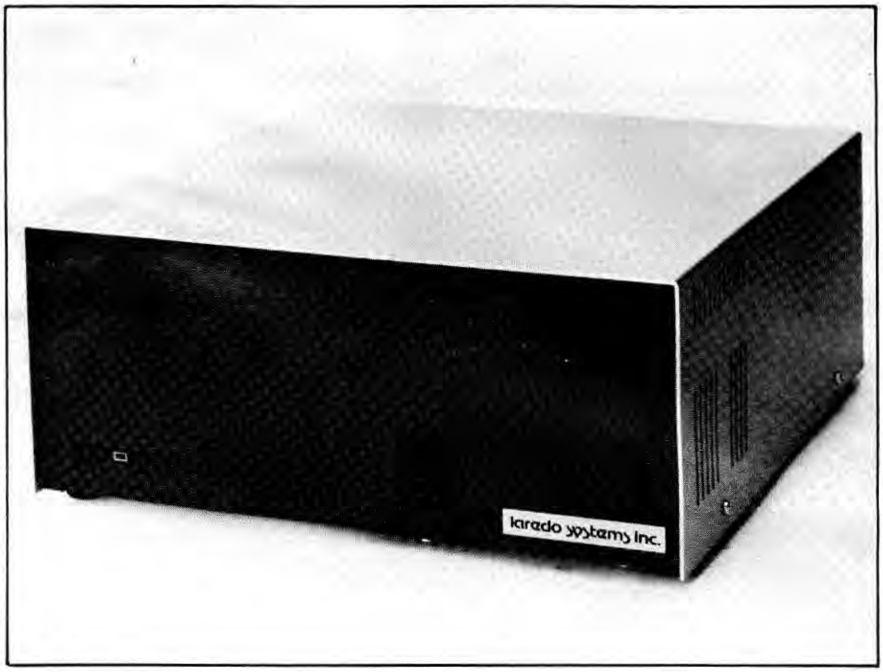
A Computer System Designed for the DME Dealer

A computer system for the DME dealer and the Model II is now being marketed.

The system features: automatic posting of repetitive monthly rentals; printing of Medicare and insurance forms, patient co-insurance statements, mailing labels; word processing; monthly sales and rental analysis; detailed age analysis; and procedure code reports. Expansion into General Ledger, Payroll and Inventory is available.

Software and hardware are available for less than $10,000 from Micromed Systems Inc., 25 South H. St., Lake Worth, FL 33460.

Reader Service ✓336



Laredo Systems LS525

Laredo Five-Megabyte Hard Disk System

The LS525, an intelligent mass data storage peripheral, incorporates a Seagate Technology ST506 5 1/4-inch hard disk drive with two magnetic disks, four read/write heads and 6.38 megabytes of total recordable storage capacity. It is controlled by a 10 MHz microprogrammed controller and operates under LDOS.

The LS525's storage capacity allows users to keep online large data bases such as inventories and mailing lists which previously required dozens of minifloppy disks. The information is immediately accessible.

For more detailed information on this data storage peripheral contact Laredo Systems Inc., 2264 Calle de Luna, Santa Clara, CA 95050-1198, (408) 980-1888. The LS525, available for the Model I and II, is priced at under $3500.

Reader Service ✓332

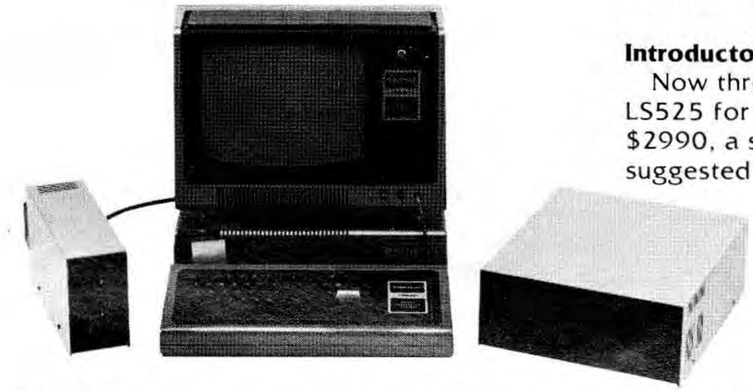
Smart Terminal Program For the Model I and III

Commwhiz is a user-friendly smart terminal program for communicating with the Source, CompuServe, other micros and a variety of bulletin board systems.

It provides a full array of features need-

THE NEW LAREDO LS525. WHEN YOU'RE READY TO GET DOWN TO BUSINESS.

Laredo Systems, Inc. introduces the LS525 five megabyte rigid disk memory system for TRS80 Models I and III. Laredo's LS525 rounds up a Seagate ST506 Drive, LDOS Operating System by Logical Systems, and Laredo's own LSI



500 Series Controller into a single-board controller that emulates the famed IBM 3370 disk system, complete with:

- Improved read/write/seek access time through full block buffering and variably tuned interleave.
- On-board data separator, micro-code diagnostics and write-precompensation.
- Increased disk media reliability and data integrity with automatic defect block allocation and extended ID fields.
- Versatile CPU compatibility provided by separate host adaptor.

Introductory Offer To TRS80 Users.

Now through October 1, 1981 buy the LS525 for the OEM/dealer price of $2990, a savings of $760 off the usual suggested retail price of $3750.

LDOS Operating System with manual and diskette available for $149.

For more information about the LS525 Memory System and the LSI 500 Series Controller, contact: **Laredo Systems, Inc.**
2264 Calle de Luna,
Santa Clara, CA 95050
(408) 980-1888

✓26

laredo systems inc.

ed for uploading and downloading, disk loading and saving, using text prepared with Scripsit or Electric Pencil, Basic programs or machine language source code files, and more. Commwhiz is easily operated by a novice.

For details on additional Commwhiz features contact VolksMicro Computer Systems Inc., 202 Packets Court, Suite C, Williamsburg, VA 23185.

Reader Service ✓337

Color Computer Adventure

Madness and the Minotaur takes you into the depths of King Mino's castle in search of his treasure. While searching through over 200 rooms for the treasures avoid the traps, hazards and creatures. You must also search for and learn a raft of magic spells which are necessary to conquer all the hazards and foes of the labyrinth. The game runs in real time so the creatures and hazards are moving and occurring whether or not you are moving. The location of the treasures, creatures, magic spells and traps are randomized for each game.

Written in machine language for a 16K TRS-80 Color Computer, this game is priced at $19.95. Extended Basic and joysticks are not required. Contact Spectral Associates, 141 Harvard Ave., Tacoma, WA 98466, (206) 565-8483 for additional information.

Reader Service ✓170

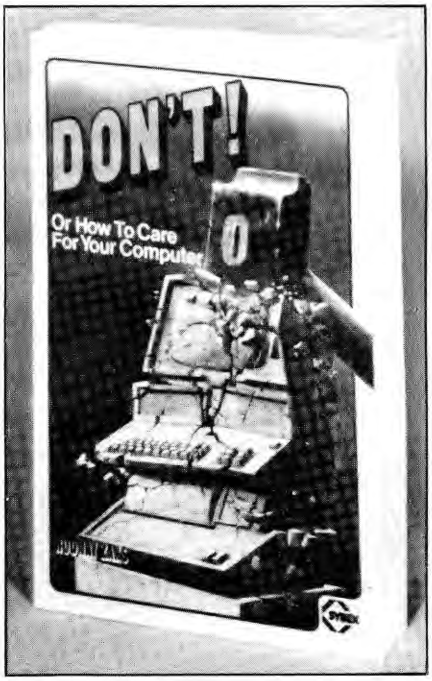
Space Waste Race A Computerized Storybook

Space Waste Race is a childrens' computerized storybook aimed primarily at pre-school and kindergarten aged children. This whimsical story, through its pictures, sound, words, and animation, both entertains and educates. It allows the child to interact with the ideas and graphics from the story in a hands-on learning situation.

Its story entertains while the learning games encourage an interest in reading, letters, numbers, space and direction.

A Model I or III 16K tape is priced at $19.95, a 32K disk at $24.95. Contact Storybooks of the Future, 527-41st St., San Francisco, CA 94121, (415) 386-5184, for additional information.

Reader Service ✓ 328



Don't (Or How to Care For Your Computer)

"Don't" or How to Care For Your Microcomputer

Don't (Or How to Care For Your Microcomputer) explains how to handle and maintain all components of a microcomputer system: the computer proper, the CRT display, the disks and tapes, and the printer.

The book is designed to save computer users unnecessary aggravation, time and money. It also features information concerning planning a computer room, preserving documentation for a system, avoiding computer theft, and more.

Priced at $11.95, *"Don't"* is available from Sybex Inc., 2344 Sixth St., Berkely, CA 94710, (800) 227-2346.

Reader Service ✓333

Farm and Ranch Management Programs

Thirteen new programs concerning farm and ranch management are now available for Models I, II and III.

The programs are: Beef Projection, Beef Ration, Dairy Cow, Feed Lot, Farm Record, Government Program, Grain Storage, Loan, Machine Cost, Record Management, Estate Tax, Feed Formulator, and Pasture Projection.

For more detailed information contact American Small Business Computers, 118 South Mill St., Pryor, OK 74361, (918) 825-4844.

Reader Service ✓171

An Online Stock Market Service

Marketscan allows the user to get daily quotations for all stock markets listed on the Toronto, Montreal, Vancouver, Alberta, New York, and American stock exchanges.

The high, low, close, and trading volume are displayed in familiar formats. Marketscan services include the ability to: analyze a specific stock on one or several exchanges at one time; monitor an entire exchange; or compare a stock's activity within a specific range of dates.

For more information on Marketscan, contact Info Globe, 444 Front Street West, Toronto, Ontario, Canada M5V 2S9, (416) 598-5250.

Reader Service ✓ 330

MX-Stand Stack Rack

The MX-Stand stack rack is designed for use with an MX-80 printer.

The rack stores up to 600 sheets of paper conveniently tucked beneath the printer giving you more desktop space. The paper compartment accurately aligns the paper with the sprocket drives for jam-free operation. In addition the bale guide prevents the paper from snagging on the bottom of the printer.

The MX-Stand is available for $14.95 from Remtron, Box 2280, Santa Clara, CA 95055.

Reader Service ✓172

Smart Terminal Program For the Color Computer

The Colorcom/E is a smart terminal program for the Color Computer. It comes in a ROM cartridge ready to plug in and run.

Colorcom/E's features and capabilities include: online and offline scrolling; offline printing of data; receiving and sending cassette files; support of any serial printer; full and half duplex; and an optional word mode to eliminate word wrap. Data can be easily edited before printing or writing to cassette.

Priced at $49.95, this program is available from Eigen Systems, PO Box 10234, Austin, TX 78766, (512) 837-4665.

Reader Service ✓342

Art for the Computer Buff

Four-color art for the computer buff, ranging from posters to a series of cartoons, is now available through Inmac.

Selections include abstract art posters and calendars of greatly magnified computer chips and wafers, cartoons of computer characters, and a stylized map of Silicon Valley.

Contact Inmac, 2465 Augustine Dr., Santa Clara, CA 95051, (408) 727-1970, for ordering information.

Reader Service ✓334

Zoom Offers New Software

Three new software programs for the Model I are now available.

QKEDT ($14.95), a utility for cassette Electric Pencil users, converts Basic programs into ASCII files which can then be read into the Pencil, and vice versa. Basic programs can be typed in using the Pencil instead of the Level II line editor.

TCHTYP ($10.95), useful for teaching children numbers and letters or for practicing your touch typing, fills the screen with one giant character on the instant any key is hit. Written in machine code, this program provides both upper and lowercase characters. A graphics editor permits modification of characters.

ZMBUG 2.0 ($17.50, for Zoom 3.6 users) searches for file names, verifies recordings, lists addresses of all files encountered, converts Basic programming into Pencil files, and loads and saves at 3,600 bits per second.

These programs are available from Zoom, Box 1496, Raton, NM 87740, (505) 445-5787.

Reader Service ✓175

Inmac's Art Line

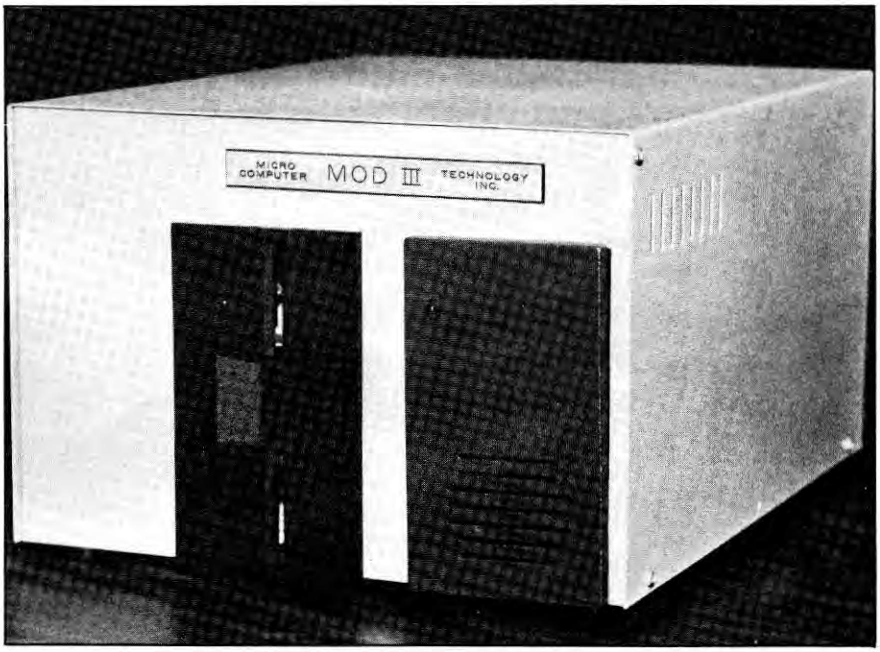
External Hard Disk Systems

A 5 and 7.5MB external Winchester Hard disk System for use with any TRS-80 is now being marketed.

This product uses Micro Systems hard disk operating system version 3.3.9 (model 1507) which is file compatible with TRSDOS and priced at $299.

The 5MB external disk drive, model 1421, is a complete self contained unit including a Winchester drive, controller, power supply, and cables, priced at $2799. For additional information contact Microcomputer Technology Inc., 3304 W. MacArthur Blvd, Santa Ana, CA 92704, (714) 979-9925.

Reader Service ✓166



External Winchester Hard Disk System

Income Tax Programs on Disk

40 Income Tax programs are now available on formatted disks for the Models I and III. The documentation supplied with each disk also explains the various methods used in programming.

The first disk includes programs for Form 1040 and schedules A, B, C, D, SE and TC and is priced at $24.95. The remaining tax programs are available on additional disks from Gooth Software, 931 S. Bemiston, St. Louis, MO 63105, (314) 727-2229.

Reader Service ✓338

Medical Research Data Management Software

Clinistat combines data-base management with statistical and graphics output.

This software package is designed for use by either the novice or professional. It has eight main features: a user designed data-base filing system; easy entry, revision, addition and deletion of data; infinite sorting/selection and computation capability; user-controlled selection of data display mode; optional printing of all or part of the data or analysis; various statistical analyses; X-Y graphics plotting and printing capability; and the ability to backup.

Clinistat is available for the Model II, 64K from MDp Software, Box 360, Great Neck, NY 11022.

Reader Service ✓178

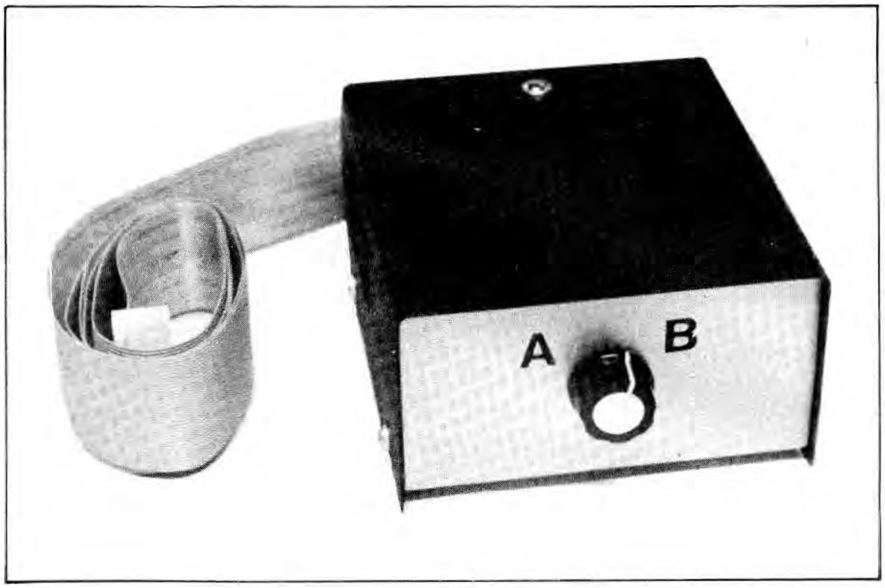
Programming The Pocket Computer

The *Beginner's Guide to Programming the Sharp PC-1211 and Radio Shack TRS-80 Pocket Computer* is a novice's guide to Basic.

The guide explains everything in simple, understandable language. Learning is initially keyed to sets of half-page cards for each lesson. The task on any card is finite and each part is clearly explained giving the beginner an immediate check on correctness.

This book is sold for $19 and is available from XCEL, 13763 Polk St., Sylmar, CA 91342, (213) 367-4366.

Reader Service ✓179



The MFJ Inductive Coupled Modem

New Modem

The MFJ-1230 originate/answer modem is used like an acoustic-coupled modem but employs an inductive coupling technique for receiving.

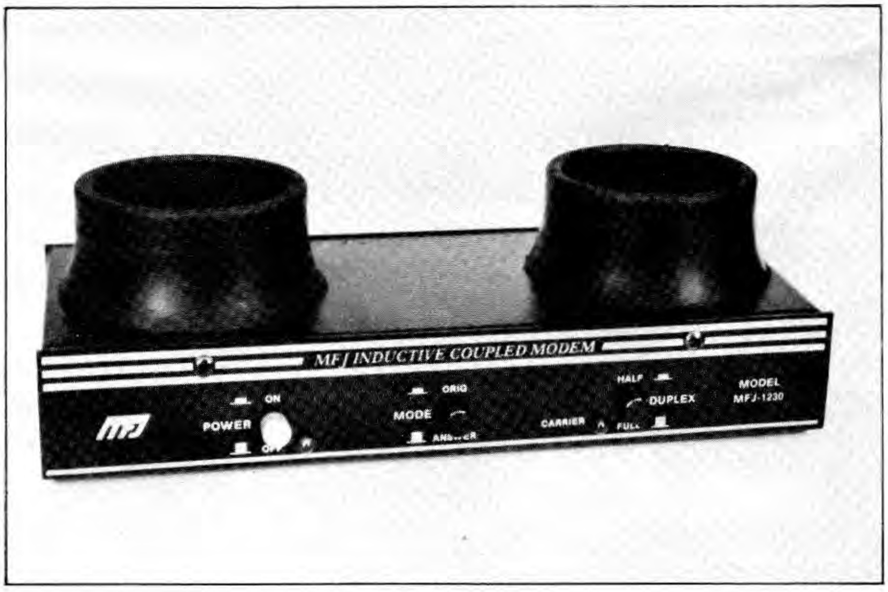
This Bell 103 compatible modem operates from zero to 300 baud, features half and full duplex operation, and is crystal controlled for high stability. It provides TTL and CMOS inputs/outputs as well as RS-232 compatibility. Also, input/output ports for a cassette tape recorder lets you save your transmitted data, load it back into the computer, or retransmit it.

The MFJ-1230 is priced at $129.95 and is available from MFJ Manufacturing Inc., 921 Louisville Road, Starkville, MS 39759, (601) 323-5869.

Reader Service ✓167

Quick-Switch Units

Quick-Switch Units allows you to switch a TRS-80 printer output back and



Quick-Switch Unit

Model III

DISK SYSTEMS

We have upgraded the TRS80 Model III 16k Computer to 48k with our own high quality memory chips and added our top of the line disk drive systems to make this your best buy of the season. These computers are functionally identical to the Radio Shack units sold at computer centers for $500 more. Only prime high quality components (described below) are used in the construction of these units. These computers have the same type double density disk drives as the original. They have the same storage capacity, are completely software compatible with TRSDOS, DOSPLUS, NEWDOS/80, LDOS, VTOS and many other disk operating systems, and are compatible with all BASIC applications software available for the Model III.

INSTRUCTIONS - For kit installations we provide complete instructions.

√**COMPUTER** - The Model III computer unit is an original unit which has extended BASIC and upper/lower case. It is packed in original factory packing, and is new and unused. Each computer, after having the drive systems and memory added, must pass a 24 hour burn-in test to assure maximum reliability.

√**DISK DRIVES** - TCS offers standard 5 1/4" disk drives from TANDON, the same people that make the original drives for the Model III. These drives have band head positioning for long life, and have a fast 5 millisecond track to track access time. This means that if LDOS or one of the other fine disk operating systems is used, the track access time may be adjusted to allow faster operation of the entire system. Some brands of drives (Siemens, older Shugart) have access times as slow as 40 milliseconds, which can result in slow system speed especially when a considerable amount of random disk I/O is needed, such as programs manipulating large lists of data. This means that you should check track access capability before deciding on a particular drive. Each of our drives is tested fully for alignment, speed and proper operation before shipment.

√**MEMORY** - The memory used in the TCS Model III is the highest quality Japanese made memory available. In a recent test, Japanese memory was shown to be over twice as reliable as American made memory of the same type. We use only first line (no surplus) chips that are rated 200 nanoseconds or faster. Each set of memory units installed by our technicians is tested using several of the best commercial memory test programs available. These tests last for a period of over 24 hours.

TEXAS COMPUTER SYSTEMS

Offers Lowest Prices on

TRS-80

√**DISK CONTROLLER** - TCS uses a high quality, double sided glass epoxy disk controller board that has GOLD plated edge connectors for reliable data connection. We use a reliable digital data separator design for the upmost in long life reliability. This means that adjustments NEVER need be made to the controller circuit for it to keep separating data correctly. This is the highest quality disk controller board available for the Model III.

√**POWER SUPPLY** - Where many companies cut corners, TCS has chosen the finest switching power supply available. A switching power supply is much more efficient than linear supplies, which means that the TCS supply creates less heat than a linear supply, and it uses less energy. As a result, the internal parts of the Model III can operate cooler (the Model III has no cooling fan) without the need of an expensive internal cooling system. This power supply is the same commercial grade switcher that is used in the original more expensive Model III.

√**MOUNTING HARDWARE & CABLING** - The disk drive mounting hardware is aluminum sheet construction for easy installation and the disk mounting holes are slotted to allow adjustment of the drives for exact alignment. The power and data cables are all included and are plug compatible with each unit. No soldering or tools are needed for installation other than a screwdriver.

KIT 1	**Controller, power supply & Mounting Hardware**	**$379**
KIT 2	**Kit 1 plus one Tandon Disk Drive**	**$595**
KIT 3	**Kit 1 plus two Tandon Disk Drives**	**$819**
KIT 4	**One Disk Drive only**	**$219**
KIT 5	**16k TCS Memory**	**$49**
KIT 6	**32k TCS Memory**	**$79**

For complete assembled and tested Model III Systems using these kits see our other ad in this issue of this magazine.

Introducing The B-TREE SERIES

B-TREE Mailing List Disk..................$49.95

A mailing list application data base. Allows the maintenance of about 1500 to 2000 names on a single double density DATA disk. Display or print in order by name or by zip code (allows 9 digits). Print mailing labels by zip, name or user selected category. Duplicate names are allowed as well as optional data fields for each record. Also allows two line addresses for long addresses or company names and phone number.

B-TREE Library Program Disk.............$39.95

A complete library maintenance data base system which allows storage of the book title, author, category, and 2 other user selected fields. Over 2000 books can be maintained on one double density DATA disk. Average search time by author is 2 seconds with 2000 records in file. Complete reports include: 1)Alphabetical listing of file by author, 2)List all books of one author by title or alphabetical by title, 3)List all books in a selected category, etc.

B-TREE Video Program Disk................$39.95

A complete data base for your video cassette collection. Allows entry of each program the stars, Video I.D. number, starting counter number, category & length. Over 1000 programs with over 300 cassettes can be stored on one double density DATA disk. Special reports include: 1)Printed Labels, 2)Alphabetical listing by stars, 3)Alpha Listing by Program, 4)Listings by cassette, 5)time left on each cassette report, etc.

WHAT IS THE B-TREE?

The above Series of programs have many exclusive features. Each application program stores a variety of data in a B-TREE Structured file. The B-Tree file organization method allows fast and efficient access to a large amount of data. With B-TREE, a whole disk of data can be accessed in alphabetic or numeric order without sorting or merging and all in a minimum of time. The user may add, delete, edit, list, print and redisplay the data without any file reorganization necessary. This is one of the BEST ANSWERS to data Storage problems on microprocessors. Fully implements the **BINARY SEARCH TREE** concepts including **insertion, deletion, editing, & traversal**. No sorting or long searches for data recovery is needed with B-TREE, yet even larger than memory files can be printed or displayed in sorted order without the necessity of sorts. Varing Key fields possible with duplicates allowed. These are BASIC programs (yet faster than many assembly language programs) with hundreds of REMark statements for easy modification or use of the modules in user programing. Also includes the best INKEY module ever written, allowing the fastest direct screen oriented input you have seen. All programs are non-scrolling with SCREEN oriented design and have all information needed on screen at all times. Single Key stroke command sequences and complete error trapping as well as controlled input of data to any field. Each package comes on a formatted diskette to run on TRSDOS 1.3, LDOS 5.1, NEWDOS80 & DOSPLUS for TRS80 Model III computers and also for Model I using LDOS or DOSPLUS. Watch for new programs in our exclusive B-TREE series in upcoming months. MODEL II versions for TRSDOS and CP/M available soon with expanded capabilities.

For fast, efficient service. **Heart of** we can air freight from Dallas

TEXAS COMPUTER SYSTEMS

P.O. Box 1327 Arlington, Texas 76004-1327

☆ Toll Free Number 800 433-5184

Texas Residents 817 274-5625

Dallas Metroplex 265-1524

* Payment: Money order, cashiers check, certified check. Prices above reflect 3% cash discount. Call for VISA/MC prices.
* Prices subject to change any time.
* No tax out-of-state. Texans add 5%.
* Many items shipped FREE. $Call.

✓25

NEW PRODUCTS

forth between two or three peripherals, or, switch one peripheral between two or three computers.

The Standard switch includes a five foot ribbon cable terminated with a 34-contact card-edge socket which interfaces a printer port connector. Rear mounted 34-pin card-edge plugs are provided for separate device connector interfaces. Versions are available with RS-232 device interfacing or Centronics 36-pin parallel device interfacing.

A two switch port costs $79.95, a three switch port is $89.95. Both are available from Access Unlimited, 401 N. Central Expwy. #600, Richardson, TX 75080, (800) 527-3475.

Reader Service ✓168

Analog Instrumentation Fundamentals

A book on Basic analog instruments—how they work, how to design them, and how to use them (*Analog Instrumentation Fundamentals*)—is now being marketed.

Analog meter movements, dc ammeters, dc voltmeters, ohmeters, rectifier-type ac voltmeters, and much more, are covered within its pages. Practical, hands-on, lab-type experiments and examples are included for clarification and self-testing. A fundamental background of electronics and basic algebra is needed.

For more detailed information contact Howard W. Sams Inc., 4300 W. 62nd St., Indianapolis, IN 46268, (317) 298-5400.

Reader Service ✓325

Inexpensive Friction Feed Conversion Kit

The Micro-Grip II adds friction feed to the MX-70, MX-80, or MX-100 Epsom printers.

This upgraded version of the original Micro-Grip easily installs leaving the existing tractor feed mechanism and capabilities completely undisturbed. Letterheads, cut-sheets, roll paper, forms, checks, and NCR forms can now be used.

This conversion kit, priced at $39.95 is available from Micro-Grip, Box 873, Langley AFB, VA 23665, (804) 826-8808.

Reader Service ✓ 326

Z80 Monitor

Bigbug is a Z80 monitor designed for the Model I and III, Level II machines putting you in control of Assembly-language programming.

Its features include a ROM-based monitor that does not need backup, tape storage with file name, direct access of I/O, printout for the debugging history, mnemonic key-in of user history, hexadecimal calculator, and built-in RS-232 driver software.

For information on additional features contact the Future Project Corporation, Box 11, Hawleyville, CT 06440, (203) 775-3062. Bigbug retails for $75.

Reader Service ✓176

The Compactor I and IV

The Compactor I and IV are two memory management products for the Model III.

The Compactor I allows CP/M applications and utility programs to be run on the Model III. The Compactor enhances but does not change the TRS-80's operating environment. When installed the user can either use CP/M (2.2), TRSDOS, TRS-Basic, or Hurricane's Z80 Diagnostic Monitor.

The Compactor IV is a dual purpose video display module. The module serves as an 80 by 24 video display and EIA standard RS-232 serial interface. The necessary cables and instructions are included.

The Compactors I and IV are priced at $450 and $475 respectively and are available from Hurricane Laboratories, Box 631, Cupertino, CA 95015, (408) 446-0777.

Reader Service ✓177

Errata

We inadvertently printed an incorrect price for Hayden Books Dentistaid program (December, 1981, p. 94). The correct price is $1000. Dentistaid is available for the Model II with CP/M. ■

CONTROL !!

is what KBE is all about ...

KBE (KeyBoard Editor) provides unparalleled support for your Model I, and now Model III, TRS-80.

KBE provides a full screen editor for BASIC, DOS, EDTASM and EDIT-80. Why purchase a screen editor for EACH environment, when KBE services them all??

KBE allows you to program 101 keys on your TRS-80 keyboard, including often overlooked keys such as shift-break and shift-enter. This allows you to embed complex command sequences and store them "out of the way" until you need them. All programmed keys (soft keys) can be toggled on or off at any time. Softkeys can be programmed to call other softkeys. Softkey definition is limited only by memory, that is, you are not limited to 255 bytes. All softkeys can be easily saved to or loaded from disk under the filename you specify. Redefine ANY key at ANY time. Have a different set of softkeys for each environment you choose: log onto Source or MicroNET with one keystroke, speed up data entry, enter program lines or subroutines quickly and easily, redefine the keyboard to accommodate esoteric keyboards. The author designed the program to assist his wife with her doctorate -- written in German. Another customer is using KBE in conjunction with a printer using a French typefont. The possibilities are unlimited!

And we're not through: KBE provides keyboard debounce, auto repeat, lower case, screen displays to printer, E/Pencil control key support, prevents printer hangup, and allows ANY ASCII character to be generated from the keyboard. KBE is compatible with all popular DOSes and double density. KBE requires a 32K disk system for operation, and is supplied on diskette with users manual.

We would say,

"Don't boot up without it!"

but you won't want to.

$39.95

MODEM 80

Communications Package

Abilities:

- Remote operation of a TRS-80 Model I or III from a terminal or a second TRS-80 through a telephone link -- files may be transferred with the unattended computer.
- Error free file transfers with another TRS-80 or a computer that can use the protocol of the CP/M program "Modem" which is widely used on computer bulletin boards (and available on CP/M user group disk number 25).
- File transfers with many other types of computers with the TRS-80 acting as a terminal. The program is set up for use with MicroNET, the Source, Forum 80's, and similar systems, but the communication parameters, character set, and control characters may be redefined to operate with many other computers.

Should you purchase Modem 80?

Just ask Bill Vermillion, Connection 80 sysop, Winter Park, FL:

"I find it is the best modem program I have used, and I am recommending it to all users of my Connection 80 Bulletin board. It is the best value that has ever been offered in a communications package. It easily outperforms packages costing 2 to 4 times as much, that don't even offer all the features. Les has truly done an outstanding programming job and no one who uses the TRS-80 for communications should be without it."

Just ask Gordon Williams, Connection 80 sysop, Lansing, MI:

This is a dynamite terminal program! If you don't have it, it is worth far more than the $40.00 price tag it carries. Not only is it very capable, it is clearly the easiest-to-use terminal program I have ever seen. I have ST80D, ST80III, UT, SMART80D, SMART80III and a couple of others. So I feel like I am in a position to compare... and Modem 80 beats them all!"

Order online: Micronet, 70150,255 or Source, TCH565

Modem 80 is compatible with *ALL* popular operating systems including LDOS, Newdos/80 (even version 2.0), DOSPLUS, TRSDOS & Ultrados.

MODEM 80 requires one disk drive and 32K

MODEM 80 is supplied on diskette with full-size manual

MODEM 80 costs just $39.95

☐ Rush me a Modem 80 I disk!
☐ Rush me a Modem 80 III disk!
☐ Rush me a KBE I disk!
☐ Rush me a KBE III disk!

Name: ______________________
Address: ______________________
City: ______________________
State, Zip: ______________________

The Alternate Source
1806 Ada Street
Lansing, Michigan 48910
517-487-3358
✓138 **517-485-0344**

Add $3.00 packaging & postage.
Add $1.75 for COD shipments.
Add 4% for charge card orders.

The Alternate Source is always accepting subscribers to their magazine. But they aren't always giving away free programs. Just now and then. Before March 31, 1982 each person subscribing/renewing for a 24 issue period to TAS and mentioning this advertisement, will receive absolutely free, a program called **FLY.** You will find this program to be literally **full of bugs** yet containing no errors. **Swarming** with action it's sure to amuse you for hours on end. **FLY** is not for sale anywhere. It is available only through this and similar advertisements, offer good until March 31 1982. Twenty four issues of TAS are $36.00. **FLY** is free. You'll **swat** yourself if you miss this one!

☐ Sample TAS $2
☐ 12 issues TAS $18
☐ 24 issues TAS $36 Send FLY!

EDUCATION 80

by Earl R. Savage

". . . the smattering of actual literacy courses were designed for a select audience . . ."

The responses to my request several months ago for computer literacy course outlines or curricula are fascinating and have lead to several interesting conclusions.

First, more readers want a computer literacy outline than have one. That should not be a surprise considering the status (or non-status) of computers in most schools.

The second conclusion was a surprise. Most of what passes for computer literacy courses actually are programming courses. Come now—being computer literate does not mean being a programmer.

The third conclusion was that the smattering of actual literacy courses were designed for a select audience—gifted, math, science, or business students. There was not a single course for the average student.

A Few Parameters

Computer literacy is not programming. I'm not knocking programming courses: Some but not all students need and want to learn programming.

Computer literacy is "knowing" computers. It is knowing something of their history, of their function, of what they can and cannot do. Computer literacy is being aware of important societal issues involving computers. It is losing, through familiarity, the distrust or fear some have for technology, especially electronic. Computer literacy is a lot of simple things. It is knowing enough to reject the "the computer made a mistake" excuse for ineffectiveness.

Everyone must be computer literate tomorrow (meaning all of today's students). The computer illiterate will be at an increasing disadvantage as computers permeate every aspect of our daily life. The time is getting short—we must prepare now.

What we need is a computer literacy course designed for all students. If an adequate course does not exist, surely we who know computers and education can put one together. If you have worked on part of a computer literacy course a lot of us would like to know about it. Let's pool our thoughts to put together a course of great potential. Send your ideas to me in Peterborough or at 6 Darl Avenue, Colonial Beach, VA 22443.

I look forward to your contributions.

For the Lazy Teacher

Are you writing instructional programs in Basic? If so, I would like to know how you find the time.

My problem is twofold: I'm both short of time and a bit lazy. I was ecstatic to discover something called an "authoring" system.

"I'm both short of time and a bit lazy."

Writing Computer Aided Instruction (CAI) in Basic is like using Basic for a word processor. It is amazing how many otherwise smart folk are writing their CAI lessons the hard way. To paraphrase a well-known quotation: Had I but two computer programs, I would sell one and buy an authoring system to feed my efficiency.

Back in May 1981, Education 80 contained an explanation of CAIWARE and Super-CAI, an authoring system from MicroGnome Division of Fireside Computing Inc. (5843 Montgomery Road, Elkridge, MD 21227). The system takes care of programming mechanics—the nuts and bolts. You need no programming experience; even if you can program, you don't have to take the time to do so. You provide the subject matter and the plan of presentation—the system takes care of the rest. Your program can be linear or forward-branching with criteria you establish. The ease and versatility for you (and the student) have to be experienced to be believed.

Do not misunderstand: No authoring system does *all* of the work. You still have to plan the lesson, unit, or whatever. But it is the same type of planning you are doing now.

Remember, no computer (even with an authoring system) is a teacher, one reason a computer will never replace you.

With the increased use of disk operation, I call to your attention CAIWARE's big brother, CAIWARE-2D. This product, though compatible with earlier taped lessons, is a giant step beyond CAIWARE. It provides you with many more features and options, especially when using disks. To name just a few, there are graphics, choice of tutorial or evaluation modes (or both), an additional form of branching, and student record keeping by session, lesson and question. If you use or plan to use disk with your CAI, the additional cost of CAIWARE-2D is a very good investment. If you are already using CAIWARE or Super-CAI, you will receive credit for its original price when you upgrade.

The folk at MicroGnome are continually upgrading and improving an already good system. A small fee will automatically bring you updates as they are released.

If you don't have an authoring system, I suggest writing to the MicroGnome himself, Don Coyne. Perhaps he can furnish details convincing you to upgrade your system.

TRSDOS Patches

I'm in favor of progress as much as the next person. When TRSDOS 1.3 came along, it was great even though it meant work. Dozens of program disks had to be changed over.

All that changing is not essential if you and your students remember to press Reset each time a disk is inserted. Forgetful as we are, however, that leads to frustration and wasted time. My preference is to change them over.

Everything was going fine until Radio Shack came along with a series of patches to improve performance further (*Microcomputer News*, Nov 81). We simply could not see typing all those patches anew for each disk. There had to be a better way—and there was.

We stumbled upon an accurate and quick patching procedure. (I say "stumbled upon" but we prefer to say designed or deduced or something like that.) Serendipity versus hard cold reason aside, we tested our method on spare disks and plunged ahead.

Since Patch commands will not function in a Basic program, we used the Build command under TRSDOS to create the program. Here is the procedure to save retyping patches, whether it's one or a dozen.

Prepare the Build program following the directions in the manual. Each patch is placed on a separate line; proofread carefully before they are entered. When the program is completed and you are back to the TRSDOS Ready prompt, type in: LIST PATCH/BLD (ASCII,SLOW).

As you see, we named our program "Patch." With the listing on the screen, proofread it a second time. Remember, a mistake in a patch can cause all kinds of problems.

When you are sure the patches are error-free, enter the command: DO PATCH/BLD. TRSDOS lists each patch and tells you it is made. With that disk patched, you are ready to patch others containing TRSDOS.

If you have two drives, place the disk with the PATCH/BLD program in Drive 1 and the disk to be patched in Drive 0. The Do PATCH/BLD command will function as before. Simply repeat the process until all of your 1.3 disks are current versions.

With one drive, you will have to copy the PATCH/BLD program to each successive disk. "Do" it and then kill it to conserve space.

This patching procedure increases the accuracy and decreases the time of making even one patch to several disks.

Reprints

Do you know that the first issues of the *TRS-80 Microcomputer News* are available? You can complete your set of the *News* by visiting your local Radio Shack store; $4.95 buys you a bound volume of the first 20 issues (through Dec 1980). The early ones contained a lot of ads but there is good stuff, too.■

MONEY DOS

an operating system for financial survival

by J.M. Keynes

"Deal with things like they is, not like they ain't."

Will Rogers once said, "Our problem is not ignorance. It's all the things we know that ain't so." The country philosopher never heard of a computer, but the quotation expresses insight into everyone's personal computer—his brain.

Consider this: We do most of what we do exactly as we have always done it, and we do it unconsciously. We store thousands of programs in our brain which dictate our responses to stimuli. Our data banks remain constant and should one access them he could *predict* our behavior.

The longer a program resides in our memory, the more difficult it is to modify. For example, you have a 30 year old program which says, "My mother loves me." Mom would have to treat you very badly for you to modify the program to read "My mother does not love me."

Thousands of investors who owned IBM stock in the fifties and sixties loaded a program which said, "IBM is a super growth stock." The fact is that IBM stopped being a growth stock ten years ago. It is worth less today than in 1971 but prices have tripled.

IBM was indeed a super growth stock until the early 70s when disaster struck. It had billions of dollars worth of giant computers leased out when one day salesmen began showing IBM's leesees their new product—a small box that did everything the huge "white elephants" did, did it faster and did it cheaper. IBM's customers could own the new small computer for what they paid to lease the IBM behemoth. My junk box is filled with circuit boards from those outmoded computers with their discrete components. All this failed to trouble all those loyal stockholders who throughout the 70s were still convinced (against all reason) that IBM was a growth stock. Many still are.

An interesting feature of successful people is their ability to modify programs when clear evidence indicates they should. Unsuccessful people, like the dinosaurs, are unable or unwilling to adjust to changing conditions.

What are some of the things we know that "ain't so?" Over 30 years ago I was sitting on the front porch of Grandpa's house in Waco, TX. He filled his pipe with Prince Albert smoking tobacco and offered a young man some sage advice. He said, "If you work hard and save your money in the bank, someday you can retire and live a carefree life just like I do." Such advice had worked for his forebears and he saw no reason to think it would fail his progeny. The advice was sound and worked for many years. Thereupon I saved a program which withstood time for 20 years.

". . . you throw money away every day."

Grandpa predicated his advice upon a belief that if one could increase the money in his savings account each year, all would be well. He was right during all those years when inflation was ½ percent. Any banker would have agreed with Grandpa. Many still do.

Until recently, the expression "purchasing power" was seldom used. Today if one evaluates investments solely on the basis of dollar increase, he is courting financial disaster. How much money you make from an investment is irrelevant. The only useful criteria today is, "How many loaves of bread will it buy?" compared to last year.

Another of Grandpa's gems was, "If you want to be a success, learn to deal with things like they is, not like they ain't." That advice will be good forever.

The Program Listing requires one entry, an estimate of the inflation rate. The results of fixed income investment will change depending on what happens to the inflation rate. If it declines you are better off and vice versa.

In 1975 I spoke of the spectre of a $1 loaf of bread by 1980. Many chuckled. Today I speak of the $3–$5 loaf by 1990 and no one laughs anymore. I hope a miracle happens but if you had to bet $100 on whether a loaf of bread would be more or less expensive one year hence, how would you bet?

Inflation has forced us all to take risks. There is no "risk free" investment which you can count on to return more after tax purchasing power than what you started with. Run any guaranteed investment you like in the program and all return more money but less purchasing power. The six percent savings account is a certain path to poverty. There are ways to increase investment return without increasing the risk.

Increase Your Bankroll

If you have $1000 or more in a demand savings account yielding six percent, you may, with little more than a stroke of the pen, increase your yield by 5–13 percent depending on conditions. If you have not yet discovered money market funds you throw money away every day. They are instantly liquid just like your demand account at the bank or savings and loan. Every major stock brokerage firm has one as well as many independent mutual fund companies. They are safe, liquid, and "no load" which means you pay no fat commission to the salesman. The *Wall Street Journal* publishes many ads daily. Call one of the toll free numbers and get a free prospectus for the details. Some even offer free check writing privileges. If you can earn an extra eight percent on just a $1,000 investment that is another $80 per year. For free. Are you so rich that you can pass up a nickel lying on the sidewalk?

Thousands of you have life insurance policies with cash value. Read the fine print and you will find that you can borrow that money at five or six percent. An old program probably says, "Never borrow on life insurance except in emergencies." Nonsense. You have been loansharked for years by those who borrow low and lend high. Why not do a little loansharking yourself? Borrow up to the hilt at 5–6 percent and loan it out at 14–16 percent to a moneymarket fund. Millions of people are missing this easy, risk-free way to make money (much to the delight of insurance companies who like loaning out your money at 17 percent).

If you are about to invest $25,000 or

more in a short term CD, check the rates on commercial paper. Commercial paper is a promissory note issued by a company for a term of 1-270 days. It is only as safe as the issuing company. Within the last few years a number of times Moody's has rated commercial paper "prime #1" (the highest rating). It yielded 3 or 4 percent more than CDs. Your banker or broker can quote the rates every business day. The commission is small. A $100,000 transaction costs $25. Your banker is familiar with commercial paper although he may have failed to mention it to you.

If you are about to purchase some common stock, ask your broker to determine if there are any convertible bonds or preferred stock issued by the company. Sometimes you can invest in a convertible and have all of the upside potential of the common stock, yet bear only 10-50 percent of the risk. Here is an example: In April of 1981 a client called with an order to buy 4000 shares of Louisiana Land Offshore Exploration (an offshoot of the giant Louisiana Land and Exploration). The price of the stock was $12.50. I checked for a convertible. The company had a four percent bond which was convertible into 80 shares of the common stock.

> *"Examine the arithmetic."*

Examine the arithmetic. The bond was selling for only $5 over parity (what it was worth in common stock 80 times 12.5). If the stock went to $20 per share, the bond would be worth 20 times 80 or $1,600, the same profit you would realize from owning 80 shares of the common stock. If, on the other hand, on October 1, 1982 the common was down to $6 per share, the bond holder would get his money back (the bond matured) while the common stock holder would suffer over a 50 percent loss.

Further, the bond holder would collect about $75 in interest payments while the common stockholder collected nothing as the company pays no dividends. The commission on 4,000 shares of common at $12.50 per share would have been over $500. The commission on 50 bonds was $250.

As this is written, the stock (over the counter symbol is "LLOE") is quoted at about $7 per share. Instead of worrying about a current loss of over $20,000 on the common, my client knows he will get all his money back on October 1, 1982. The stock *may* rise in price to over $12.50 per share in the meantime and he can cash in for a profit. This is not a story of a big killing, but was it Ben Franklin who said, "A penny saved . . ."

In the future I will badger you about examining your alternatives before you invest. "How can I examine alternatives if I do not know about them?", you ask. Just keep up your subscription and I will show you many alternatives. I may tell you of a stock currently selling for $19 per share that, if the company was liquidated tomorrow, would fetch over $100 per share. I do not support my two million dollar home and expensive wife by writing articles. ■

```
10 CLEAR: CLEAR1000: CLS
20 U$="#.##": T$="####.##": S$="###.##": V$="##.#": X$="######.##": Z$="###
###"
10000 CLS: PRINTCHR$(23)"WILL ROGERS ONCE SAID,": PRINT"OUR PROBLEM IS NOT
IGNORANCE": PRINT"IT'S ALL THE THINGS WE KNOW": PRINT"THAT AIN'T SO!
10010 PRINT: PRINT"WE ALL KNOW THAT A 10 YEAR TAX  EXEMPT BOND YIELDING 9%
IS": PRINT"A PRUDENT INVESTMENT,OR PERHAPS A 20 TREASURY BOND": PRINT"YIELD
ING 11% ???
10020 PRINT: PRINT: PRINT"TO EXAMINE THE RESULTS OF THESE OR ANY OTHER INVE
STMENT": INPUT"PRESS 'ENTER'";PO
10030 CLS: PRINT"THIS PROGRAM WILL ALLOW YOU TO DETERMINE THE ACTUAL": PRIN
T"RESULTS OF YOUR INVESTMENT IN TERMS OF PURCHASING POWER.
10040 PRINT: PRINT"FOR SIMPLICITY, WE SHALL ASSUME A GALLON OF GAS NOW COST
S": PRINT"$1.00 AND WILL INCREASE IN COST AT THE ESTIMATED RATE OF": PRINT"
INFLATION.
10050 PRINT: PRINT"IT IS FURTHER ASSUMED THAT YOUR YEARLY DIVIDENDS, INTERE
ST,": PRINT"ETC. IS RE-INVESTED AT THE RATE OF ORIGINAL INVESTMENT.
10060 PRINT: PRINT"NOTE:  AFTER EACH ENTRY, YOU MUST PRESS 'ENTER'
10070 PRINT
10080 INPUT"AMOUNT OF YOUR INVESTMENT IN $.";A
10090 INPUT"NUMBER OF YEARS?";Y
10100 INPUT"YEARLY RETURN 12% IS ENTERED AS 12";R
10110 INPUT"YOUR INCOME TAX BRACKET 40% =40";TB: T=(100-TB)/100
10120 RR=(R*T)/100+1
10130 INPUT"ESTIMATED INFLATION RATE IN %";B: BB=B/100+1
10140 C=R*T: Z=1: Q=A
10150 FOR I=1TOY
10160 A=A*RR
10170 Z=Z*BB
10180 NEXT: I=I-1
10190 X=A/Z: W=X/Q: V=(1-W)*100
10200 CLS: FORF=1TO1000: NEXT
10210 PRINT: PRINT"TODAY YOUR $";Q;" INVESTMENT BUYS ";Q;" GALLONS OF GAS.
10220 PRINT: FORF=1TO1000: NEXT: PRINT"IN ";I;" YEARS YOUR INVESTMENT WILL
HAVE GROWN TO $";: PRINTUSINGX$;A
10230 PRINT: FORF=1TO1500: NEXT: PRINT"IT SEEMS THAT YOU HAVE DONE WELL UNT
IL YOU BUY SOME GAS....": FORF=1TO1500: NEXT: PRINT: PRINT"WHICH WILL BE $"
;: PRINTUSINGS$;Z;: PRINT" PER GALLON.
10240 FORI=1TO1500: NEXT: PRINT: PRINT"NOW YOU CAN BUY ONLY ";: PRINTUSINGZ
$;X;: PRINT" GALLONS OF GAS.
10250 FORI=1TO1500: NEXT: PRINT: PRINT"WHICH MEANS YOU WILL LOSE ";: PRINTU
SINGV$;V;: PRINT" % OF YOUR PURCHASING POWER!
10260 TR=B/T: PRINT"YOU MUST INVEST YOUR MONEY AT ";: PRINTUSINGV$;TR;: PRI
NT"% TO BREAK EVEN.
10270 PRINT: PRINT: INPUT"TO EXAMINE ANOTHER INVESTMENT, PRESS 'ENTER'. ";L
L: GOTO10030
```

Program Listing

INPUT/OUTPUT

By James E. Keogh

Send any questions or problems dealing with any area of microcomputers to Input/Output, 80 Microcomputing, *80 Pine Street, Peterborough, NH 03458.*

I am looking for a book or a series of articles on sorting algorithms (i.e. bucket sort, bubble sort, binary tree sort, shell sort). I do not want a set of language specific programs but rather general theory and explanation to be used in designing specific programs. In particular I am interested in sorting records by a given field in each record. I am limited to a Z80 running either Basic, Assembly or Fortran.

L.N.
Riberalta, Beni. Bolivia

I searched several magazine back issues for articles dealing with sorting algorithms with similar results. Your best bet is to find software using one or more of these programming techniques. Contact the author of the software and ask if he can help you.

I have a TRS-80 and am planning modifications. Will the warranty be violated if someone other than a Radio Shack employee repairs the unit? I have had the unit for more than a year.

A.E.
Savannah, GA

Radio Shack warranties do not extend beyond 90 days. If a continuing problem was evident during the first 90 days, Radio Shack will continue to work on the problem as if it was still under warranty. All repairs made by Radio Shack carry a 30 day warranty. Unless this applies to you, your warranty has already expired. A word of warning! Be sure you know what you are doing before attempting any modifications.

This may sound like a strange problem but recently I purchased Radio Shack's Space Warp software. One of the ways to fight off the space invaders is to use the lasers. This allows you to fire as many times as you need without running out of ammunition. According to the instructions you are to set the angle of the lasers, setting the direction of fire, and then press number five to fire the lasers. By the time I determine and set the angle the space invaders have shot me three times and I lose the game. How can anyone fire in such a short time?

M.K.
Lakeville, MN

You do have a problem! Firing the lasers before you get fired on is the fun of playing Space Warp. Some of our readers have had a similar problem but have developed a trick to out smart the invaders. Defend your planets by firing the lasers without pre-selecting a firing angle. Just press the number five. The lasers fire at zero degrees or directly to the right of your space ship. Some "commanders" have found it easier to quickly position the space ship so the invaders are directly to the right of it. All you have to do is fire.

I want to design a program using a fast graphics display. I read this is possible using assembly language. My problem is I just got my feet wet with Basic. I do not want to start all over again. Do you have any suggestions?

R.T.
New Bedford, MA

Tackling a new computer language can seem like a nightmare. If you have a TRS-80 Level II you may be able to accomplish faster displays by using the POKE and PEEK commands. For a better understanding of these commands consult the Level II Basic Reference Manual.

I am about to convert from cassette to a disk drive system for my TRS-80. Can I use my tape programs with a disk system?

E.H.
New Haven, CT

The best thing to do is to rekey in your programs and then store them on disk.

I am responsible for the administrative functions of several doctors. Recently I purchased a TRS-80 disk system. Are there any TRS-80 disk programs available to handle administrative billing and record keeping for doctors?

B.W.
Englewood Cliffs, NJ

There is bound to be such a program in existence. The problem is locating it. Many small software companies offer programs designed for specific professions. There are several things you can do to obtain the program you need. Check a software source catalogue available at some Radio Shack stores. Also check the ads in computer magazines. If you still cannot find the exact software, consider purchasing existing software designed for general business use and find a local programmer to modify the program for your needs.

Like many computer hobbyists I want to make money using my TRS-80. I decided to design programs for school systems to be used in the classroom. What computers are commonly used by school systems for classroom use?

R.S.
Fort Worth, TX

The school market for classroom microcomputers is growing. Although I have not seen any studies, a major educational publishing company also entering this market is preparing programs for the TRS-80 Model III 32K with dual floppies; Apple II Plus 48K DOS with dual floppies; Atari 800 16K with single floppy and cassette.

I recently purchased the Radio Shack Micro Movie software package. It is just great! I have ideas for programs using many of its techniques. Is it possible for me to design a program around Micro Movie? I thought I would load my program in first, then the Micro Movie program and then load both on a single tape. Can this be done?

B.W.
Sidney, OH

Hold on a second! First, the Micro Movie software cannot be used without permission from the authors. You will quickly find yourself in legal trouble using software without permission. Your sec-

✓See List of Advertisers on page 290

INPUT/OUTPUT

ond problem will be with your program. Unless you write it in assembly language you will lose the program as soon as you load the Micro Movie software.

I bought my TRS-80 about six months ago and am beginning to develop good programs. My problem is when I type in Run, the word Ready appears on my program. Is there something wrong with my TRS-80?

P.F.
Vicksburg, MS

There is nothing wrong with your microcomputer but there is a small problem with your program. Every program should begin with the instruction to clear the screen (CLS). This statement will remove anything on the screen prior to running the program.

I enjoy Radio Shack's chess software. I have several friends I play chess through the mail with. What I would like to do is print out the chess board and my move from the program. Can I modify this program?

R.S.
Madison, WI

For all practical purposes the answer is no. The program is not designed for printing the board and moves.

I intend to purchase a disk drive for my microcomputer. The salesman suggested two disk drives are better than one and that I should not buy any unless I get both drives. Is he right?

C.E.
Decatur, IL

Two disk drives are better than one only if you need the extra capacity. For business you should seriously consider a two drive system. For personal use a single drive will probably do the job. Later if you feel you need more capacity you can always add another drive unit.

I want to learn Basic in a formal setting. My local Radio Shack Computer Store offers a course as does a local college. Which one do you suggest?

M.R.
Poughkeepsie, NY

The answer depends upon the type of computer you intend to program. Basic computer language is used on mini's, micro's and some mainframes. The course offered by Radio Shack will teach you Basic as it is used for a TRS-80. You will need additional training in Basic for Apple and other computers. The college, on the other hand, may not be offering a Basic course for the TRS-80. Research both courses before deciding.

I am forever hearing about owners of microcomputers ordering products using their computer's screen without leaving home. I have never seen such a thing. Is it available?

T.B.
Catello, ID

There is some confusion about ordering goods from home via a computer. The system you describe is being tested for the cable television market. The cable company sends a signal over the cable to your home television showing you goods. Using something similar to your microcomputer, you would then send a signal back to the cable company's computer to process your order. It is still a ways in the future. A few mail order computer companies have an arrangement whereby you can use your TRS-80 or any microcomputer with a modem to order products, but you do not see the products on your screen.

Soon after purchasing my TRS-80 I bought a Microline printer. If I hook this printer up to my TRS-80 will this make my 90 day warranty invalid?

A.K.
Austin, TX

That is a sticky question. Technically if you have problems not connected nor caused by the Microline printer, Radio Shack should honor the warranty, although the burden of proof may be on you. TRS-80 owners just bring the key board, expansion interface and cables to their Radio Shack store. The technician then checks the equipment without the printer.

My friend has an Apple II. Can I use his Basic programs on my TRS-80?

Y.T.
Charleston, N.C.

It is not that easy. Each manufacturer makes slight changes in Basic which make interchanging programs impossible. You can take a listing of the program and convert the program for use with the TRS-80.

I have heard a lot about word processors. For about $1,000 you can buy a TRS-80 microcomputer with word processing software. You can also buy word processors for upwards of $15,000. What is the difference?

T.K.
Medford, OR

They are all basically the same. The higher price does not necessarily represent higher quality. In certain circumstances you will have more capacity with a TRS-80 than with a $15,000 word processor. Look for 80 Microcomputing's *word processor issue coming in July 1982.*

In a few months I will be located in a city where only 220 VAC 50 cycle current is available. On a recent visit I tested my TRS-80 using a 220–110 transformer. The computer performed flawlessly except for video ripple. Is there a simple solution to this problem? Where can I get a copy of the schematic for the monitor? My computer was purchased in the United States.

L.N.
Riberalta, Beni, Bolivia

Your problem is because of the higher current load on the power supply. The best thing to do is to purchase the power supply for the Model I and expansion interface in the country you are in. These power supplies are specifically designed for 220 volts 50 cycles. As for the schematic, you need service manual 26-1201. It costs $1.49 plus $1.50 shipping and handling. Write to:
National Parts
900 East Northside Drive
Fort Worth, TX 76102 U.S.A.
Attention: Order Desk

I recently received a catalogue of surplus electronic and microcomputer parts. Is it possible to build a microcomputer from these parts?

R.B.
Marion, OH

Sure, if you know what you are doing! But before you start ordering various components you must know what components you will need to build your own computer. Then you will have to obtain all the specifications for the available components to find out if they are compatible to each other. If you are not sure which components to purchase ask the advice of a microcomputer technician or engineer.

I have been trying to decide if I should purchase a personal computer. I have tested a few computers in stores but would feel more comfortable trying one at home for a few weeks before buying. Can computers be rented?

W.G.
Chicago, IL

You will have to find someone willing to rent one to you. Most stores, including many Radio Shack stores, have a policy of not renting equipment to customers. Many stores do let customers try the equipment in the store for as long as the customer wants during normal store hours.

✓See List of Advertisers on page 290

Copernica Mathematica

by Bruce Douglass

Approximation Theory deals with finding easily computed functions which approximate more obscure functions. It has two major concerns: Does the approximating function converge to the real function, and how quickly does the approximating function converge?

SIN, COS, LOG, ATAN, SQR, and EXP are typical mathematical functions. Interestingly, none of these functions can be easily figured on a computer. A computer can directly perform logic functions (AND, OR) and two arithmetic functions (ADD, SUB) on 16 bit integers. All other operations are in terms of these simple functions. Non-integer functions, such as SQR, may be calculated only as approximations.

The Rational and the Irrational

Recall that numbers are represented in floating point format, and there is a limit on the accuracy of the numbers we use on the computer. Ask your computer what ⅓ is, and you will get 0.3333333. Double precision adds to the number of threes, but nothing the computer produces really equals ⅓. The computer gives an approximate value for non-integers.

You can get inside the computer via assembly language and improve the precision. The gain in accuracy is offset by the loss in time and memory utilization. Regardless of how much you improve accuracy, the same basic restriction remains: You are approximating non-integers. You'll never quite reach true non-integral values.

Well, that's not exactly true. You can use non-floating point number representation for rational numbers. MuMATH, in fact, does this. If you ask MuMATH what ⅓ is, it will tell you ⅓. But what about irrational numbers? An irrational number, such as SQR (2), cannot be represented by a fraction of integers. MuMATH will tell you its value is SQR (2); exact, but not useful if you want a floating point.

The only way to work with rational and irrational numbers and functions is to approximate them.

Level II Basic has certain function approximations built in. Ask for SIN(50) and it will give you a single precision number. If the accuracy isn't great enough, there is some way a home micro can tell the typical lay person SIN(X) to 12 or 16 decimal places. You need only an algorithm and a routine. First, let's consider some theory.

A Bit of Calculus

A function's derivative is merely the value of the slope of the curve defined by that function. A straight line has a constant slope; its derivative will always be a constant. A non-linear function (one that is not a straight line) has, by definition, a variable slope. That is, the slope of the line tangent to the curve depends on the part of the curve you are considering. F (X) = X↑2 has a slope of 2*X. The slope of its tangent line varies depending on where you are (the X value).

The slope of the tangent line is the rate of change of the function per unit change in the variable. If F(X) is the function, the derivative is usually written F'(X).

A second derivative (written F''(x)) is the slope of the line tangent to the tangent curve of the function F(X). In other words, whereas the first derivative of F(X) is the rate or change of F(X), the second derivative is the rate of change of the rate of change. Most functions are infinitely differentiable: You may take any order of derivatives of the function as you like.

In the case of the line defined by F(X) = 2*X + 3, the rate of the change of F(X) is 2. Every time you change X by one, F(X) will change by 2. The rate of change of F'(X) (the second derivative) is zero. That means the amount of change in F(X) per change in X remains constant.

For F(X) = X↑2, the rate of change F'(X) equals 2*X, which is not a constant. Since the value of the derivative F'(X) depends on where you are, the rate of change of F'(X) (F''(X)) is not zero (it is, in fact, equal to 2).

As you might imagine, you can take derivatives until the cows come home, although they might equal zero after a while. Some functions' derivatives never become zero, regardless of how many derivatives you take. SIN(X), SQR(X), and EXP(X) are examples of this kind of function.

This superficial look at differential calculus should give the uninitiated, qualitative sense of what derivatives are and what they mean. But you need not know calculus to implement these approximations on your computer. You can find algorithms in reference books and write efficient approximation routines without understanding the higher mathematics.

The rest of this month's column explains how to find an algorithm, how to

```
F(X)    =    F(A) + (X - A)*F'(A) + (X - A)↑2*F''(A)/2!
             + (X - A)↑3*F'''(A) + ... + (X - A)↑n*Fn(A)/N!
             + ...
where
             F(A) is the function F(X) evaluated at X = A
             Fn(A) is the nth derivative of F(X) evaluated
             at X = A
             N! = N factorial = N*(N - 1)*(N - 2)*...*2*1
             C↑N = C to the Nth power
```

Fig. 1

Example: Angle in radians
Actual value: COS(1) = 0.54030 23058 681397
Approx value: COS(1) = 0.54030 23062 ......

Terms	Value	Error
2	0.5	4*10↑-2
4	0.54166 66666 666667	1*10↑-3
6	0.54027 77777 777778	2*10↑-5
8	0.54030 25794 365079	2*10↑-7
10	0.54030 23037 918871	2*10↑-9
12	0.54030 23058 795628	1*10↑-11
14	0.54030 23058 681399	2*10↑-16
16	0.54030 23058 681397	<10↑-16
18	no increase in accuracy with double precision	

Fig. 2

See List of Advertisers on page 290

write a routine implementing the algorithm, and how to watch for (and tiptoe around) violations in algorithm constraints.

Taylor's Series

Taylor's series (see Fig. 1) is an infinite series that converges to the value of the function in question. It presumes that the function has infinitely many derivatives, not unreasonable for most functions. A function infinitely differentiable over an interval may be written as the sum of the terms in a Taylor series. Translated, if you can take the derivative of a function as many times as you want (without limit), a certain series converges to the value of that function when you add its terms. The only catch is that the series has infinitely many terms. The theorem says nothing about how fast the series will converge, only that it eventually will.

We've used a term that we haven't defined. If function F *converges* to function G, and F is an infinite series, then as we take more and more terms of F, the value becomes closer and closer to G. To put it mathematically,

$$\lim_{n \to \infty} (F_n) = G$$

as n, (the number of terms taken) approaches infinity, the value of F approaches the value of G. If you take a finite number of terms (10 or 10,000,000), you can still get closer by adding the value of the next term to F. That is what is meant by an asymptotic approximation.

Taylor's series converges to the value of the function at X. The MacLaurin series is Taylor's series where A equals zero.

You may approximate a function by truncating the series, i.e. by taking a finite number of terms. Since the series converges, the value of each term becomes less. By taking the difference between two successive terms of the series, we can compare the accuracy of our finite series with a predetermined value, and get the level of accuracy we want.

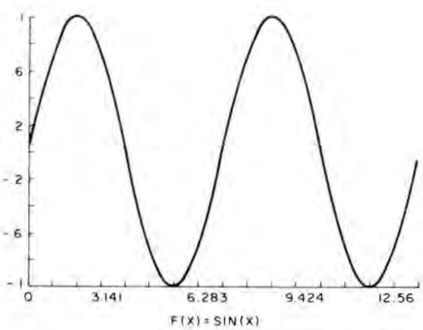


Fig. 3

First Find an Algorithm

Designing algorithms is part cold mathematical reasoning and part inspiration from the Cosmic Muffin. A good algorithm is most often not a truncated Taylor series, although that is one way to get an approximation. Good series approximations are often dependent on the number of terms for their coefficients. Three good sources of approximating algorithms are *Aproximation for Digital Computers* by Hasings, *Handbook of Mathematical Functions* by Abramowitz and Stegun (Dover Books), and the recently released *Basic Scientific Subroutines Vol. I* by Ruckdeschel (Byte Books).

Let's look at an example. The Taylor series approximation for COS(X) is:

COS(X) = 1-x↑2/2! + X↑4/4! -X↑6/6! + X↑8/8! + ...
↑ means "raise to a power"
! means "factorial"
Function is expanded around the point A = 0
(MacLaurin series)

Let's shoot for an accuracy of about 10^{-9}. Our approximation will differ from the real value of COS(X) by only about 10^{-9}. The above approximation would take many terms because the coefficients (1/i!) are not optimal for a finite series approximation. The series was designed to converge at infinity. By linear regression we can determine better coefficients for this approximation. We can, in fact, find the best coefficients for this particular number of terms.

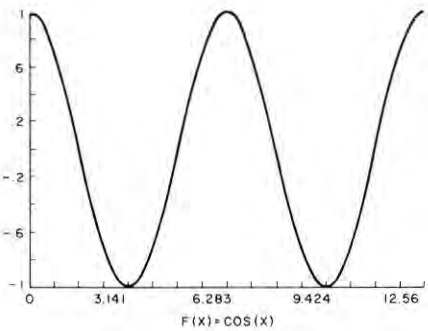


Fig. 4

```
10 REM COSINE APPROXIMATION
       ACCURATE TO 2*10[(-9)
20 DEFDBL A-K:DEFINT S,Z:CLS
30 A=-.49999 99963:B=.04166 66418:S=1
   C=-.00138 88397:D=.00002 47609:
   E=-.00000 02605:G=1.5707 6327:
   K=0.07174 53292 51994:G1=6;28318 5307
   G2=3.1415 92564
40 INPUT"ANGLE IN RADIANS (1) OR DEGREES (2)";Z
50 INPUT"ENTER ANGLE";H
60 IF Z=2 THEN H=H*K REM CHANGE TO RADIANS
70 IF H>G1 THEN H=H-G1:GOTO70
80 IF H>G THEN H=G2-H:S=-S:GOTO80
90 J=1+H*H*(A+H*H*(B+H*H*(C+H*H*(D+H*H*E))))
100 PRINT "COS(";H;") RADIANS =";J*S
110 GOTO40
```

Program Listing 1

If we decide we need greater accuracy, and so add terms, we must recalculate all coefficients. We have altered the contour of the approximation in hyperspace (haven't you always wanted to use that word in an article?).

In our COS example, we can get our error to be less than 2 * 10^{-9} with only six terms!

0 <= X <= PI/2 (RADIANS)
COS(X) = 1 + A*X↑2 + B *X↑4 + C*X↑6 +
D*X↑8 + E*X↑10
A = -.49999 99963
B = .04166 66418
C = -.00138 88397
D = .00002 47609
E = -.00000 02605
ERROR <2*10↑-9

Compare this with the truncated Taylor's series in Fig. 2. The optimal coefficients do not differ much from those in the Taylor series itself. As the desired accuracy increases (and acceptable error becomes smaller), the difference between optimal and Taylor series coefficients becomes more important. Notice in Fig. 2 that it takes 10 terms to get the same accuracy using a Taylor series as with the above approximation using six terms.

Dodge Those Constraints!

This is only accurate for a certain range in X values. Also, X is in radians; some backward parts of the world use

```
10 REM ROUTINE TO CALCULATE F RATIO
       ACCURATE TO 10[-4
20 REM V1=DEGREES OF FREEDOM (NUMERATOR)
       V2=DEGREES OF FREEDOM (DENOMINATOR)
       F=VALUE OF F RATIO
30 C1=.196854:C2=.115194
   C3=.000344:C4=.019527
40 INPUT"ENTER VALUE OF F";F:
   INPUT"ENTER NUMERATORS DF";V1:
   INPUT"ENTER DENOMINATORS DF";V2
50 A=2/9/V2:B=2/9/V1
60 X=F[(1/3)*(1-A)-1+B
70 X=X/SQR(B+F[(2/3)*A)
80 C=1+X*(C1+X*(C2+X*(C3+X*C4)))
90 P=1-1/2/C[(-4)
100 PRINT"P(F) WITH ";V1;"AND";V2;
     "DEGREES OF FREEDOM WHERE F =";F
110 PRINT"IS  ";P
120 END
```

Program Listing 2

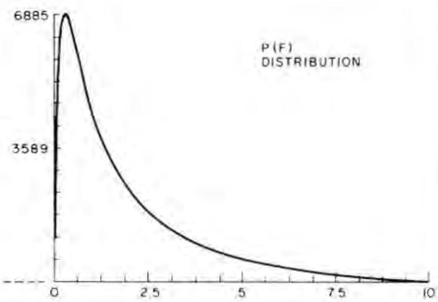


Fig. 5

✓See List of Advertisers on page 290

degrees for measuring angles.

The latter problem is trivial: Just multiply degrees by a constant (K = 0.0174 53292 519943) to convert to radians (note that the constraint on X is changed to 0 <= X <= 90 in degrees). The former problem is simple, but perhaps not so obvious.

If you know what a sine wave is, then you know that the SIN function undulates with a certain frequency, or periodicity. If we move the graph of the SIN function (see Fig. 3) a certain amount, the value of SIN is exactly the same. The amount we move is called the period of the SIN function.

The COS function is the SIN function moved over, or displaced, along the X axis (Fig. 4). COS is, likewise, periodic. The period for SIN and COS is 2*PI. If X is greater than 2*PI, we can subtract 2*PI from X without changing the value of COS(X). But what if X is between PI/2 and 2*PI? Now we must use other facts about the COS function. COS is an even function: COS(X) = COS(-X). Also, COS(X) = -COS(PI-X). If X is negative, we can change the value of X to ABS(X) without changing the value of COS(X). If X is greater than PI radians (or 180 degrees), we can subtract PI (or 180 degrees) from X and get -COS(X).

Now we know enough to write an algorithm that will give us COS(X) to 10^{-9}. We have the basic algorithm, and know how to sidestep the series constraints by using unique properties of the cosine function itself.

Program Listing 1 illustrates a number of points. First, notice that all pertinent variables must be double precision and other variables are defined as integers. Second, all coefficients are defined before the actual cosine calculation. Since the Basic interpreter must change the ASCII characters into floating point numbers, it is much faster to define them once. Then the program must only call the variables, already in floating point format.

Notice that no exponents appear in line 90. An alternate way of writing line 90 is;

```
90 J = 1 + A*H↑2 + B*H↑4 + C*H↑6 + D*H↑8 + E*H↑10
```

Why use the longer way? First, the exponential function in the Level II interpreter is single precision, and gives results only as accurate as the exponential function itself. Second, this method is faster. The number of floating point multiplications is reduced greatly by nesting parentheses. Try writing it using H*H*H*H for H↑4 and count the number of operations. The savings can be quite significant.

The integer sign variable S enables us to keep track of the COS after the angle reductions in lines 70 and 80.

Remember when you write your own routines to use adequate precision, watch out for violations of constraints, and nest those parentheses!

Rational Polynomial Approximation

Let's look at a different example. Inferential statistics uses the F ratio. The F ratio is a measure of variance between groups divided by an independent measure of variance within the groups.

If the F ratio is greater than one, there is more variance (difference) between the groups than there is within the groups. Thus the groups are different. If the F ratio equals one, the groups are the same. It is a question to be solved (shortly) how different from 1 greater than 1 really is. It has to do with the size of the groups (actually degrees of freedom, for you pedants out there).

The distribution of the F ratio (see Fig. 5) is a function of the value of F and the degrees of freedom within (how many per group), and the degrees of freedom between (how many groups). The area of the graph from zero to F for a particular curve gives the probability that the F ratio is different from one, i.e. the chance that the groups are really different. Therefore the total area under the curve equals one.

This could be done by numerical integration, but functions of more than one variable don't usually integrate very nicely. The formula for the area of the F ratio is given below:

$$P(F) = \left[\frac{\int_0^\infty e^{-x} x^{\left(\frac{m+n}{2}-1\right)}dx}{\int_0^\infty e^{-x} x^{\left(\frac{m}{2}-1\right)}dx \int_0^\infty e^{-x} x^{\left(\frac{n}{2}-1\right)}dx}\right] \int_0^F m^{\frac{m}{2}} n^{\frac{n}{2}} x^{\left(\frac{m}{2}-1\right)} (n+mx)^{-\left(\frac{m+n}{2}\right)}dx$$

In the *Handbook of Mathematical Functions* we find just such an approximation, namely,

```
P(X) = 1-1/2*(1 + c1*X + c2*X↑2 + c3*X↑3 + c4*X↑4)↑-4
c1 = .196854
c2 = .115194
c3 = .000344
c4 = .019527
error < 2.5 * 10↑-4
```

This is an example of a rational polynomial approximation. That is, it is a ratio of polynomials: The numerator is a trivial polynomial (i), but the denominator is a fourth order polynomial raised to the fourth power.

We also need to find a single number X from the three variables. The same source gives it:

$$X = \frac{F↑(1/3)*(1 - 2/(9*v2)) - (1 - 2/(9*v1))}{SQR(2/(9*v1) + F↑(2/3)*2/(9*v2))}$$

where

F = value of the F ratio
v1 = degrees of freedom (numerator)
v2 = degrees of freedom (denominator)

A rather ominous looking problem, this beastie is easy to tame. First, we may use exponents and SQR functions since the accuracy of our approximation (10↑-4) is less than the accuracy of these functions (10↑6). To put it another way, our error is greater with the approximation of the F ratio; it is the limiting function in terms of accuracy.

Program Listing 2 approximates P(F), the area under the curve from zero to F.

As you can see, the constants were defined before the execution of the approximation, and parentheses were nested. This short simple program is good approximation of the rather ominous F distribution.

We have briefly looked at why we should want to make approximations, what an algorithm is, how to get around some constraints, and how to change the approximation algorithm into an approximation routine in Basic. While this is not all there is to approximating complex functions, I hope you use these techniques fearlessly in your programming repertoire. ■

Centronics 730

Same as Radio Shack
Line Printer II

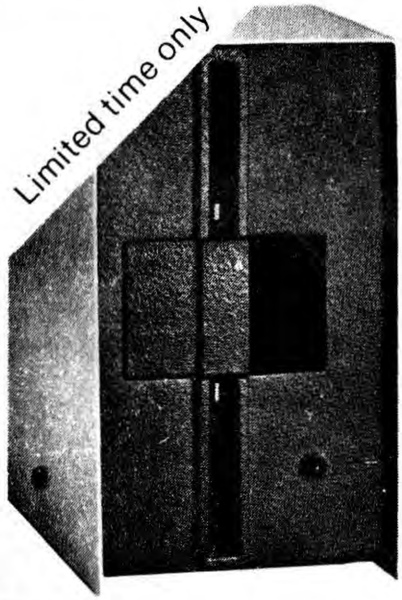
Once $995

Brand New

Now $399

While they last

Plus $10 for shipping & insurance



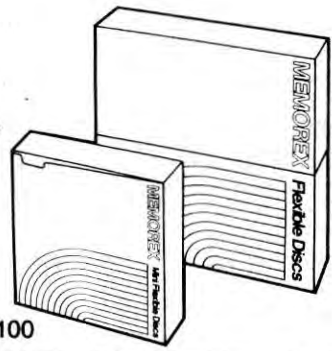
5¼ Disc Drives

For TRS-80 Mod. 1 35 or 40 track capability

$279

Single or double density (with proper interface)
Choice of MPI or Tandom drives

Add $7.50 for shipping



5¼ Discs for TRS-80 and Apple

Prime First Quality
MEMOREX Discs as low as **$1.89 per disc** In lots of 100

Part No.	10 or More	50 or More	100 or More
3401	**$2.09**	**$1.99**	**$1.89**
3431 with Hubring	**$2.35**	**$2.25**	**$2.15**

MiniMicroMart, Inc

master charge **943 W. Genesee St. Syracuse, N.Y. 13204 (315) 422-4467** VISA

WE WILL NOT BE UNDERSOLD

CP/M® Software	Software/Manual only
Computer Pathways	
Pearl (level 1)	$ 99/$25
Pearl (level 2)	$299/$40
Pearl (level 3)	$549/$50
Digital Research	
PL/I-80	$459/$35
BT-80	$179/$30
Mac	$ 85/$15
Sid	$ 65/$15
Z-Sid	$ 90/$15
Tex	$ 90/$15
DeSpool	$ 50/$10
Micropro	
WordStar	$319/$60
Customization Notes	$ 89/$na
Mail-Merge	$109/$25
WordStar/Mail-Merge	$419/$85
DataStar	$249/$60
WordMaster	$119/$40
SuperSort I	$199/$40
Spell Star	$175/$40
Microsoft	
Basic-80	$289/$na
Basic Compiler	$329/$na
Fortran-80	$349/$na
Cobol-80	$574/$na
M-Sort	$124/$na
Macro-80	$144/$na
Edit-80	$ 84/$na
MuSimp/MuMath	$224/$na
MuLisp-80	$174/$na
Organic Software	
Milestone	$269/$30
Supersoft	
Diagnostic I	$ 49/$20
Diagnostic II	$ 84/$20
Disk Doctor	$ 84/$20
Forth (8080 or Z80)	$149/$30
Fortran	$219/$30
Fortran w/Ratfor	$289/$35
Other	less 10%
Unicorn	
Mince	$149/$25
Scribble	$149/$25
Both	$249/$50
Data Base	
FMS-80	$649/$45
dBASE II	$595/$50
Access/80	$699/$50
Pascal	
Pascal/MT+	$429/$30
Pascal/M	$189/$20
Miscellaneous	
SpellGuard	$299/$25
The Last One	$549/$na
SuperCalc	$269/$50
CBASIC-2	$ 98/$20
MicroStat	$224/$25
StatPak	$449/$40
Micro B+	$229/$20
Apple Software (Business)	
Micropro	
Wordstar	$269
MailMerge	$ 99
Wordstar/MailMerge	$349
SuperSort I	$159
Spellstar	$129
Personal Software	
Visicalc 3.3	$159
CCA Data Mgr	$ 84
Desktop/Plan II	$159
Visiterm	$129
Visidex	$159
Visiplot	$149

Personal Software (cont.)	
Visitrend/Visiplot	$ 229
Zork	$ 34
Miscellaneous	
Micro Courier	$ 219
Super-Text II	$ 127
ASCII Express	$ 59
Apple Software (Entertainment)	
Wizard & Princess	$ 28
Mystery House	$ 24
Flight Simulator	$ 29
Raster Blaster	$ 26
Space Eggs	$ 18
Sargon II	$ 29
ABM	$ 22
Micropainter	$ 29
Apple Panic	$ 28
Pool 1.5	$ 26
Apple Accessories	
Z-80 Softcard	$ 299
Keyboard Enhancer	$ 110
Apple Joystick	$ 49
Sup-r Mod	$ 25
CPS Multifunction Card	$ 199
Videx Board	$ 249
16K Card	$ 159
Sup-r Fan	$ 39
ALF9 Voice Board	$ 149
CCS Cards	$Call
CCS Parallel Model 7720	$Call
CCS Serial Model 7710D	$Call
CCS Centronics Model 7728	$Call

Disk Drives For TRS-80* Model 1

CCI-100 5¼", 40 Track	$ 299
Add-ons for Zenith Z-89	
CCI-189 5¼", 40 Track	$ 389
Z-87 Dual 5¼" system	$ 995
Drives for Z-90	$Call

External card edge and power supply included. 90 day warranty/one year on power supply.

Corvus 5M	**$ 3089**
Corvus 10M	**$ 4489**
Corvus Mirror	**$ 699**
Shugart 8" 801R Raw Drive	$ 399
TANDON 5¼" Raw Drive	$ Call
Power Supplies	$ Call
Diskettes—Box of 10	
Maxell 5¼"	$ 40
Maxell 8"	$ 45
BASF/Verbatim 5¼"	$26.95
BASF/Verbatim 8"	$ 36
Plastic File Box—Holds 50 5¼" dskts.	$ 19
Plastic Library Case 5¼"	$ 3
Plastic Library Case 8"	$ 4
Head Cleaning Diskette	$ 25
Floppy Saver	$10.95
Floppy Saver Rings	$ 6.95
16K RAM Kits	
One Kit	$ 19
Two Kits	$ 37
200ns for TRS-80*, Apple II, (specify): Jumpers	$ 2.50
Computer Systems	
Altos ACS8000 Series	$ Call
Atari 400	$ 359
Atari 800	$ 789
Call for other Atari products	
Zenith Z89, 48K	$ 2149
Zenith Z90, 64K	$ Call
Call for other Zenith products	

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require one to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

Terminals	
Adds Viewpoint	$ Call
Zenith Z-19	$ 719
Televideo 910	$ 519
Televideo 920C	$ 729
Televideo 950	$ 929
S-100 California Computer Systems	
Mainframe	$ 349
Z80 CPU	$ 239
64K RAM	$ 569
Floppy Disc Cntrl	$ 339
Integrated Sys. w/int. cables, tstd.	$1975
2P + 2S I/O	$ 269
4 Port Serial I/O	$ 249
4 Port Parallel I/O	$ 179

Casio Calculators		
Pocket Comp.	FX702	$199.00
Desk Printr/Calc.	FR100	$ 79.95
Scientific Calc.	FX8100	$ 49.95
Game Watch	CA90 Plastic	$ 49.95
Game Watch	CA901 Steel	$ 69.95
Calendar Watch	AX210	$ 59.95

Printers **NEC Spinwriter**

7710 R.O. Ser	**$2395**
7710 Ser w/tr.	**$2595**
7720 KSR w/tr.	**$2795**
7730 R.O. Par	**$2395**
7730 R.O. Par w/tr.	**$2595**
NEW 3500 Series	**$ Call**
Epson MX-70	**$ Call**
Epson MX-80	**$ Call**
Epson MX-80FT	**$ Call**
Epson MX-100	**$ Call**
PaperTiger 445 Gr. & 2K	**$ Call**
PaperTiger 460 Gr. & 2K	**$ Call**
PaperTiger 560 Gr.	**$ Call**
IDS Prism 80	**$ Call**
IDS Prism 132	**$ Call**
PaperTiger Access.	**$ Call**
Anadex DP-8000	**$ 849**
Anadex DP-9500/01	**$1389**
Okidata Microline 80 Fric. & pin feed	**$ Call**
Okidata Microline 82A Fric. & pin feed	**$ Call**
Okidata Microline 83A 120 cps	**$ Call**
Okidata 84 200 cps	**$ Call**
Centronics 739	**$ 739**
C.Itoh Starwriter I 25 cps, par.	**$1525**
C.Itoh Starwriter I 25 cps, ser.	**$1620**
C.Itoh Starwriter II 45 cps, par.	**$1950**
C.Itoh Starwriter II 45 cps, ser.	**$2075**
Axiom GP-80M	**$ 319**
Data South 180 cps	**$ Call**
Olivetti DY 211 Daisy Wheel	**$ Call**
Monitors	
Leedex 12" B&W	$ 129
Leedex 12" Green Screen	$ 139
Leedex 13" Color	$ 329
Sanyo 9" B&W	$ 149
Sanyo 12" Green Screen	$ 238
Sanyo 12" B&W	$ 219
Sanyo 13" Color	$ 399
Zenith 13" Color	$ 349
Zenith 12" Green Screen	$ 129
Telecommunications	
Prentice Star Modem 1-yr. guar.	$ 125
Univ. Data System UDS103LP	$ 149
Univ. Data System UDS103JP	$ 215
Novation Cat	$ 139
Novation D-Cat	$ 149
Novation Auto-Cat	$ 199
Novation Apple Cat II	$ 339
D.C. Hayes Smart. Modem	$ 249
D.C. Hayes Micro-Modem II	$ 295
CCI Telnet Com. Package	$ 135

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

420-423 Rutherford Ave., Dept. **MO2M**
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 VISA®

✓See List of Advertisers on page 290

80 CALENDAR

February

16-17 The Yankee Group, Cambridge, MA. **The Future of IBM, a seminar,** Plaza Hotel, New York City, NY.

17-20 Management Science America, Inc., Atlanta, GA. **Convention for users of MSA accounting software,** San Francisco Hilton, San Francisco, CA.

22-24 The Interface Group, Framingham, MA. **Federal DP Expo—show for end users in the multi-billion dollar federal government marketplace,** Sheraton Washington Hotel, Washington, DC.

22-24 Technical Education Research Centers, Cambridge, MA. **Workshop on microcomputers in education,** Michigan State University, East Lansing, MI.

23-24 The Yankee Group, Cambridge, MA. **The Future of IBM, a seminar,** Sunnyvale Hilton, Palo Alto, CA.

26-28 Adventure International, Longwood, FL. **Computer Expo '82 trade show,** Orlando, FL.

March

1-2 Michigan Association for Computer Users in Learning, Wayne, MI. **Sixth annual convention featuring sessions on facets of education uses for computers,** Western Michigan University, Kalamazoo, MI.

3-7 Catalyst, Jersey City State College, Jersey City, NJ. **Microcomputer Week '82, "an international event of significance to educators,"** Jersey City State College, Jersey City, NJ.

7-9 American Management Association, New York City, NY. **Course on paperwork management,** New York City, NY.

9-10 The Yankee Group, Cambridge, MA. **Seminar on the Super Service Bureau,** New York City, NY.

12-13 Seattle Pacific University and National Council for Computers in Education. **Fifth Annual Computers in Education Conference,** Seattle Pacific University, Seattle, WA.

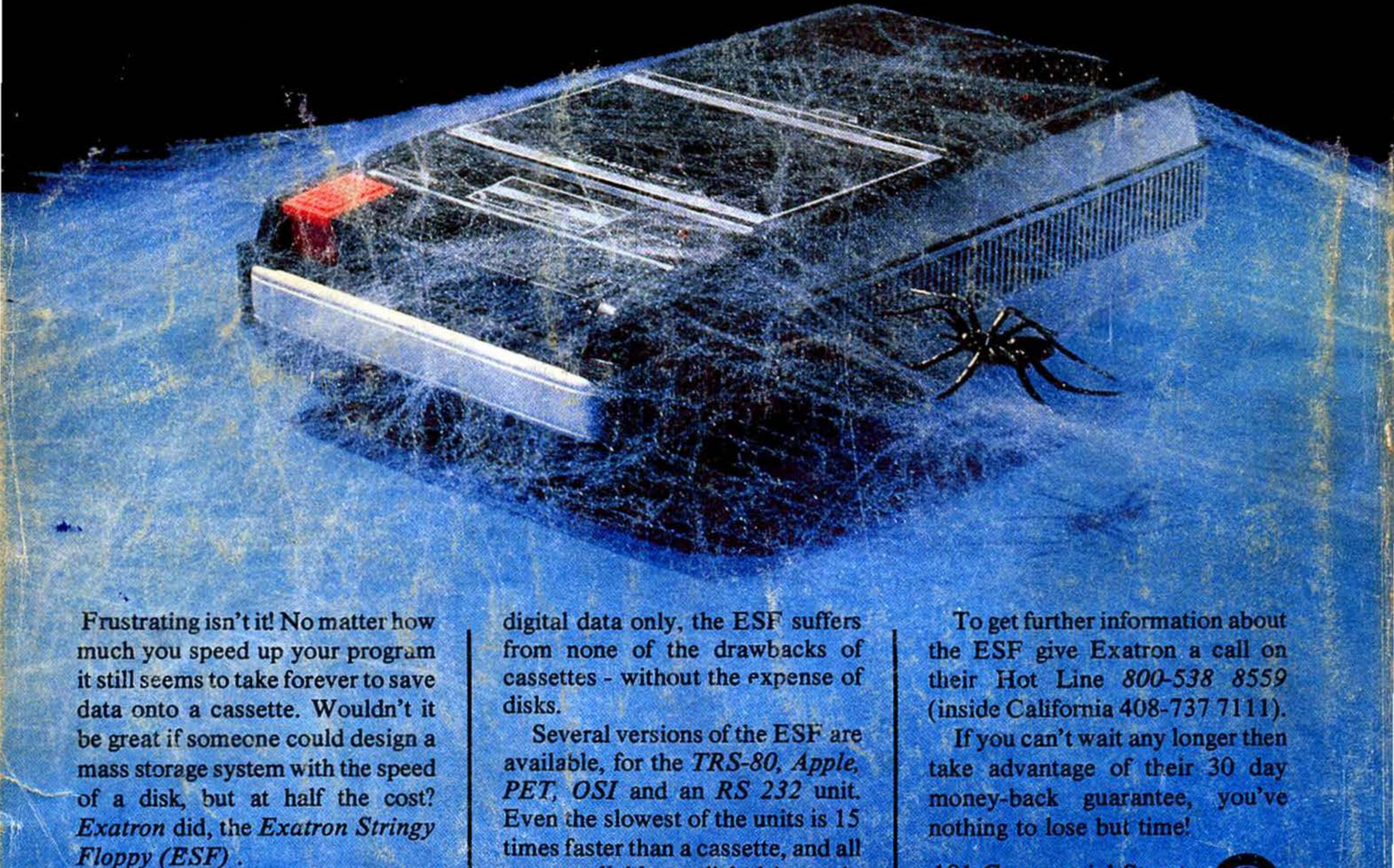
16-18 McGraw Hill International Training Systems, Edutronics Division, New York City **International Training Conference: Tools for Data Processing Training,** McGraw-Hill Building, New York City, NY.

Coming Next Month

Be prepared for an eyeful when *80 Micro* takes a look at Color Computer graphics. Graphics? you say. Dull! Tedious! Boring! Not so, replies Irwin Rappaport, and he'll show you an easy way to program pictures. Meanwhile, computer magicians Jake Commander and Dennis Kitsz will tell you how to create dazzling rainbow imagery. And don't miss Roy Green's approach to translating Bob Boothe's graphic phenomena into Color Computer fireworks.

69

TIRED OF WAITING?



Frustrating isn't it! No matter how much you speed up your program it still seems to take forever to save data onto a cassette. Wouldn't it be great if someone could design a mass storage system with the speed of a disk, but at half the cost? *Exatron* did, the *Exatron Stringy Floppy (ESF)*.

Totally self-contained, the ESF is an extremely fast, reliable, and economical alternative to cassette or disk storage of programs or data. All of the ESF's operations are under the computer's control, with no buttons, switches, knobs or levers to adjust or forget.

The ESF uses a miniature tape cartridge, about the size of a business card, called a wafer. The transport mechanism uses a direct drive motor with only one moving part. Designed to read and write digital data only, the ESF suffers from none of the drawbacks of cassettes - without the expense of disks.

Several versions of the ESF are available, for the *TRS-80, Apple, PET, OSI* and an *RS 232* unit. Even the slowest of the units is 15 times faster than a cassette, and all are as reliable as disk drives - in fact a lot of users say they are *more* reliable!

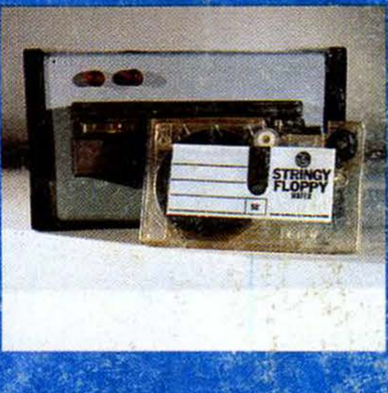
excellence in electronics

exatron

To get further information about the ESF give Exatron a call on their Hot Line *800-538 8559* (inside California 408-737 7111).

If you can't wait any longer then take advantage of their 30 day money-back guarantee, you've nothing to lose but time!

181 Commercial Street
Sunnyvale, CA 94086



TRS-80, Apple and PET are trademarks of Tandy, Apple and Commodore respectively.