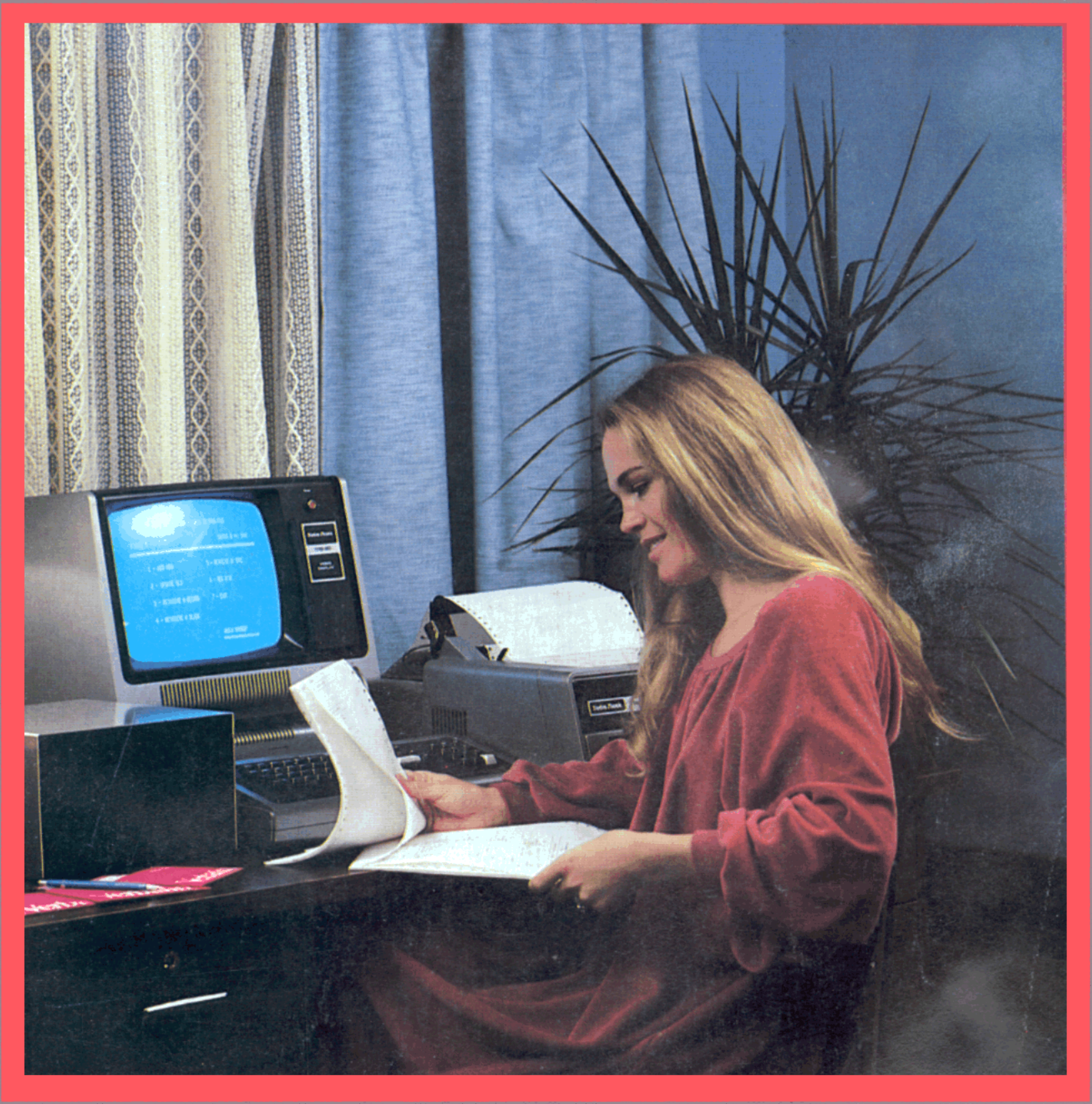
January 1980 $2.00

80 microcomputing

THE magazine for TRS-80* users

Disk Primer:
Mass storage demystified.

Basic BASIC renumber:
Less than 12 program lines!



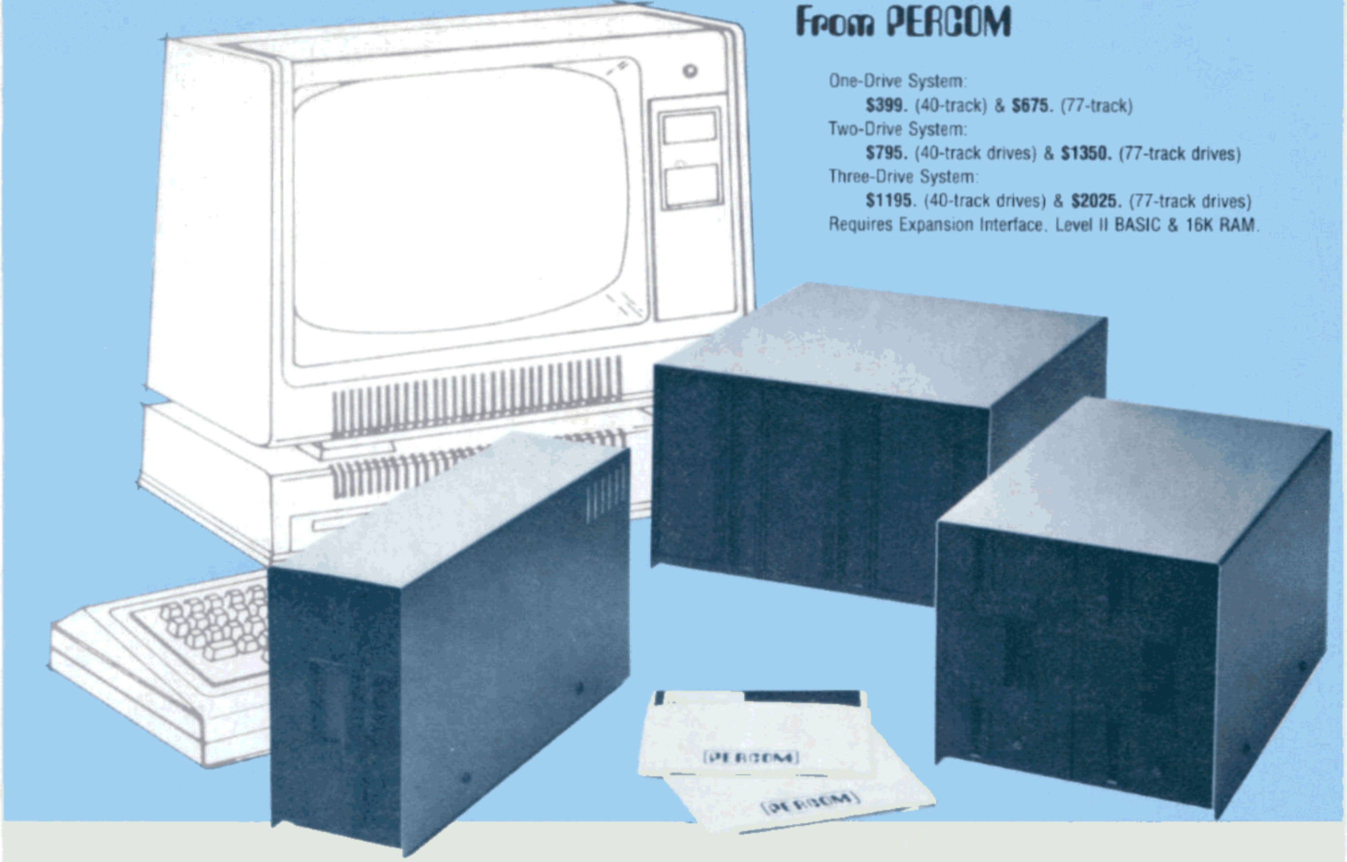
Business, Education, Utility, Games plus much more!

1st Issue

*a trademark of Tandy Corporation

From PERCOM

One-Drive System:
$399. (40-track) & **$675.** (77-track)
Two-Drive System:
$795. (40-track drives) & **$1350.** (77-track drives)
Three-Drive System:
$1195. (40-track drives) & **$2025.** (77-track drives)
Requires Expansion Interface, Level II BASIC & 16K RAM.



Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY
*TRS-80 and TRSDOS are trademarks of Tandy Corporation and Radio Shack which have no relationship to PERCOM DATA COMPANY

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND, TX. • 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

NOW... Exclusively Yours from ACS

TRS-80® -DOS 3.0

Disk Operating System by the Original Author

ACS is not affiliated with Radio Shack or Tandy Corp.

☐ DOS 3.0
EFFECTIVE SEPTEMBER 1
ONLY **$49.95**

Check these DOS 3.0 features . . . that go far beyond 2.1

NO KEY BOUNCE • UPWARDS COMPATABLE FROM 2.1 • CHECK EOF
SEEK • RE-READ • RE-WRITE • LOC • VARABLE LENGTH RECORDS
SKIP • DISK LOGGING OF MESSAGES • BOOT • CHAIN • WRITE EOF
PURGE • SET • RESET • ROUTE • RUN & LOAD FOR 1 DRIVE SYS.
XFER • FORMAT W/O ERASE • DIR FROM BASIC • PATCH • LINK
USER DEFINED KEYS • KEY AUTO REPEAT • UPPER AND LOWER
CASE DRIVER • SHIFT LOCK • RS-232-C DRIVERS
• PAUSE • USER DEVICE CONTROL BLOCKS
• MULTI PROTOCOL COMMUNICATIONS

HARDWARE FOR TRS-80®

☐ **Pertec Disk Drives FD-200** .... **$375.00 ea.**
These are 40-track Drives that are completely compatible with the TRS-80 and Radio Shack Drives. 3.0 DOS $20.00 extra with disk drive. Will allow Turning Diskette over and Write on other side.

☐ **4-Drive Cable for Pertec Drives** **$35.00**

☐ **DECwriter III, 132 Character** **$2500.00**
110 to 9600 band EIA tractor feed keyboard printer. This is truly the nicest printer available. (30 day delivery)

☐ **701 Centronics TRACTOR FEED Bidirectional Printer** **$1500.00**
2½ times as fast as the Radio Shack 779 Printer, has full size 132 Char. Carriage Bell tone. Complete with Cable plug in and use. Shipped Freight COD.

☐ **200 ns 16K Dynamic Memory Clips for Keyboard or Expansion Interface, Lifetime Guarantee, complete** **$110.00**
Lifetime Guarantee. Complete with Instructions and Jumper Blocks.

☐ **10 Key Numerical Keypad Kit** **$79.95**

☐ **TRS-80® Level II – 16k** **$750.00**

☐ **Expansion Interface** **$275.00**

☐ **RS-232-C Interface** .... **$ 89.00**

ORDER NOW AND SAVE

Just list the items you want and mail this convenient coupon.

SOFTWARE BY ACS

☐ **Monitor No. 3** **$29.95**
Complete Machine Language Monitor for TRS-80 features: Find, EDIT, Relocate, Symbolic Dump to Tape, etc.

☐ **Monitor No. 4** **$49.95**
All of the commands that reside in Monitor No. 3, plus: RS-232 I/O, Disk Program I/O, Symbolic Dump to Disk for Loading into Disk Editor/ASM., Track & Sec I/O for modification.

☐ **PCLEND** **$15.95**
Will Patch ASCII files of Basic Programs or text or DATA FILES so that they may be loaded into the Disk Version of the Electric Pencil for Editing purposes comes on Cassette that will automatically create a Disk file of PCLEND.

☐ **MAKE TAPE AND MAKE DISK for Cassette Dealers** **$69.95**
These are two programs that will allow you to take any type of Program from Disk and store it on tape for mailing purposes. When the user receives the program in the mail on cassette, it is loaded into the computer which will automatically make a Disk file of the program.

☐ **CP/M & C BASIC for the TRS-80®**
CP/M Includes: MOVCPM, STAT, PIP, Dump, DDT, ASM (8080), ED, plus 6 user manuals.
CP/M **$150.00**
C Basic-2 Includes: XREF2, CBAS2, and manuals.
C BASIC 2 .... **$99.95**

☐ **G2 LEVEL III BASIC for TRS-80®** .... **Special $39.95**

☐ **TELCOM – Telecommunications for the TRS-80®** **$29.95**
Telecommunications for the TRS-80® allows one TRS-80® to communicate with another through the RS-232-C over the phone line.

VISA • Master Charge

Orders received by 6:00 p.m. shipped next day on Master Charge, Visa, Certified Check or Money Order. Personal Checks require 14 days to clear. No C.O.D. Collect calls not accepted. All Hardware warranted through ACS for 1 full year. Software guaranteed for replacement only. Prices subject to change without notice.

AUTOMATED COMPUTER SOFTWARE SERVICE
(615) 244-2798
Division of
Computer World INC.
625 Main Street • Nashville, TN 37206

Send Check or Money Order payable to –
SOFTWARE • P.O. Box 60097 • Nashville, TN 37206

Quan.	Description	Unit Price	Total

HANDLING CHARGE $1.50
TENN. RES. ADD 6% SALES TAX **TOTAL**

☐ Check
☐ Money Order
☐ MasterCharge
☐ Visa
Card No. ____________
Exp. Date ____________

Name ____________________
Address ____________________
City ____________ State ____________ Zip ____________

Great News from HMCT

TRS-80* MODEL I AND MODEL II IN STOCK

Before you purchase your TRS-80* Model I and Model II from your local Radio Shack, or consider Mail order for a discount, let me offer you an alternate choice. Houston Micro Computer Technologies, Inc. has been providing TRS-80 Software and Hardware to hundreds of users for almost 2 years. We are not a mass merchandiser nor are we a discount house, but a group of professionals dedicated to helping businesses implement microcomputers to their greatest advantage. Here are just a few of the advantages we offer over Radio Shack.

DELIVERY	Less than 30 days**
RELIABILITY	Each unit 100% tested prior to delivery
SATISFACTION	Sold by computer professionals who understand what you need your computer to do
VALUE	Trade in value on Model I systems dependent upon condition and configuration
SERVICE	National service on all products we sell including TRS-80s*
CONFIGURATION	As a distributor for many lines of computer peripherals we can tailor your system to your exact hardware requirements
SOFTWARE	We are an already established Model I Systems House with an extensive software library

Maybe you believe a mass merchandising organization or a discount operation is who you should trust your computer hardware selection to. Personally we like to think you would prefer to deal with specialists and professionals who will spend the time with you personally or on the phone to assist you in your choice of equipment based on your needs and specifications.

All equipment purchased from Houston Micro Computer Technologies, Inc. is fully warranted and backed by national service organizations including Radio Shack.

For further information call collect, **713/661-2005** or write.

** Delivery of Model II subject to availability.

*TRS-80 Is a registered trademark of the Tandy Corporation.

Texas residents add 6% sales tax • MasterCharge • Visa

HOUSTON MICRO-COMPUTER TECHNOLOGIES, INC. ✓13

Home and Business Computer Specialists

5313 BISSONNET • BELLAIRE • TEXAS • 77401 • 713/661-2005

January 1980 Issue #1

80 microcomputing DATA

80 Microcomputing (ISSN # applied for) is published monthly by 1001001 Inc., Pine Street, Peterborough, NH 03458. Application to mail second class postage rate is pending at Peterborough, NH 03458 and at additional mailing offices. Phone: 603-924-3873/4. Subscription rates in the U.S. are $15 for one year and $40 for three years. In Canada $17 for one year and $46 for three years. In Europe please contact Monika Nedela, Markstr. 3, D-7778, Markdorf, W. Germany. In South Africa contact 80 Microcomputing, P.O. Box 782815, Sandton, S. Africa 2146. Australian distribution by Katherine Thirkell, Sonotron Instruments, 17 Arawatta Street, Carnegie, Victoria 3163, Australia. All other foreign subscriptions $20 (one year only) surface mail. 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.

This isn't TRS-80™

TRS-80 is More Than a Microcomputer. TRS-80 is Radio Shack's growing *family* of cost-effective computers, peripherals and software — supported by service before, during and after the sale. In scarcely two years, over 100,000 TRS-80s have gone to work for businesses, educators, professionals, and folks desiring to combine their own ingenuity with a personal computing system. Why is TRS-80 so successful?

E-x-p-a-n-d-ability! The $499* TRS-80 4K starter system pictured can be expanded to a 48K RAM, 4-disk system with printer — or anywhere in between — to handle your needs. There are dozens of ready-to-run cassette and disk programs. And TRS-80's BASIC language makes it easy to learn to write your own.

Peripherals! More memory, disk drives, printers, a voice synthesizer, and an acoustic coupler for telecommunications are available through Radio Shack outlets everywhere. More is on the way.

New TRS-80 Model II. Systems start at just $3450* — offering features of small computers by IBM and others, but at 33% to 66% savings. Model II is twice as fast as Model I, with expansion capabilities to 64K RAM and two megabytes of disk storage. Features include vectored interrupts, direct memory access, 80x24 screen, and more.

Sales and Service! We've opened 50 specialized Computer Centers, 100 expanded Computer Departments in Radio Shack stores, plus 105 Service Centers in the USA alone. We offer training classes, a free newsletter for TRS-80 owners, and a toll-free customer service number.

We're in Computers to Stay! Radio Shack has what it takes to be your computer company, today and in the future. We're making the commitment now — in facilities, research, and product development — to make sure TRS-80 remains your best small-computer value!

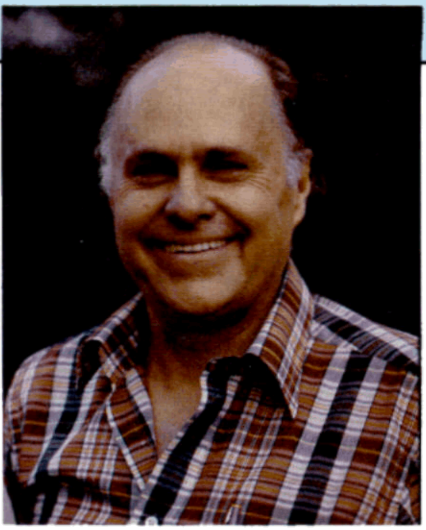
A DIVISION OF TANDY CORPORATION
FORT WORTH, TEXAS 76102 • OVER 7000 LOCATIONS IN 40 COUNTRIES
*Retail prices may vary at individual stores and dealers.

Radio Shack®

...This is TRS-80™
The biggest name in little computers™
FREE
TRS-80 Catalog
Send to: Dept. CMA-446,
1300 One Tandy Center
Fort Worth, Texas 76102
Yes! Send the 24-page full-color catalog
You Bet! Have a representative call me. Number below.
NAME
FIRM
ADDRESS
CITY
STATE
ZIP
PHONE
CASINO
INVASION FORCE
FLYING SAUCERS
57 PRACTICAL PROGRAMS & GAMES IN BASIC
Inventory Control I

80 REMARKS
by Wayne Green

Where We're Coming from/Where We're Bound



I'm Wayne Green, the editor and publisher of *80 Microcomputing*. To give you a better idea of what I have in mind for this magazine, I'd like to give you an idea of its origin and my concepts . . . .

First, I want to make it clear that this magazine is not connected with Radio Shack or Tandy. I call 'em as I see 'em and don't pull the punches. Where Radio Shack deserves credit, they'll get it. Where I think they are screwing up, I'll be blunt about that. I don't ask that you like me—that's *your* problem, not mine. I like *you* and I will be working for your best interests . . . and so will the magazine.

The TRS-80 draws mixed reactions. It is derisively called the Trash-80 by many in the microcomputer industry. It is also adored by tens of thousands of users. I don't love or hate it . . . and I will try to be as objective as I can. Love it or hate it, you really can't ignore the TRS-80. There are more of them out there than any other single computer . . . more than all other microcomputers combined, and Radio Shack is selling them just as fast as they can be made.

Yes, I know about the keyboard bounce problem and the crummy cassette loading circuit that drives users up the wall. Yes, the monitor has the bandwidth of a cheap black and white television set, instead of clear and crisp graphics, but, hells bells, it *is* a cheap black and white television set with a bit of conversion for computer use.

Yes, the keyboard needs more air blown into it to keep it cool, and the power supply is marginal. Sure Radio Shack is five months back ordered on disk drives. No one has ever claimed the TRS-80 is perfection.

What it is is a damned good computer at a fantastic price. All the gripes have some relatively simple answers or fixes. Heck, you can make a longer list of grumbles about a Volkswagon Beetle . . . but it is *still* a great little car.

In the pages of this magazine we will try to come to terms with the problems and offer solutions. We'll make sure that as much of the information is written in plain English as possible. But it won't take long, even for the rankest beginner, to start getting some understanding of how the TRS-80 works—and how to get a lot more out of it.

We'll have information on the newest in Radio Shack accessories and peripherals. We'll not only evaluate them, but have articles on their use written by the readers. These will give you ideas for improving your own system and getting much more out of it.

The large numbers of TRS-80 systems have naturally produced many small firms with supporting equipment and programs. As more and more uses evolve for the TRS-80, we'll be bringing you understandable articles on the accessories involved. Our readers will tell you what they think of many of these gadgets.

The News section will keep you up to date on the latest peripherals and software for your system. There are more uses than can be imagined right now . . . for writing music, watering the lawn, interfacing with video recorders, security systems, running the house, replacing mail with electronic mail . . . it is endless.

The most important part of the magazine will be the software which is published and the evaluations of commercially available software. Few of us have the time, talent and experience to write complicated programs, so we need to patronize those who do have this wonderful combination. Most of the programs in the future will be published through program publishers such as Instant Software . . . and you will want to know what is available and how good it is.

Business Applications

The low cost and prodigious work capacity of microcomputers make them invaluable for every business. As more and more programs are developed, we'll be watching them and evaluating them for you . . . and your business.

A business computer has to be able to handle a large number of jobs such as bookkeeping, word processing, electronic mail, security, payroll, message center, file handling, etc. It also has to work day in and day out, and if it fails, be quickly repaired. *80 Microcomputing* will keep these parameters in mind, pushing the industry to fill these needs.

I think we're heading into a computerized society which will support, just in our country, perhaps 200 million microcomputers in business, schools, homes and cars, etc. With an average life of five years, an investment of $1,000 in hardware and an equal amount in software . . . plus service . . . plus inflation . . . I think within ten years we'll have a market for microcomputers on the order of $50 billion per year.

Computer Aided Instruction

An educational system which is both fun and unrestrictive holds the best possibilities for our future and I think microcomputers are going to be a big part of this. Firstly, computers make wonderful teachers because they do not get emotionally embroiled with the student. They have infinite patience with a student.

I don't know what directions computerized education will take, but I see the video tape recorder and the microcomputer combined. I see educational programs in full color with professional photography showing how things work and all described by authorities in that field. Children in the future may learn many times what we do today because the material is so interesting. This may result in learning concepts and knowledge at home on computer and learning skills in schools with the help of teachers.

Feedback Is Important

This magazine is aimed at the average TRS-80 owner. Let us know if the material is over your head. If you think it's too simple, again let me know.

While most of the firms in the field are trying hard to put out the best possible products at a reasonable price, there are (unfortunately) some chislers out there. We try hard to spot these chaps and keep their ads out of our magazines, but now and then someone outwits us. I want to hear from you when you feel you have been shafted.

The way to register a complaint under most circumstances is to cite the step by step growth of the problem and put it all into a letter to the firm, citing dates and amounts, phone calls, letters. Then mark on the bottom that you've sent a copy to Wayne Green—and do it. I'll see what I can find out about your problem . . . and this pressure often works miracles. If I see a pattern of complaints, I may stop their right to advertise until I feel that they are responsive.

"Try to think of 80 Microcomputing as more of an informal newsletter than a pontificating magazine."

You Are The Author

All of my magazines have one thing in common—they are written almost entirely by the readers, *you*. If you buy something which should be written about, write about it.

If you get some software which is great, write about it. If it is awful, say so. Try to think of *80 Microcomputing* as more of an enormous and informal club newsletter than a pontificating magazine.

The editor of *80 Microcomputing* is not trying to impress you with his great knowledge of computers, I just want to provide you with entertainment, a way to learn more about your system, a way to save money on developing it.

Submissions

You may want to know more about how to prepare articles. Not difficult. We do have a poop sheet which goes into details, but you can get started right away.

Material must all be typewritten in upper-case and lowercase type, double spaced with generous margins. Good photos of equipment are very important, no Instamatics, get a camera hobbyist or professional photographer. Program listings should be printed out with an impact type of printer if at all possible. If you haven't got a listing then send along the program on cassette and we'll print it in our lab. Send the cassette anyway.

Always include return postage, unless you don't want things back. Programs you're also submitting to Instant Software should be so indicated. In some cases we'll publish first in a magazine and then make the program available on cassette or disk.

"The first magazine I published was in 1952 about amateur radio Teletype."

Since I'm the driving force behind 80 you probably want to know a bit about me—unless you already do by virtue of reading my other magazines down through the years.

I got bit by the ham radio bug a little over 40 years ago, when I was an electronics technician on a submarine. After the war and college I tried radio and television broadcasting before starting a high fidelity manufacturing business. Then I tried publishing and enjoyed it more than anything I'd tried.

The first magazine I published was in 1952 about amateur radio Teletype. Later I became editor of *CQ*, a ham magazine.

I started my own magazine for hams in 1960, that was *73* magazine. *73* is now the world's largest ham publication, with subscribers in over 200 countries.

When Mits put the first microcomputer kit on the market in 1975 I organized and did most of the work to get *Byte* magazine started—it is now owned by McGraw Hill.

When I felt there was a need for a magazine aimed at beginners in computing, I started *Kilobaud Microcomputing* in January 1977. It's doing well and will continue to help newcomers into the computing field, though with less emphasis on the TRS-80. I will still be publishing the more complicated TRS-80 articles in *Kilobaud Microcomputing*.

Feeling that computer and business education is more than a small need, I've gotten together with a local college, Franklin Pierce College, Rindge, New Hampshire, and they are developing a series of courses for people interested in microcomputing as a career.

The Software Emergency

After writing in my editorials for a couple of years urging some firms to get involved with publishing software, I finally gave up pushing and started to work setting up one myself. The result is Instant Software, Inc. It is no less work to write complicated business programs for a microcomputer than for a mainframe IBM monster. Those of you involved with data processing know that program packages run usually around $20,000 to around $50,000.

My concept is to sell the $50,000 programs for a relatively low price with a royalty of, say, $5. By selling it through computer stores by the thousands, the programmer gets his $50,000 and ten thousand or more computer owners are able to use a $50,000 program, for a fraction of the price.

Instant Software already has nearly 100 programs on sale for the TRS-80 and hundreds more in the works. The operation has nearly 50 people working full time, plus many more doing part-time work. The company operates out of offices in Peterborough, New Hampshire—and has probably the largest microcomputer laboratory in the world.

"My outspokenness aggravates a lot of people, but . . . if I don't kick ass . . . I don't get results.

My outspokenness aggravates a lot of people, but I find that if I don't kick ass now and then I don't get results.

Other than as editor of the magazine I live a normal work-a-holic's life. I enjoy some of the better TV programs, but by the use of a couple of VTR systems I am able to route this entertainment into my own time pattern rather than that of the networks. I like movies, when I have time. I read a lot, perhaps something over a hundred magazines a month, plus several books a month. I very much enjoy skiing, skin diving, driving (RX7 and 280Z), cooking, eating good food, traveling (I was at the electronic shows in Seoul, Osaka, Taipei and Hong Kong in early October . . . and I'll be at the Winter Consumer Electronic Show in Las Vegas in January . . . Saroc hamfest same place a couple days later . . . then the annual Ham Industry Workshop in Aspen . . . with skiing . . . in mid January), hamming (I have a World Record on 10 GHz), microcomputing, etc.

In addition to editing and publishing magazines and newsletters for the ham industry, or for my old submarine crew reunions, I work with the Chamber of Commerce on growth patterns for the Peterborough, New Hampshire area. I'm also working to get amateur radio and microcomputers into more of the emerging nations. They need them both badly.

Little of this would be possible without a fantastic support crew—Sherry Smythe, who does most of the actual running of things, Jeff DeTray, my assistant, who does most of the hard work . . . and about 120 others. I come up with the ideas—get them started—and add momentum every now and then.

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

One of the most unpleasant tasks for many people in and out of business is keeping financial records. Not only is precision required, but accounting recordkeeping is dull, routine and necessary. For this reason, bookkeeping machines were one of the earliest automated devices available in the business environment. Originally, a miracle of springs and levers, the most recent versions were partially and then completely transistorized.

Today's bookkeeping machine is a full blown mini or micro computer and it is doing its thing in thousands of offices all over the country. These commercial machines have much in common with the TRS-80. In fact, the 80 is so well suited for many bookkeeping tasks, much software is being written specifically for the small bookkeeping market.

This column will explore some of the more useful TRS-80 bookkeeping software that is available and its application to financial recordkeeping.

I intend to include a little systems theory as well as a review of each program. If you don't scare easily, you should begin to understand your own requirements so you can choose the system that suits your needs.

The Requirements

Accounting recordkeeping systems generally include the following elements.

- Journals or "Books of Original Entry"—Document, Symbol
- Subsidiary Ledgers such as Accounts Receivable—Circle Symbol
- General Ledger

These books can be maintained in many forms; as ledgers, account cards or on continuous forms called registers. From an accounting point of view, form is not as important as content. Regardless of the form utilized, the more information contained on the record, the more valuable it is.

Journals typically include activity for only one month. An account summary is then recorded or posted into the general ledger. The ledger thus becomes an index of fiscal activity with each account pointing to the detail contained in the journal. Account balances can then be arranged and detailed on income statements and balance sheets.

The value of the general ledger as an index or pointer to detailed journals becomes apparent when questions must be answered concerning balances reported on the financial statements. Obviously, if sufficient detail is contained in the index, the "audit" trail to the source material will be easier to follow.

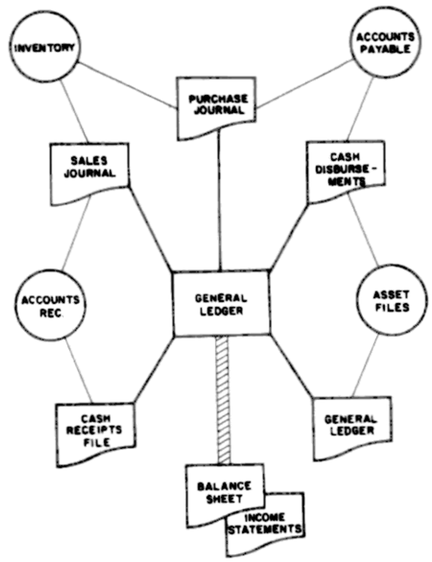


Figure 1.

The subsidiary ledgers such as accounts receivable, accounts payable and inventory should also have as much activity as possible. When cash is received, or payments are made, it is important to know what is paid for, so that discrepancies can swiftly be resolved. Similarly, inventory activity records are most valuable when historical receipts and sales can be examined.

Enough of theory. How does this affect my TRS-80 bookkeeping project? Well, in a nut shell, the TRS-80 is not good at keeping historical records. This is because of the limited capacity of the mini diskette.

In fact, this restricted capacity even limits the number of accounts that can be retained. For example, Radio Shack's general ledger will only permit 100 accounts. The accounts payable and receivable programs typically allow only 350 accounts per drive.

Saving Space

To save limited disk space, most designers of recordkeeping systems use "Balance Forward Recordkeeping." This technique reports period activity with an opening balance, a summary of net changes in the balance and the closing balance. Obviously, research as to why changes have occurred will be quite difficult with this technique.

Systems that use "Balance Forward Recordkeeping" must have a method to provide an Audit trail. This method is usually a long and involved report, analyzing the distribution of every transaction that affected the account.

One system, The Bottom Shelf's CRAS (Check Register Accounting System), solves the problem of an audit trail by storing each month's cash disbursement detail on a separate diskette. At $5.98 per diskette (Radio Shack) this system will require over $70.00 worth of diskettes for one year's data storage. To analyze one year's account activity, 12 diskettes must be inserted—a tedious but necessary task.

If you must have a Radio Shack system, then the Model Two is the answer. This computer is capable, when fully expanded to handle up to 2 Megabytes or over 5½ times as much storage as the Model One.

Another major consideration in financial recordkeeping system design is response time. Response time is the time the system takes to respond to an operator's inquiry.

By now, almost all users of the 80 know the difference between machine language and BASIC programs. In the assembler manual a simple program is presented that turns the screen white almost instantaneously. A BASIC program to do the same task takes many times longer. Since most financial programs are written in BASIC, the time required for the machine to respond becomes important when evaluating a program's suitability for the business environment. A response time of two seconds can seem like eternity to the operator when posting a large volume of transactions.

Software Performance

Now let's discuss some currently available financial software. First, a few ground rules:

1. I cannot mention software if I have never seen it.
2. My opinions are my own.
3. If I have seen it and it has no merit—I won't mention it.
4. I am not and do not claim to be a "Crackerjack" programmer.

The first software package is the Radio Shack General Ledger. In accounting systems designs, it occupies the central block of the schematic.

The General Ledger package (cat. 26-1552) requires a dual disk 32K system with a line printer. Tractor feed is optional but recommended. The system is capable of handling up to 240 entries per session (more on that later) with a maximum of 1830 per month. The system can accommodate no more than one hundred accounts.

It is supplied on disk in an attractive binder which includes two diskettes, documentation and program listings. The system is organized so that drive zero disk contains the DOS.

As a package, version 1.1 is a substantial amount of work. The documentation even includes a brief letter of introduction to the president of the firm and a work session. The work session uses examples which include sample data entries, accounting theory and sample output reports. If followed as presented, it constitutes a training program on how to use the system.

However, the accompanying program documentation is inadequate. In fact, the only positive thing about the program documentation is that it will save you the trouble of listing it on your printer. Missing, are necessary items such as: Rem statements in the programs; definition of variable assignments; program logic flow charts; file descriptions.

Elimination of spaces between instructions and the multiple statements on each line further complicate program analysis. If you plan to personalize the system to add some options, you are in for a very rough time.

Since listings are provided, I doubt that it was Radio Shack's intent to mislead you. The program listings were not intended for the user and therefore no investment was made in providing a minimal acceptable level of program documentation.

Strong Points and Weak Ones

As an accounting system, the General Ledger has some strong points and some weak ones. It is income statement oriented with provisions for six categories of Income Statement Items.

Using these categories, the user can group financial activity into summary areas for quick analysis and comparison purposes. Two of the categories are permanently assigned to Sales and Cost of Sales. The other four are user-programmable. These four categories will contribute significantly to statement readability and evaluation of operating results.

The income statement includes a percentage column and a year to date column. It is designed to be printed on 8½ by 11 inch paper and should be kept in a loose leaf binder. In this manner, next year's data can be added to the book and a comparative analysis readily obtained. The system does not provide for a budget or any other historical comparison.

The balance sheet on the other hand, comes straight out of a basic Accounting textbook. There is no provision to group assets and liabilities into current and non-current portions. As a result, the important liquidity ratios of working capital, current and quick cannot be calculated easily. This is unfortunate, since all of the information is available and would make a very desirable system extension for the small businessman.

In addition, it would be very desirable to develop a report of other key management information such as the following:

1. Solvency Ratios such as:
   - —Times Interest Earned
   - —Debt to Equity Ratios
     - ● total debt to total assets
     - ● long term debt to capitalization
     - ● total debt to equity
2. Fund Management Ratios
   - —Average Collection Period
   - —Inventory Turn Over
3. Return on Investment Ratios
   - —Return on Total Assets
   - —Return on Capital
   - —Return on Net Worth
   - —Investment Turnover

Once the system is set up and all accounts and categories defined, data is entered during "sessions." Each session consists of two phases, data entry and report generation. As long as activity is entered, a session report must be generated.

The system assigns a serial number to every document entered into the system. This internal serial number is intended to provide an audit trail to each bit of accounting data entered irrespective of any number which may be noted on the document. Therefore, note the serial number on the document when posting to prevent duplicate entry and file the source documents in a numerical file to facilitate your reference. The system provides utilities to permit sorting information by the "document number" or account code.

"Unfortunately, the account distribution report is not only unwieldy but is hard to use."

Slow Posting, but Recommended

The actual posting procedure is quite involved. The system records a document name, issues a date of the original document, posts a date if different than the session date (useful at year end), a total amount and account distribution.

Because the document amount may be distributed to multiple accounts, the system brings up the account name when the number is keyed to prevent erroneous coding. All postings must zero balance, that is, the debits must equal the credits or an out of balance message is flashed.

This balancing procedure is a good one. However, it means that a minimum of three totals must be entered in the system for every transaction; the transaction total amount, the debit amount and the credit amount. The system can calculate the debit or credit amount automatically if two of the required amounts are entered and a question mark is inserted instead of the third.

Although the examples included with the ledger package indicate that cash disbursements can be posted, the cumbersome nature of the posting process will limit the number of eligibles for this procedure to those with only a few checks each month.

It does not take too many sessions to realize that interactive processing of cash disbursements is a time consuming and tiresome task.

Since this is a major source of general ledger information, the ledger user may well opt for another method of processing cash disbursements. The summary totals developed from that method can then be posted in the Radio Shack ledger once a month.

Despite the slowness of the posting procedures, I recommend Radio Shack's General Ledger to small business people who do not desire to customize the package. The user documentation and the "Bug Free" operation make it a good choice.

For the professional accountant, however, the limited number of balance sheet options creates problems in summarizing activity. The accountant will probably be forced to type up the balance sheet independently.

CRAS

Two of the most common books of original entry are the cash receipts and the cash disbursements journals. Combining these two journals creates an automated "checkbook." Such a system has been developed by The Bottom Shelf, Inc., in their Check Register Accounting System (CRAS).

CRAS is supplied on two diskettes, one of which is labeled, Security. To use the system, the diskette must be adjusted to include the 2.2 version of Radio Shack's operating system. The initialization procedure is brief, and once completed, need never be repeated.

Unfortunately, the Backup and Format utilities are not copied to the systems diskette. When needed, these must be extracted from a complete systems diskette.

Included with the diskettes is an eight page narrative, with several pages of illustration. While not up to the documentation level of the General Ledger by Radio Shack, the user documentation is generally complete and informative.

There is no program documentation of any sort. In fact, an attempt to view the program by hitting BREAK is fruitless. Use of another operating system to see the directory quickly reveals that each program is password protected and therefore can only be accessed by the start up procedures imbedded on the CRAS systems diskette. Clearly, The Bottom Shelf does not want its software customized.

Although it is possible to backup the systems diskette, it is not possible to backup the security diskette. In addition, the system will not run without the security disk being inserted during the startup procedure.

These procedures should discourage copying of the system and I applaud the author's effort. Good system software should be paid for and protected, otherwise, the flood of software for popular machines such as the 80 will quickly cease. It must be quite discouraging to spend many man-months developing and debugging a system only to see it copied as quickly as it is released.

The User Works Too

CRAS is plainly a very substantial effort. Once the package is running, an eleven element menu is presented. All system operations are driven off this menu and good use is made of messages prompting the user to make the proper response.

This is fortunate, because the user has a lot to do.

As I said, each month's activity is stored on a different diskette. Therefore, when using several month's data, the diskettes must be swapped to get the desired results.

As an example, a September bank reconciliation could include data from June, July, August and September. Each month's diskette would have to be inserted to develop an outstanding check list. Ugh!!! However, if the prompts are followed and patience is observed, the method works.

Although the CRAS system analyzes receipts and disbursement activities by account, it is not a general ledger. The system allows up to 60 separate accounts. When the system is initialized, it assigns account numbers from 69 down for income or deposit accounts and the unassigned remainder is considered expense accounts. These are numbered from 10 up.

The system can display the net cash balance on hand, on request.

CRAS supports only one checking account, which must be specified at initialization time. Users with more than one bank account must set up a separate system for each account. Since each system requires a minimum of 13 diskettes, this can rapidly become expensive.

The benefit of this approach is that each diskette can hold a large amount of data. The documentation does not indicate capacity because the amount of storage required for a month can be significantly affected by size of the comment or suspense file.

The Suspense File

The suspense file is an interesting system feature. This file reminds you of all commitments due during the calendar month. Data such as payroll tax payment dates, ad valorem tax due dates and the amounts of regular recurring bills can be stored here. The file can also be used to hold birthdates and other personal information.

When you must enter data into the system, the transaction entry subsystem is used. This subsystem assigns a serial number to each item just as the Radio Shack Ledger did. Unlike the ledger, there is no need to specify debits or credits. The operator simply specifies a check or deposit and the sign of the data entered is specified by the system.

Data entry is interactive and the following are entered when required:

- Payor or Payee
- Date of the transaction
- Amount
- Check #
- Purpose of the payment or receipt
- Up to 64 characters of notes regarding the transaction
- Account distribution data, a routine that facilitates distribution to multiple accounts by using arithmetic.

After all data is entered, the operator is asked to check its validity. A "no" answer to the validity question forces a total restart. There is no selective editing feature.

Each transaction is filed on the disk immediately. With the slow access time of the drive, this appears to last forever. However, if the time is used to gather papers for the next entry, it passes quickly.

A nice feature is the spooled check printing data. If the check number is not specified, the system places a flag in the transaction data. At a later time, the check printing subsystem can be activated. This subsystem prints checks either on continuous forms or individual check forms.

Unlike the General Ledger, reports are not generated until requested, but once requested, seem to print for hours. The basic report is the monthly activity report. This report includes a check register, check register notes, and account analyses, bank reconciliations and account distribution data.

With the exception of the account distribution report, all reports are easy to read and informative. The bank reconciliation subsystem in particular, takes the work out of reconciling a bank balance. Unfortunately, the account distribution report is not only unwieldy (each transaction is repeated 12 times) but is hard to use. Tracing data through this report is time consuming and the report should be redesigned to cut out all the unnecessary printing.

Despite several rough edges, the CRAS system is quite usable as a bank account handling system. It is in fact a complete voucher system. If a list could be generated of checks requested but not printed and a vouchers payable balance could be obtained, this could be quite useful in a commercial environment.

In succeeding months, I hope to continue this format of theory and program reviews. If desired, I will be glad to respond to correspondence concerning business systems and the use of the TRS-80 in particular. Write care of the "80 Accountant" to the Editors of 80 Microcomputing.

UNLIMITED 80's

by Sherry Smythe

In August I was chatting with Hy Siegel, publicity manager at Radio Shack HQ in Fort Worth, about the multitude of different TRS-80 applications. The idea of gathering information about these applications and presenting them in *80 Microcomputing* grew into a full-fledged project, taking me all over the country.

In Denver I visited with John and David Young, who run Colorado Aerial Photo Service, CAPS, a company that supplies aerial photos to dozens of clients—from realtors to map makers.

As aerial photos are a permanent, reproducible record of "the ways things were," they are also especially valuable in legal cases involving land disputes.

John and Dave are using a TRS-80 with 48K to help run their business.

Their largest and most ambitious program will store and access information on each of the thousands of negatives stored on over 150 rolls of film dating back to 1948. Each roll is up to 250 feet long and contains up to 280 9" x 9" frames. Each negative will be indexed by scale, the latitude and longitude of the center point, the date and flight data.

Using this program, they can enter a geographical location and a list of negatives covering that area will be output. Once 30 years of data has been programmed, successive flights can be entered and all the information can be accessed almost instantaneously.

John and Dave have had only minimum exposure to computer programming prior to acquiring their TRS-80. They've added an expansion interface, a lineprinter and two floppy mini-disc drives.

John Young and Sherry Smythe flank a copy of CAPS' photomosaic of metro Denver, used by NASA to verify program data.

They are sandwiching their self-education on computer language and programming techniques between carrying on business as usual at CAPS and other projects like improving and designing lab equipment and remodeling a small house for CAPS offices.

During my visit their programs designed to help produce photos, handle bookkeeping chores, supply market information and data storage were not all completed. When they are, Instant Software will review them for possible mass-marketing.

CAPTAIN 80

by Bob Liddil

Here's Captain 80 sitting in front of a 32K dual disk drive TRS-80 interfaced to an IBM Selectric. The Instant Software building is nearly empty now and few people take notice of me sitting here in green tights and yellow cape. A few of the local townsfolk did gape a little, but that was because the irridescent red 80, with slash lightning bolts, glows in the dark.

The quiet of the evening is the only time I have to use any of the nearly $125,000 worth of computer hardware located in the Instant Software complex. During the daytime the whole place is overflowing with hotkey programmers and brilliant hardware technicians. With these guys, constantly reviewing software and hooking up new gadgets to make computer life easier, a mild mannered copywriter for the advertising department—ye olde secret identity—has very little daylight line time.

Dreams really do come true. Just two weeks ago I was living in Southern California, selling color murals of local lakes and doing a little programming on the side. Being an avid follower, of Microcomputing Magazine, I caught Wayne Green's REMarks about careers at Microcomputing and Instant Software.

Could I get a job working for a great computer organization in the middle of scenic New Hampshire? Nah! Not me!

Wayne wasn't bluffing. One letter, one phone call, five grueling days on the road and one flat tire later; I, my van and my TRS-80 arrived in one of the prettiest places I've ever seen. They put me to work the next day.

Before I had even had a chance to unpack my Captain 80 outfit, I was on the road to Philadelphia heading for the computer show. I saw some really dynamite things there. Percom, for example, had a neat little color graphics board for the TRS-80 displaying supurbly clear full-spectrum video.

This graphics display was a pleasure to behold after two black and white years before the keyboard. But that wasn't all. The same guys had another TRS-80 interfaced to a Texas Instruments Speak and Spell! Push a key (letter or number) and the speak and spell voice pronounced it. No words or phrases yet but they're working on it.

The thing that impressed me was how it SPOKE! Unlike the R/S speechbox or the computalker of S 100 fame (both of which sound like machines), the speak and spell sounds like Brother George teaching Sunday school to six year olds. It's clean, clear, crisp and a heck of a lot cheaper than $400+. They say it'll be fully programmable to the 80 very soon.

Some of the software on display, besides our own venerable Instant Software, was a real trip to play. Without a doubt, the hit of the younger set was Bee Wary; Leo Christopherson's outrageously funny game featuring a tarantula and a bee. What registered on the faces of the people I saw playing Bee Wary was nothing less than pure delight.

What first attracted my attention to the Acorn software booth were the screams of laughter coming from a crowd of hardcore computerists witnessing the death throes of a mortally wounded electronic tarantula. The tarantula was shaking a tentacle at a smirking bee while uttering spider noises from a conveniently placed audio unit. Swearing revenge and bemoaning its fate, the tarantula finally expired after making sure it had gotten the last word. I recommend this program to anybody under a hundred.

A lot of companies were offering tapes. I scraped some up for review, including Pirates Cove from the Scott Adams Adventure series. There were a number of companies offering Adventure, so the tape ought to be easy to find.

Pirate's Cove was an agonizing and frustrating puzzle. I played it on the sixth floor of the Philly Hilton for hours on end. It gave up clues grudgingly and tied me in knots trying to get them. I LOVED IT!! There was real challenge to this program, unlike some lesser things I've seen. The frustration level is a part of the game. After five evenings of concentrated play in the ISI lab, Pirate's Cove was finally solved. Scott, wherever you are, I hope a mongoose perches on your window and throws acorns at you! ■

80 INPUTS

Timed Input Subroutine

There is a short subroutine that can be implemented on the TRS-80 to allow timed input. Timed input could be put to very good use in teaching programs. *Microcomputing* published a few math-teaching programs and a timed input of about two seconds would be an added encouragement for the student to be prompt with his answer. Many games could use a timed input for a more interactive game.

The routine below shows that the timed input is actually very simple. It is surprising that more people have not already run across this in their programs. It uses the INKEY$ function in a FOR-NEXT loop. The value of N determines the amount of time allowed for input. About one second passes for every multiple of 100 that is in N.

```
10  N = 200 : REM About 2 seconds, but changed easily
20  FOR I = 1 TO N
30  A$ = INKEY$ : IF A$ = "" THEN NEXT I : PRINT
    "TIME" : STOP
40  PRINT "ANSWER IS JUST IN TIME"
50  END
```

A little bit of trivia about the TRS-80: Try running a program, and while it is in execution press SHIFT, down arrow and 'A' (all in that order), holding each down while the next is pressed. When all three keys are pressed hit the BREAK key and execution of the program is terminated. The combination of these keys, sends out the same strobe-combination onto the keyboard as the single BREAK.

The CPU cannot determine that the BREAK key was not actually depressed. Of course this isn't practical for ordinary purposes since simply hitting BREAK is easier, but it is surprising to see at first.

Greg Perry
6104 E. 48th St.
Tulsa OK

TRS-80 More Memory Free!

Yes that's right, more for free.

If you, like many of us, have updated your system to 16K, you have 4K chips doing nothing but nagging at you. After asking several people, Why can't those 4K chips be plugged into the interface? The reply in all cases was: They won't work!

Well, I half accepted this.

But the question was always there: Why won't they work? After all, you just plug in 16K chips in place of the 4K.

Finally when the newness wore off my system I plugged the 4K chips into my interface which had no RAM installed. I used sockets Z9 to Z16 and came up with an increase of 8K.

Don't ask me how 8K from 4K chips. I don't know and I'm not going to look a gift horse in the mouth; he might have bad breath.

All the functions work normally. Memory size 40960 & ?MEM comes back with 23814. I feel sure this will work if you have 32K. Plug the 4K into Z1 to Z8. If that doesn't work, then plug 4K in Z9 to Z16 and 16K in Z1 to Z8.

Good luck!

Joe Pugliesi
Burkburnett TX

Cool Your System Crash

I operate a 48K, 4 disk TRS-80 model. Since I perform inventory policy simulation in the system, and the programs take about 8 hours to run, system problems were hampering my work. The TRS-80 runs continuously, and I was experiencing system crashes every two days or so. All tests on the system were negative, and the blame was placed on "a noisy electrical environment."

I solved my problems in the following fashion: I procured myself a fan and set in on "low" so that the airflow cooled the whole system, and I installed a line filter to the power lines. I bought the 10 ampere filter from R.B. Co., 57 Thomas Olney Commons, Providence, RI 02904, for $23.50, which included postage and handling. Since these introductions, the system runs very cool and has not had any problems in over 30 days of 24 hour per day operation. I routinely perform software checks on the hardware.

I hope this tidbit helps other TRS-80 users.

George S. Lechter
Boston, MA

Radio Shack's TRS-80 Cassette Modification

Recently, while on vacation in the Midwest, and naturally having brought along my TRS-80, I stopped in a local Radio Shack store to purchase some newly distributed software.

While there, I happened to see the May issue of Radio Shack's new TRS-80 newsletter. It mentioned a free cassette modification for the Level II computers. I was thinking I would have it put in when I got back home.

In the meantime I purchased Micro-chess and the new Invasion Force, both machine language programs. When I attempted to load the Invasion Force program, it wouldn't load, even after much trying. I got the Micro-chess program to load after about a ½ hour. I decided to take the Invasion Force program back to Radio Shack where the manager willingly exchanged the tape for me. When I tried to load this tape the same thing happened, but after about 3 hrs. I finally got the tape to load. Everytime I use a different tape though I've got to fiddle with the recorder volume for at least 15 minutes to get the tape to load.

About a week later I was in Denver visiting a cousin who also has a TRS-80 Level II. When we tried to load some of my tapes into his computer they wouldn't load initially. After a considerable time we finally found the right volume on his cassette recorder and one of the tapes loaded. But every time we changed tapes we had to find a new volume level for the recorder. This happened using machine language tapes as well as BASIC tapes.

After returning home to Connecticut, I took my TRS-80 keyboard in to the local Radio Shack store and asked to have the cassette modification installed. The unit was shipped off the next day. One week later I stopped in at the store and picked up the unit which had returned from Boston, the regional repair facility.

After trying out the computer I was really impressed. Using the CTR-41 recorder I was able to use volume levels as low as 3 or as high as 10 to load tapes, whereas before I could use only 7 or 8.

The modification consists of 2 IC's, 2 diodes and a resistor all mounted on a 1½" × 1½" board located just to the right and below the keyboard space bar.

I think Radio Shack is extremely wise in putting out this free modification.

I would recommend that anyone that doesn't use a disk system get this modification. It sure saves on hair and aspirin.

Harold W. Smith
RFD #1 Box 213
Gales Ferry, CT

80 APPLICATIONS
by Dennis Kitsz

Five years ago, when electronics hobby magazines heralded the birth of the microcomputer age, I was still bargaining with a New Jersey electronics designer and former musician, whose "digital sequencer" was, he told me, going to change my life as a composer.

The price was too high, I said. With the gestures of a sideshow magician, hands gliding over buttons and knobs to the visual music of blinking LEDs, he made my synthesizer bark and growl, play a rag and sound sirens. I placed my order.

Two years later, when the long-delayed sequencer succumbed to "production problems" for the last time, I renewed my electronics magazine subscriptions, recommencing my search for an electronic music controller. That search led to the purchase of a TRS-80 from a cluttered but trusted Radio Shack in eastern Pennsylvania. In the spring of 1978, I felt like a pioneer; no matter that the '80 came complete with keyboard, cassette, monitor and interactive human language. I had a computer at a time when computers were still Big and Elsewhere, and when the word itself invoked tantalizing images ranging from entrepreneurs of starched-and-striped IBM to Stanley Kubrick's emotional HAL of "2001" fame.

It grew from the computer to My Computer, for I had taught it many things. Even when it was turned off, it seemed quietly wise. With growing confidence, I took soldering iron in hand to build a latched digital-to-analog converter—finally, the TRS-80 was to become the sequencer for my voltage-controlled synthesizer.

Well, it wasn't so easy. BASIC was ponderously slow, and available memory disappeared in a dozen measures. Friends were amazed at my computer transcription of Bach's 21st Prelude, as I became quietly furious. One of us—this so-called sophisticated computer or myself—was not living up to expectations. I soon learned that the TRS-80 could make my synthesizer resound more professionally than a dozen virtuoso performers . . . but I was a rank amateur at playing the computer!

Since that time, I have learned machine language for its speed and exonomy; have modified the computer to satisfy my eyes and ears; and have added devices of my own creation to soothe my wallet. Its schematics have become a familiar roadmap. That ideal music sequencer is still incomplete, but when I finally turn on the power, I know it will perform precisely as I wish.

What does this specialized need have to do with the general use of the TRS-80? Simply stated, microcomputers, from the S-100 "boxes-of-boards" approach to the ready-to-run TRS-80, can provide powerful and challenging services to our lives. Those services can range from entertainment through serious business and academic application to the creation of high art. And the TRS-80 is ideally suited to this multiplicity of choices.

Did you know, for example, that the '80 can produce sound effects with no more hardware than an amplifier? That the circuitry for a higher-speed machine is already in place? That a black-on-white video image is a one-dollar modification? That there is enough space for 12% or more of your own ROM? Or that there are two additional alphabets present in the TRS-80 character generator?

On the other hand, can you imagine the TRS-80 monitor controlling all the electrical systems in a dozen homes? That it can dial your telephone? Chart your biorhythm? Time your photography?

In the coming months, this column will present unusual applications of the TRS-80 in everyday and not-so-everyday life. Hardware and software extensions and fixes will be described, and some of the fascinating inner workings of this first computer-for-the-people will be revealed. I welcome your questions and suggestions, addressed to me at *80 Microcomputing*, or at my home address, *Roxbury VT* 05669.

CLUB 80
by Ross Wirth

What does every person reading this magazine have in common? Answer: an interest in the TRS-80. As such the readers of this magazine form the largest user group in the world!

Every time I attend a user group meeting I am amazed by the amount of information made available to me. This is the type of material I want to publish in this column every month, but I'll need your help.

Send your comments or opinions to Ross Wirth, 15906 E. 96 St. N., Owasso, OK 74055. Please include an SASE for an individual reply.

If You've Got a View of Your Own Let's Hear It!

Software Reviews

Currently, hundreds of TRS-80 programs are being sold by as many companies and individuals. How do you pick the good from the bad?

I'd like to use this column to pass along comments on various programs, so you can make a more informed purchase.

Let's start off with a definite purchase recommendation: Android Nim. Nim, alone, is a good game, but adding some of the best animation ever is frosting on the cake.

Using INKEY$ for data input allows the androids to be in constant motion looking up, down, left and right. (They blink their eyes too!) When the number of androids in a given row to be removed is input, the master android counts the androids remaining in that row. If there are enough, the master android nods yes, pulls a gun and zaps each one in turn, while all other androids look on. If the data input is invalid the master android shakes his head no and the action goes on unaltered.

Would-be programmers of games with the computer as opponent should take a lesson from Android Nim. In Nim the player who goes first controls the game and can force a win. The computer sometimes makes a mistake and lets the knowledgeable human have a chance. The probability of a mistake is low but allows you a chance to turn the tables on the computer.

Rumor has it that a new version is under development using sound effects while zapping the androids. Adding sound effects might be difficult because of the lack of memory in a 16K system. This is a large program and uses about all 16K. (I have a line printer driver that resides in upper memory and had to delete most of the instructions to get enough array space to run the program.)

From the enthusiastic response to Android Nim at a local computer show it seems I am not alone in saying this program should be high on your list of software to purchase.

Balance Your Own Checkbook

For every Android Nim, though, there are other programs that should never have been written.

Radio Shack's Personal Finance Program is a definite loser. My bias against this program starts with the feeling that a computer should be used only where it does a job better than any other method. *I* can balance my checkbook faster than it takes to enter the data, let alone loading the historical data from tape (slooooow)!

Having the data on tape rather than disk, is another red flag. I prefer to do without, than tolerate slow I/O.

However, if you insist on a computerized budget and checkbook balancing program, keep looking. This 4K level I package would only be a disappointment.

Disagree with my opinion? Send me your comments on these programs. If you have strong feelings one way or the other about any program you purchased, let me know so we can pass the word along. By concentrating our buying power on the good programs, the losers will be removed from the market.

Hardware Reviews

A Level II 16K TRS-80 can survive all kinds of line hash and electrical interference. Add an expansion interface, more memory and a disk and your system can suddenly become very sensitive. A local company had their system in a room with a couple of typists. Everything worked fine until they upgraded to DOS. When a typist turned on her IBM typewriter, there was a good chance they'd be rebooting their DOS and lose their existing work! When the problem occurred during a disk save, it was worse. With the program being saved in a condensed format, a one bit loss can change an entire command. Believe me, you can ruin an entire program very quickly. A couple of line filters cured all problems.

Other companies have installed separate circuits for their computer and one, located in an industrial area, has installed an isolation transformer. In any business system the software and data is usually irreplaceable. Taking such precautions—recommended by Radio Shack—is very good insurance.

An interesting side note is a comment made by a customer service rep from Fort Worth. He reported that radio frequency developed by a bad florescent light can cause similar interference. Has anyone had this problem?

Do you have an application using Radio Shack's Voice Synthesizer or Voice Recognition modules? If so, drop me a line.

Application Reviews

Business applications are where the real benefits of the TRS-80 are realized. Let's take a quick look at how some businesses are saving money with their new computer.

Small computers are for small businesses, right? Wrong! Many large companies operate with minicomputers and a few have purchased microcomputers, including the TRS-80. In one company with millions of dollars already invested in large computers and a multimillion dollar systems payroll, a TRS-80 application is alive and well. The TRS-80 budgets operations at a dozen manufacturing plants.

The principal reason for going to the microcomputer is its independence. The user now has hands-on control of his data and can quickly make program modifications without fighting the bureaucratic red tape found in any large organization.

One program is used to do all the calculations that had previously been done by hand. A second program consolidates data from two or more plants. The entire development time was less than one month, complete with education of all personnel. Break even is expected within the first year.

Faster and more accurate calculations cannot be stressed too much. In the program debug phase a large number of errors were discovered. Almost all the errors, were found to be in the manual calculations. In the past these errors may have gone undetected.

To prevent errors, the program validated entered data by checking it against known ranges. Furthermore, all data was entered as a percentage and these were added together to verify that they equalled 100%.

A report was also available for cross validation with the data received from the plants. Total investment: Level II, 48K, 1 disk drive and a Centronics printer. Program development cost approximately $3000.

Software for the 80's

Welcome to the first issue of *80 Microcomputing*. We think that Instant Software has the best cross-section of TRS-80 programs now available. Look over these pages, make your own comparisons, and decide for yourself. After you've made an intelligent and informed decision, use the coupon to get a $1 discount on our software. Happy computing!

LEVEL II

MODEL ROCKET ANALYZER AND PRE-FLIGHT CHECK Let your TRS-80 help you enjoy the fast-growing hobby of model rocketry. The complementary programs included are:

•**Model Rocket Flight History Prediction** – This program will compute the flight characteristics for almost any model rocket. Engine and body tube data included covers Estes, Centuri, Flight Systems, A.V.I. Astroport, C.M.R., and Kopter products.

• **Weather Forecaster** – Before you launch your rocket, get an up-to-the-minute weather forecast. Just enter your location, elevation, average temperatures for January and July, and barometric pressure. You'll be the short-range weather forecaster for your area.

For a successful launch, you'll need TRS-80 Level II 16K. **Order No. 0024R $7.95.**

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE Buck Rogers never had it so good. Engage in extraterrestrial warfare with:

•**Ramrom Patrol** – Destroy the Ramrom ships before they capture you.

•**Tie Fighter** – Destroy the enemy Tie fighters and become a hero of the rebellion.

• **Klingon Capture** – You must capture the Klingon ship intact. It's you and your TRS-80 Level II 16K battling across the galaxy. **Order No. 0028R $7.95.**

PERSONAL BILL PAYING

NOTE: This package can take the headaches and/or penalties out of paying your bills.

In a business office the accounts payable (bills) are usually paid on or immediately before their due date. That way, the payer gets the fullest use of his money without incurring penalties for being behind in paying his debts. Now you can take advantage of this system for your monthly bills, letting your TRS-80 do all the drudgery and record keeping.

This useful package provides a computerized list of all your bills and payments. You can access as many as 22 accounts, all of which can be named – up to 15 characters per name. Each account is listed by number, amount owed, due date, and present activity.

Don't confuse this system with a "checkbook" program. The functions of this package are threefold: (1) to monitor your bills; (2) to order payments most effectively; and (3) to make historical comparisons of individual accounts or specific months.

After you load the program, it displays a menu of 11 activities. They include:

Build and Maintain Files
List All Accounts
List Current Accounts
Make Payment(s) to Account
Enter New Bill to Account
Display Payment History of Individual Account (includes date paid, check number, and 12-month total)
Display Payment History of Selected Month
Delete Account
Delete Prior Month's Payment
Save File on Tape
Input File from Tape

After you have updated the records by entering new bills, paying bills, or changing the accounts, you can save all the information on data tape. This data tape will then be input for the next time you use the package. Maybe it can't make paying bills all fun and games, but it should relieve some of the agony. Level II 16K required. **Order No. 0103R $7.95.**

FINANCIAL ASSISTANT Compute the figures for a wide variety of business needs. Included are:

•**Depreciation** – This program lets you figure depreciation on equipment in five different ways.

•**Loan Amortization Schedule** – Merely enter a few essential factors, and your TRS-80 will display a complete breakdown of all costs and schedules of payment for any loan.

•**Financier** – This program performs thirteen common financial calculations. Easily handles calculations on investments, depreciation, and loans.

•**1% Forecasting** – Use this simple program to forecast sales, expenses, or any other historical data series.

All you need is a TRS-80 Level II 16K. **Order No. 0072R $7.95.**

TRS-80 UTILITY II Let Instant Software change the drudgery of editing your programs into a quick, easy job. Included in this package are:

•**CFETCH** – Search through any Level II program tape and get the file names for all the programs. You can also merge BASIC programs with consecutive line numbers into one program.

•**CWRITE** – Combine subroutines that work in different memory locations into one program. This works with BASIC or machine-language programs and gives you a general checksum.

This package is just the thing for your TRS-80 Level II 16K. **Order No. 0076R $7.95.**

CARDS This one-player package will let you play cards with your TRS-80 – talk about a poker face!

•**Draw and Stud Poker** – These two programs will keep your game sharp.

•**No-Trump Bridge** – Play this popular game with your computer and develop your strategy.

This package's name says it all. Requires a TRS-80 Level II 16K. **Order No. 0063R $7.95.**

SPACE TREK IV Trade or wage war on a planetary scale. This package includes:

•**Stellar Wars** – Engage and destroy Tie fighters in your attack on the Death Star. For one player.

•**Population Simulation** – A two-player game where you control the economy of two neighboring planets.

You decide, guns or butter, with your TRS-80 Level II 16K. **Order No. 0034R $7.95.**

DOODLES AND DISPLAYS II Wait until your children get hold of this package:

•**Doodle Pad** – Draw pictures and save them on cassette tapes.

•**Symmetrics** – An electric kaleidoscope that changes from black to white and back again. It's almost hypnotic!

•**Drawing** – Like Doodle Pad, but for the serious artist. Over 40 user commands!

•**Random Pattern Display** – The computer does the drawing, but those with itchy fingers can tamper.

•**Mathcurves** – Bring those geometry lessons to life. Six different geometrical curves on the screen of your TRS-80.

•**Rugpatterns** – Yes, it does design rug patterns; and with a choice of user or computer control, it can do a whole lot more.

For the Level II 16K TRS-80. **Order No. 0042R $7.95.**

BOWLING LEAGUE STATISTICS SYSTEM This package is the answer to the prayers of harried bowling league scorekeepers. The Bowling League Statistics System will keep a computerized list of league data, team data, and data for each bowler. It is extremely flexible and has a total of 16 different options to let you modify the program to suit your league's rules. The program is very easy to use and has extensive "built-in" aids to help you along. Requires TRS-80 Level II 16K. **Order No. 0056R $24.95.**

AIRMAIL PILOT

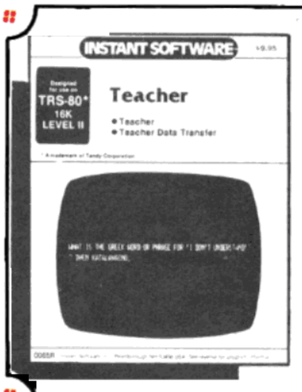
July, 1922: The newly formed Postal Air Service needs a pilot to fly the mail on the perilous Columbus to Chicago run. The pilot must have lightning reflexes, excellent flying skills, and no regard for danger. We think that *you* are right for the job. You'll be flying the "Jenny," an open-cockpit biplane, which has a fuel capacity of only twenty-six gallons. If you don't refuel often, you may run out of gas, and there are no gas pumps at 5,000 feet.

But fuel consumption is only one of your worries. You must also beware of electrical storms, severe crosswinds, hail, deadly downdrafts, and navigation errors. It will be a race against time, as well as a struggle against the elements, so there's an onboard clock to time your flight. The excellent graphics and constant action will keep you glued to your keyboard.

You'll have a (flying) circus of fun with Airmail Pilot from Instant Software.

Level II 16K required. Helmet and goggles optional. **Order No. 0108R $7.95.**

TEACHER



What do you need to learn? Would you like to know all of the cranial nerves? Electronic color codes? Civil War battles? Signs of the zodiac? Whatever your subject matter, the Teacher package can help you learn it. You simply input up to twenty questions and answers at one time. Next, review the material, and then take the test until you have your lesson down pat. The program gives you up to three hints per question and even offers graphic rewards for children, all at your discretion. All the information can be saved on cassette tape for reuse.

This package also contains the Teacher Data Transfer program, which allows you to combine several tests on one tape. That means you can learn a number of lessons sequentially without changing tapes.

Teacher is an effective instrument for anyone who needs to learn a lot of material in the shortest possible time. For decades leading educators and computer scientists alike have been promoting the future role of computers in education. Now you and your family can reap the benefits of computer-assisted instruction in your own home. The program is furnished with a blank data cassette tape. You'll need a TRS-80 Level II 16K. **Order No. 0065R $9.95.**

LEVEL II

DEMO II The company that brings you more programs for the dollar is proud to offer Demo II, an extraordinary package that contains programs to suit your every mood:.

•**Tic-Tac-Toe** – Fun for the whole family! You all know the rules, and this version gives you three different levels of difficulty, one of which is sure to suit everyone in the family.

•**Time Trials** – You won't have to leave your house to experience "pre-race" excitement. It's you against the clock as you maneuver your car through the curves, chutes, and chicanes of the computerized course.

•**Maze** – Somewhere within a ponderous maze, the secret home square waits for you to uncover its presence. If you're not alone, your computer will be happy to offer a two-player version so that you can both race to uncover your respective squares first! Amazing.

•**Hangman** – How many people have grown up playing different versions of that old grade school standby, hangman? If you've never played the game on a computer, you're in for a special treat! You or the computer will supply the word. With each wrong guess, the poor hangman's figure grows – and only you can spare him!

•**Wheel of Fortune** – "There's one born every minute," or so goes the old saw. In this simulation of the carnival wheel of fortune, you have your choice of the regular or the "crooked" version, where you can't help but win. The casino will even give you the keys to the place!

•**Hurricane** – OK, all you disaster buffs, here's a program that will let you chart the path of oncoming hurricanes anywhere in the world, using data available from the National Weather Service.

•**Bugsy** – Sure, everybody talks about computer bugs, but how many people have ever seen one? It's you against the computer in this game, with the computer rolling the dice, to see which one gets to add another part to his Z-80 Bug. The first one to complete his "curse of computing" wins.

•**Horse Race** – It isn't every day you get to see a horse race . . . at least not until now! Up to 100 bettors can cheer their horses to the finish line with their choice of win, place, or show. For the TRS-80 Level II 16K. **Order No. 0049R $7.95.**

BASIC AND INTERMEDIATE LUNAR LANDER Bring your lander in under manual control. The BASIC version is for beginners; the Intermediate version is more difficult, with a choice of landing areas and rugged terrain. For one player with a TRS-80 Level I 4K, Level II 16K. **Order No. 0001R $7.95.**

DEMO III This is the big one: big on value, big on fun – the perfect package for the beginner and old hand alike. Check out this list of programs!

•**Race 1** – It's you against the clock as you careen around the track in this simulation of a high-speed car race.

•**Target UFO** – Rack up a big score by destroying all the UFO's in the shortest possible time.

•**Life** – Experiment with population density factors in this simulation of the life cycle of a colony of bacteria.

•**Phone Number Converter** – Let your computer figure out clever words for all those hard-to-remember phone numbers.

•**Biorhythm** – Fact? Or fantasy? Who can say for sure? In any case, you and your friends will be able to plot your biorhythmic curves whenever you want!

•**Graphics Program** – No user commands, just sit back and enjoy as your TRS-80 demonstrates its artistic abilities.

•**Race 2** – After you've mastered Race 1, you'll be ready for this more advanced version – with a choice of five different tracks!

•**Horse Race** – Up to nine players can place their bets and watch the ponies run. The computer will keep track of the winnings.

•**Drawing Board** – Your TRS-80 supplies the "pencil and paper," and you supply the ability. Messages or drawings may be stored on cassette for later unveiling.

•**24-Hour Clock** – That's right, this program allows your computer to act as a digital timepiece. Perfect for sporting events. For the TRS-80 Level II 4K and Level II 16K. **Order No. 0055R $7.95.**

LEVEL I AND II

GOLF/CROSS-OUT Have fun with these exciting one-player games. Included are:

•**Golf** – You won't need a mashie or putter – or a caddie, for that matter – to enjoy a challenging 18 holes.

•**Cross-Out** – Remove all but the center peg in this puzzle, and your neighbors will call you a genius.

You'll need a TRS-80 Level I 4K, Level II 16K. **Order No. 0009R $7.95.**

VIDEO SPEED-READING TRAINER

You can increase your reading speed and comprehension with this package. It uses the principle of the tachistoscope, a device that teaches by displaying images for a fraction of a second. These programs can train you to recognize words and phrases quickly, so that your everyday reading becomes an uninterrupted process.

With this three-part package, you can learn to recognize numbers, letters, words, and phrases. You start at your present level of skill by choosing the number of prompts to be shown and the length of time they will be visible. After the cue is flashed on the screen, you type back what you saw. If your response is incorrect, there is immediate feedback as the correct answer is displayed for several seconds. All cues are presented randomly so that the exercise questions cannot be learned in sequence.

The computer will monitor your progress and will automatically speed up presentation as you improve. Conversely, should you miss more than half of the questions, the speed will drop to an easier level.

The speed range is from 10 to 100, so you can readily determine your rate of progress. To increase your throughput, you'll need a Level II 16K. **Order No. 0100R $7.95.**

AIR FLIGHT SIMULATION Turn your TRS-80 into an airplane. You can practice takeoffs and landings with the benefit of full instrumentation. This one-player simulation requires a TRS-80 Level I 4K, Level II 16K. **Order No. 0017R $7.95.**

OIL TYCOON Avoid oil spills, blowouts and dry wells as you battle to become the world's richest oil tycoon. Two players become the owners of competing oil companies as they search for oil and control their companies. Requires a TRS-80 4K Level I or II. **Order No. 0023R $7.95.**

BOWLING Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. For the TRS-80 Level I 4K, Level II 16K. **Order No. 0033R $7.95.**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. This program requires a TRS-80 Level I or II 16K. **Order No. 0043R $7.95.**

HAM PACKAGE I This versatile package lets you solve many of the problems commonly encountered in electronics design. With your Level I 4K or Level II 16K TRS-80, you have a choice of:

•**Basic Electronics with Voltage Divider** – Solve problems involving Ohm's Law, voltage dividers, and RC time constants.

•**Dipole and Yagi Antennas** – Design antennas easily, without tedious calculations.

This is the perfect package for any ham or technician. **Order No. 0007R $7.95.**

ELECTRONICS I This package will not only calculate the component values for you, but will also draw a schematic diagram. You'll need a TRS-80 Level I 4K, Level II 16K to use:

•**Tuned Circuits and Coil Winding** – Design tuned circuits without resorting to cumbersome tables and calculations.

•**555 Timer Circuits** – Quickly design astable or monostable timing circuits using this popular IC.

•**LM 381 Preamp Design** – Design IC preamps with this low-noise integrated circuit.

This package will reduce your designing time and let you build those circuits fast. **Order No. 0008R $7.95.**

BEGINNER'S BACKGAMMON/KENO Why sit alone when you can play these fascinating games with your TRS-80?

•**Backgammon** – Play against the computer. Your TRS-80 will give you a steady, challenging game that's sure to sharpen your skills.

•**Keno** – Enjoy this popular Las Vegas gambling game. Guess the right numbers and win big.

You'll need a TRS-80 Level I or II. **Order No. 0004R $7.95.**

SPACE TREK II Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power, and warp drive. It's you alone and your TRS-80 Level I 4K, Level II 16K against the enemy. **Order No. 0002R $7.95.**

BUSINESS PACKAGE IV Business Package IV gives you, the businessman, a superb tool to help you make those important decisions. This package includes:

•**Business Cycle Analysis** – This program isn't a crystal ball, but it can show you your business's expansion and contraction cycles. You can plot any aspect of your business on a graph and see, in black and white, just what's happening. This program will give you access to information you couldn't get before.

•**Financial Analysis** – Would you like a financial assistant who could instantly give you the figures for almost any kind of investment? Financial Analysis can handle annuities, sinking funds, and mortgages, and compute bond yield and value. You'll have the facts you need at the tips of your fingers with this program.

Included in the package is one specially marked blank data cassette for use in storing essential business data.

Business Package IV, with its combination of analytic functions and convenience features, is an invaluable asset for any businessman. All you need is a TRS-80 Level I 4K or Level II 16K. **Order No. 0019R $9.95.**

LEVEL I

HEXPAWN/SHUTTLE CRAFT DOCKING/SPACE CHASE/SEA BATTLE This four-game package is sure to provide hours of fun for the whole family.

•**Hexpawn** – Turn your TRS-80 into a model of artificial intelligence by playing a simple game.

•**Shuttle Craft Docking** – Land your shuttle craft on the starship – even through varying gravity fields!

•**Space Chase** – Seek out and destroy the enemy delta that's hidden in the star field.

•**Sea Battle** – You must find and destroy the enemy fleet.

This package requires a TRS-80 Level I 16K. **Order No. 0041R $7.95.**

BUSINESS PACKAGE III This package can change your TRS-80 into a full working partner for any businessman:

•**Inventory** – Maintain a computer-based inventory for a constant inventory system.

•**Commissions and Percentages** – Let your computer figure out markup and discount calculations, sales tax and more. This is a perfect time-saving package for any small business.

For the TRS-80 Level I 4K. **Order No. 0061R $7.95.**

HOUSEHOLD ACCOUNTANT

•**Budget and Expense Analysis** – Impose order on your tangled financial affairs. This program makes provision for twenty-seven expense categories and three income sources. Budgets can be reviewed monthly, quarterly, and yearly. Dollar and percentage comparisons can be made between budgets, months, and year-end totals, so you can see where your money is going.

•**Life Insurance Cost Comparison** – Learn how your choice of life insurance policy can save you money. This program can help you decide between term or dividend-paying whole life policies. It will also compare within categories, when, for example, two whole life plans vary in dividends and in cash values. You can store and display up to six different value comparisons. Requires a Level II 16K. **Order No. 0069R $7.95.**

LEVEL I

SPACE TREK III Let yourself go to the far ends of the solar system—and beyond. This package includes:
•**Stellar Wars**—Shoot down the Tie fighters and destroy the Death Star.
•**Planetary Lander**—Land your spacecraft and plant your flag across the solar system.
These one-player games require a TRS-80 Level I 4K. **Order No. 0031R $7.95.**

FUN PACKAGE I Why call it "Fun Package"? Judge for yourself! This entertaining package includes:
•**Rocket Pilot**—Flying it is easy—it's the landing that's tough!
•**Paper, Rock, Scissors**—It's the time-honored game just as you remember it, played against your TRS-80.
•**Hex I**—Just when you master this puzzle game, the computer will increase the difficulty.
•**Missile Attack**—Use your missiles to protect your city from jet attack.
Requires a Level I 16K TRS-80. **Order No. 0037R $7.95.**

TYPING TEACHER Tired of being a two-fingered wonder at the keyboard? Do you tell your friends that you use the "Columbian" method, because you discovered it yourself? Well, you can change from that awkward hunt-and-peck approach and start typing like a professional with this practical instructional program from Instant Software.

This seven-part program turns your TRS-80 into both typewriter and instructor as it takes you from initial familiarization with the keyboard through the typing of words and phrases and on to complete mastery of the keybaord.

On-screen diagrams show you the correct placement for fingers, and the program's practice sessions will quickly improve your coordination. In no time at all you'll be typing in that smooth, free-flowing style you've always admired. For the 4K Level I TRS-80. **Order No. 0099R $7.95.**

DOODLES AND DISPLAYS I Here's a mixed bag of programs that's sure to entertain:
•**Doodle Pad**—Draw pictures and save them on cassette tapes.
•**Symmetrics**—Turn your TRS-80 into a kaleidoscope.
•**Video Display**—Follow the bouncing cursor as your TRS-80 draws its own pictures.
•**Mathcurves**—Bring those geometry lessons to life as the computer draws six different geometrical curves.
•**Rugpatterns**—A never ending stream of symmetrical patterns that's sure to spark your imagination.
All you'll need is a 16K Level I TRS-80. **Order No. 0030R $7.95.**

CAR RACE/RAT TRAP/ANTIAIRCRAFT Enjoy these challenging, fun-filled programs:
•**Car Race**—You and a friend can race on a choice of two tracks.
•**Rat Trap**—Trap the rat in his maze with your two cats. For one player.
•**Antiaircraft**—Aim and shoot down the enemy airplane. Requires Level I 4K TRS-80. **Order No. 0011R $7.95.**

KNIGHT'S QUEST/ROBOT CHASE/HORSE RACE This varied package of one-player games will give you hours of fun.
•**Knight's Quest**—Battle demons to gain treasure and become a full-fledged knight.
•**Robot Chase**—Destroy the deadly robots without electrocuting yourself.
•**Horse Race**—Place your bet and cheer your horse to the finish line.
These programs require a TRS-80 Level I 4K. **Order No. 0003R $7.95.**

CAVE EXPLORING/YACHT/MEMORY These three programs are not only fun, but stimulating as well:
•**Cave Exploring**—Search for fabulous treasures as you explore the magic cave. For one player.
•**Yacht**—A two-player game of strategy and chance. The computer rolls the dice and keeps score.
•**Memory**—Two players can pit their memories in this program based on a popular television show. You'll need a TRS-80 with Level I and 16K. **Order No. 0010R $7.95.**

STATUS OF HOMES/AUTO EXPENSES Two long-awaited programs that have got to save you money at work or in the home:
•**Status of Homes**—This program will allow you to keep track of all the expenses involved in building one house or an entire subdivision.
•**Auto Expenses**—Find out *exactly* what it costs you to drive your car or truck.
These programs require a TRS-80 Level I 4K. **Order No. 0012R $7.95.**

BUSINESS PACKAGE I Keep the books for a small business with your TRS-80 Level I 4K. The six programs included are:
•**General Information**—The instructions for using the package.
•**Fixed Asset Control**—This will give you a list of your fixed assets and term depreciation.
•**Detail Input**—This program lets you create and record your general ledger on tape for fast access.
•**Month and Year to Date Merge**—This program will take your monthly ledger data and give you a year to date ledger.
•**Profit and Loss**—With this program you can quickly get trial balance and profit-and-loss statements.
•**Year-End Balance**—This program will combine all your data from the profit-and-loss statements into a year-end balance sheet.
With this package, you can make your TRS-80 a working partner. **Order No. 0013R $29.95.**

DEMO I This package is just the thing to show your friends what your TRS-80 can do. Included are:
•**Computer Composer**—Compose and play music using only a standard AM radio.
•**Baseball**—Play baseball with your computer while it does the scorekeeping.
•**Horse Race**—Place your bet and cheer your pony to the winner's circle.
•**ESP**—Test your powers of extrasensory perception.
•**Hi-Lo/Tic-tac-toe**—Guess the secret number or get three in a row.
•**Petals Around the Rose**—Can you figure out the secret behind the five dice?
•**Slot Machine**—Turn your computer into a one-armed bandit. These programs require a TRS-80 Level I 4K. **Order No. 0020R $7.95.**

DESTROY ALL SUBS/GUNBOATS/BOMBER This package of three programs is fun for the whole family. Included are:
•**Destroy All Subs**—Hunt down enemy subs while avoiding mines and torpedoes. A one-player game.
•**Gunboats**—Try to blow the enemy's ship out of the water. For one or two players.
•**Bomber**—Carefully release your bomb to destroy the moving submarine. A one-player game.
To enjoy these programs, you'll need a TRS-80 Level I 4K. **Order No. 0021R $7.95.**

PERSONAL FINANCE I Let your TRS-80 handle all the tedious details the next time you figure your finances:
•**Personal Finance I**—With this program you can control your incoming and outgoing expenses.
•**Checkbook**—Your TRS-80 can balance your checkbook and keep a detailed list of expenses for tax time.
This handy financial control for the home requires only a TRS-80 Level I 4K. **Order No. 0027R $7.95.**

*A trademark of Tandy Corporation

SEE INSIDE BACK COVER FOR LIST OF DEALERS NEAR YOU.

Make a great buy even better . . .

Why pay full price? Take this coupon to the store nearest you and get $1.00 off on any of our software. If the store nearest you doesn't stock Instant Software, then use the order blank at right to order your software direct, or call Toll-Free 1-800-258-5473.

INSTANT SOFTWARE

$1 This coupon good for one dollar off any Instant Software package when purchased through your local computer store. **$1**

49MOI

Buyer's name ____________

Address ____________

City ________ State ________ Zip ________

Signature ________ PACKAGE NUMBER ________

ATTENTION DEALER: Include coupons with your next order. Credit will be applied toward purchase. If you are not as yet selling Instant Software, write: **Instant Software, Dealer Sales, Peterborough NH 03458** for catalog and dealer net price schedule, or call our marketing department at 1-603-924-7296 for more information.

Limit one coupon per package. Offer void where prohibited by law.

Expires 1-31-80

Name ____________

Address ____________

City ____________

State ________ Zip ________

☐ Check ☐ Money order

☐ VISA ☐ AMEX ☐ Master Charge

Card No. ____________

Expiration Date ____________

Signed ________ Date ________

Order your Instant Software today!

Quantity	Order No.	Unit Cost	Total Cost
		Handling	$1.00
		Total Order	

Instant Software Inc. Dept. 49MOI
Peterborough, N.H. 03458 USA

Instant Software™ Inc.

✓2

80 NEWS
edited by Michael Comendul

Computer Store Openings

Computer Science Systems, San Antonio, TX has opened its first TRS-80 software store in San Antonio.

The store, the first of four to be opened over the coming months features accounting programs for the business community, as well as, games and advice for the computer enthusiast.

Stores are slated for Dallas, Houston and Hidalgo, Texas.

In addition to software the stores will carry Mini-Disk Drives, Diskettes and Printers for Tandy Company's TRS-80 computer. Support items offered will include, paper, ribbons, memory expansion kits and modification kits.

10 Million Character Storage for TRS-80

LaSalle Computing, Inc., Norristown, Pa., has introduced a mass storage medium for the Radio Shack TRS-80 system, providing the user with capabilities of a mini-computer system.

LaSalle has developed the Series/3 Disc Operating System software linking the Pertec 5 to 20 million character hard disc and Cameo model DC500 Disc Controller with the TRS-80. The system requires a user-supplied 32K model 1 TRS-80 with expansion interface and a single Radio Shack disc drive.

The Pertec 10 million character disc drive, Cameo controller and all system cables is priced at $4,795 F.O.B., Anaheim, California and may be ordered through LaSalle Computing, Inc. or Cameo Data Systems, Inc. Availability is stock to 60 days. The Disc Operating System software is priced at $600.

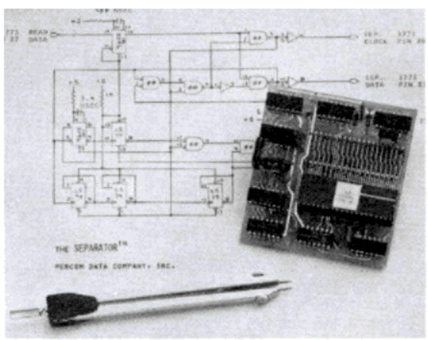
For further information contact Dick Harding, LaSalle Computing, Inc., P.O. Box 116, Blue Bell, PA. 19422, Larry Covey, V.P. Marketing, Cameo Data Systems, Inc., 1928 S. Anaheim Blvd., Anaheim, California 92805.

TRS-IBM Interface

The TRS-80 may now be interfaced to the IBM Model 50 Electronic Typewriter. A hardware interface for connecting the Model 50 to the CPU or the Expansion Interface and software including a printer drive program, patches for the Electric Pencil, as well as other special application programs (including a program for typing TV/Movie scripts), are now available from Mediamix, Universal City, CA.

The IBM Model 50 when combined with the Mediamix products can type in 10 and 12 pitch regular type, or proportionally spaced type, all with right justification.

Sentence and word underlining, title centering, indented paragraphs and an array of special codes are available.



The Percom Separator.

Data Separation Reliable

A plug-in adapter for the TRS-80 and MP-F mini-disk controllers which eliminates the data read errors caused when clock and data bits are not reliably separated during playback, has been introduced by Percom Data Co., Garland, TX.

The problem relates to the higher storage density of the inner disk tracks, and is not uncommon with either controller.

Called the Separator, Percom's adapter may be installed without making any changes to the host system.

The Separator uses an external data separation circuit that compensates for a phenomenon referred to as "bit shifting."

According to Percom, bit shifting arises during playback (reading) of high density data and this is the source of most read errors.

An assembled and tested Separator adapter sells for $29.95, including installation instructions.

Interface Allows Level II

The new CH14 interface eliminates the need for a RS-232 interface and requires no software. It is designed to interface the Heathkit H14 printer to the Centronics parallel port on the TRS-80 expansion interface. The user plugs one cable into the printer port on the expansion interface and the remaining cable has a dip plug that connects inside the H14 printer. The installation takes 5 minutes and allows Level II commands, LPRINT and LLIST to be used. The CH14 is available for $99.95.

Cost Effective Computer Services, 728 S. 10th St. Suite #2, Grand Junction, CO 81501.

DOT Again

Disco-Tech, Santa Rosa, CA has introduced DDT, a new "disc drive timing" program for both TRS-80 and Apple II microcomputers.

DDT lets every disc drive owner keep track of his disc drive motor speed and adjust it, averting possible data loss and incompatibility among diskettes.

Disco Tech's DDT program works on any drive, and provides a real-time graphic display of motor speed which allows adjustment within one-tenth of one RPM, out of an optimum 300 RPM.

DDT comes with documentation, a 32-page manual that takes the user step-by-step through the analysis and adjustment procedure.

Disco-Tech is marketing its DDT program in two versions, for the TRS-80 and for Apple II. Available in retail stores the TRS-80 DDT program may be purchased on cassette at $14.95 or on diskette at $19.95. The Apple version of DDT is available only on diskette at $19.95. DDT is also available direct from Disco-Tech.

Inventory Control for TRS-80

An inventory control system for the TRS-80 computer has been released by National Software Marketing Inc., Hollywood, Florida. The package operates on a 32K Level II system with 2 to 4 floppy disks and a printer.

The system stores and instantly retrieves up to 3000 items (1000 per disk drive), which can contain a six digit item number and a 24 character description. The master file contains quantity on hand, quantity on order, reserved stock, safety stock level, average cost, standard price, month and year-to-date sales in units and dollars, month and year-to-date cost of sales and average weekly usage. A complete audit trail of purchases and sales is maintained.

The system is menu driven allowing selection of entry, editing, updating and file maintenance. The reports produced include Inventory Status, Price List and Reorder Analysis and Sales Analysis. Price is $89.95.

NSM, P.O. Box 6195, Hollywood, FL 33021.

Tape Digitizer from Alphanetics.

New Tape Digitizer

A device for efficient data storage and retrieval is now available to TRS-80 users. A new tape digitizer eliminates bad loadings and permits copying of data and program tapes without using the TRS-80.

The tape digitizer is fully compatible with Level I and Level II formats and tapes reproduced on this device can be played back on any ordinary cassette.

Two features of the digitizer are its ability to make copies of tapes and "system" tapes that are better than the originals.

With most tapes, low volume levels to full volume will produce reliable loading when using the tape digitizer and circuits filter out hum and noise while compensation for large volume variations enables recovery from minor tape drop-outs.

To load programs or data, the digitizer is connected as a buffer between the cassette recorder and the "Tape Earphone" plug.

Accompanied by an instruction sheet and necessary cables, the digitizer retails at $44.95 postpaid.

Write Alphanetics, P.O. Box 597, Forestville, CA 95436.

Avoid Microwelding

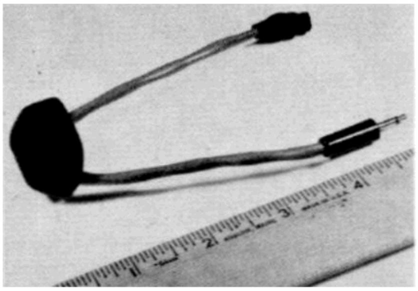
Web Associates, Monrovia, CA, is attacking the all-too-common TRS-80 cassette drive hang-up, a special problem according to Web research, if the user is running a lot of data saves or loads, during which the cassette recorder is turned on and off several times a minute.

The actual failure, said Web, is caused by a phenomenon known as microwelding, which occurs as a result of excessive current and heat build-up in the TRS-80 cassette control reed relay.

The microwelding is further compounded by a slight, self-holding, electomagnetic force induced by the high recorder current. This added electro-magnetic force is why, in most cases, the hang-up goes away when the cassette recorder is manually turned off. Web Associates has developed the TBUFF module, no larger than an ice cube, that simply plugs in line with the REMOTE cable between the TRS-80 and the recorder. TBUFF reduces the current passed through the reed relay in the TRS-80 while delivering full power to the recorder.

TBUFF is available in two models—N and R—and requests the CTR model number or REM jack polarity accompany each TBUFF order.

TBUFF retails for $9.95 from Web Associates, P.O. Box 60 NA, Monrovia, CA 91016.



The Web REMOTE Cable.

User's Group Sponsors Fair

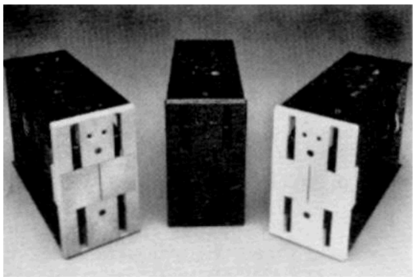
Preliminary plans are underway for the first West Coast Computer Fair sponsored by the TRS-80 Users Group of Sarasota.

The fair will include all areas of interest and all makes of computers. Tentatively scheduled for February or March, 1980, we welcome responses from individuals, businesses, or educational institutions.

If you are interested in reserving a commercial booth, giving a lecture or demonstration, or would like additional information, please contact Carmen A. Gianforte, Program Chairman, 19 North Boulevard of Presidents, Sarasota, Florida 33577.

Compact Dual Drives

A new 5¼" dual floppy disk drive that packs two reliable drives with speed and increased storage capacity into a standard 3¼" x 5¾" x 8½" package, has been designed by Energy Equipment, Inc.



Dual floppies from Energy Equipment

"Another DAM Floppy" offers 400K bytes of usable information (10 sectors of 512 bytes, dual density format, 40 tracks) that can be stored on two standard 5¼" diskettes.

The dual floppy accesses random data at 12 ms. per track, seek time with a head load time of only 15 ms. and a settling time of just 12 ms. In combination, it gives you an average random access to 400K bytes of information in just 270 ms.

A ceramic, straddle erase read-write head decreases time tolerance for sectoring, while a precision ground "V" groove ensures correct head positioning on any standard 5¼" diskette. Contact Energy Equipment, Inc. for more information.

The Datestones of Ryn

The Datestones of Ryn is an exciting new solo fantasy adventure from Automated Simulations.

A dark and deadly labyrinth of caves and tunnels hides the date stones stolen from the ducal calendar of Ryn (pronounced rune).

The Datestones of Ryn has fourteen options with a map of the dungeon.

The Datestones of Ryn comes complete with a colorful 16-page illustrated manual, program cassette and command summary card.

The Datestones of Ryn requires Level II, 16K and comes on cassette or disk for $14.95.

Automated Simulations, P.O. Box 4232, Mountain View, CA 94040.

Disk Drive Cleaner

A Disk Drive Head Cleaner for the TRS-80, is available in the form of a mini diskette with reusable head cleaner on both sides. In addition, a program is included that does the cleaning thoroughly and automatically. A cleaning solution is also provided.

The Disk Head Cleaner will allow more reliable disk drive operation and save the user the cost of head cleaning maintenance, according to TBS, Inc. Atlantic, GA.

TBS is in the process of developing the cleaner, which sells for $12.95, for other computers.

For more information see your local TBS

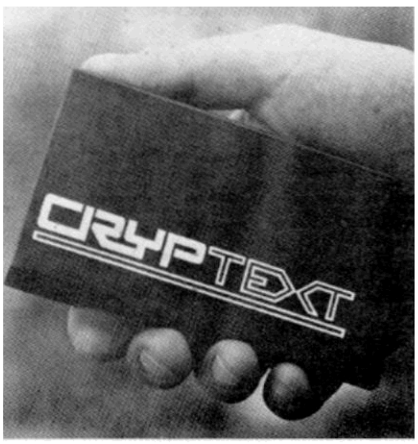
dealer or write The Bottom Shelf, Inc., P.O. Box 49104, Atlanta, Georgia 30359.

Plug in CRYPTEXT

CRYPTEXT, a hardware encryption device, plugs into the back of the TRS-80 or into the expansion interface via an optional cable.

The pocket-size device allows you to secure inventory and financial data, technical and proprietary information, graphics, programs or text. The encrypted information can be stored on cassettes or disks.

Used with a modem, CRYPTEXT transmits data or messages by telephone or other



CRYPTEXT Device.

communication channels. Other uses include generating pseudorandom numbers for games or scientific programs.

Data throughput is greater than 15K bytes/second and power consumption is low at less than 100 milliwatts.

The CRYPTEXT is available for under $300 each.

Write: CRYPTEXT Corporation, P.O. Box 425, Northgate Station, Seattle, WA 98125.

Diagnostic Software

Diagnostic software for the TRS-80 computer system is now available from VR Data, Folcroft, PA.

The new software is written in separate programs to test RAM, ROM, CPU, and I/O circuitry for errors. All the programs detect malfunctions in hardware circuitry.

The diagnostic software is available for $34.95 each on cassette or diskette from VR Data, 777 Henderson Boulevard, Folcroft Industrial Park, Folcroft, PA 19032.

INFOBOX

InfoBox is a new information system for the TRS-80. The user can store, look up, delete, save and read files from cassette or disk with five one-letter commands.

The basic unit in InfoBox is the item. An item is limited only by memory size. InfoBox loads and runs. There is no initialization. Because it is an all-in-one program, it can perform all functions with no switching back and forth among cassettes or programs.

Versions of InfoBox are available for the TRS-80 Level-II and TRSDOS. Both versions use about 3K. While Infobox does not require disk or printer, both versions have commands to print selected items on a printer. The price is $19.95.

Micronybble Systems, 63 Dana Street, Cambridge, MA 02138.

Dynacomp Programs

Four new programs, on cassettes, are being offered for TRS-80 (Level II) users by Dynacomp, Webster, NY. These include Flight Simulator, a program which allows the user to take-off, fly, navigate and land an airplane; Valdez, a simulation of the navigation of a supertanker through ship and iceberg traffic with the goal of reaching the oil port of Valdez, Alaska; Bridge 2.0, a computer version of the card game, which both bids and plays; and Hearts 1.5, a card game that pits the player against two computer opponents.

The software requires 16 Kilobytes of program memory and ranges in price from $9.95 to $17.95.

Dynacomp, P.O. Box 162, Webster, NY 14580.

200 KBYTE 77-Track Mini Disk Drive

Microcomputer Technology, Inc (MTI), Santa Ana, CA has introduced a large capacity mini drive for the Radio Shack TRS-80 computer.

The new mini disk system, identified as the Model TF-7, features 77 tracks and has the largest storage capacity of any mini drive now available for the TRS-80: 195 Kbytes of online storage. To realize the full potential of the 195 Kbyte capacity, a new, 77 track version of the MTI/APPARAT DOS+ disk operating system is also being made available by MTI.

The new model TF-7, 77-track unit is priced at $625 and is available for off the shelf delivery.

Micro-Backgammon 1.5

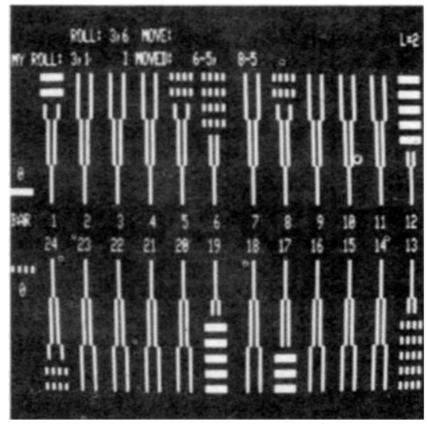
Micro-backgammon 1.5, for the Radio Shack TRS-80, is a backgammon playing program released by Questar Software. It includes three levels of play ranging from beginner to expert.

The computer moves its pieces using both animated graphics and a literal display.

The Questar game includes a number of features: you can set up or alter any position; dice rolls are random or input for tournament play; and you can switch sides with the computer, watch it play against itself, or set up and have it play any position or dice roll. All moves are checked for legality.

Micro-backgammon 1.5 is written in Z-80 machine language. Two versions are included on one cassette so that it can be run on both Level I and Level II 4K TRS-80's. A complete instruction booklet comes with the cassette. The price is $19.95.

Questar Software, P.O. Box 723, Wichita, KS 67201.



Micro-Backgammon by Questar.

FLIST Allows Pagination

A new software program from Faulk & Associates allows pagination on hard copy listings; date and time printing on listings; user data inserted in the heading on the listing; support for various printers including user defined line length and page length.

FLIST is loaded as a temporary extension to either Level II BASIC, or Disk BASIC, and supports the new DOS 2.2 extensions to BASIC (including BASICR and recovery of BASIC programs already in memory).

When running under Disk BASIC, FLIST adds a new BASIC command to DOS CALLED BASICF. This loads the FLIST program and adds the temporary interface to BASIC providing the new command FLIST to be useable from BASIC. BASIC itself is not modified and this product will not affect the performance of any BASIC programs.

FLIST resides in high memory for either a 32K or 48K machine under DOS or Level II.

The BASICF command is available only on version 2.2.

The price is $19.95.

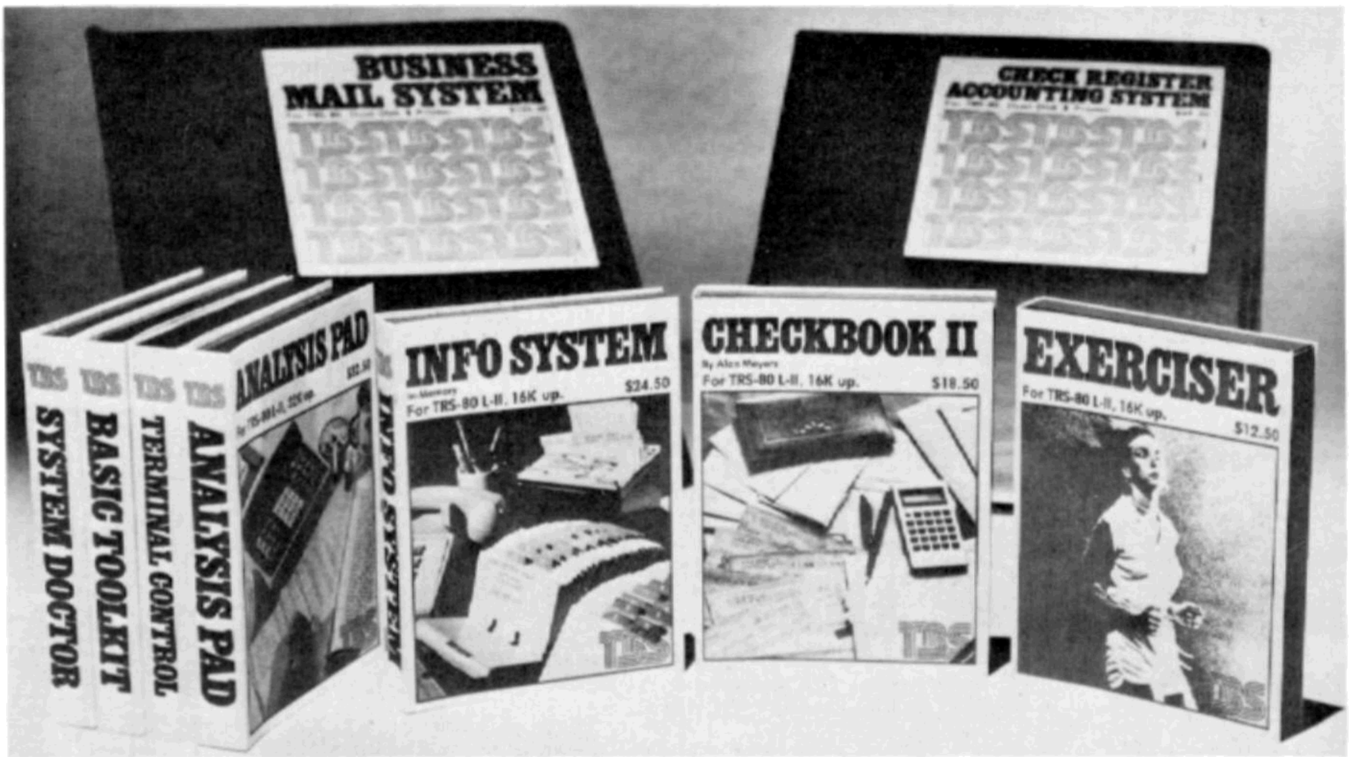
Write Faulk & Associates Software, 2531 E. Commonwealth Ave., Fullerton CA 92631.

Cybermate Package

Cybermate, Nazareth, PA, has available a full line of programs, games, puzzles and graphics for the TRS-80.

The programs include language and grammar; games of Monopoly, Scrabble or poker;

THE BEST.



If you're not content with just playing games, **TBS** is producing applications software for your TRS-80 Level II that makes it a practical tool.

CHECKBOOK II by Alan Meyers is the finest program of its kind yet published. With superb graphic screen displays, it does everything necessary to keep your checkbook balanced. Data is input directly into a five-column screen display with a field for alpha or numeric codes. Editing is done easily in any or all columns. **CHECKBOOK II** will accurately balance and reconcile your checkbook, handling balances up to $1,000,000. Your balance brought forward is always in memory. Outstanding checks are listed and easily saved. You can also search for an entry by any field except amount, and all checks with matching entries will be displayed and totaled. A numeric sort routine is included. Screen prints can be made to a line printer from almost any point in the program. In addition, the 32-48K version can write files to disk. This and the 16K version are included on the same tape. For $18.50, **CHECKBOOK II** is the top of the line in personal checkbook programs.

INFORMATION SYSTEM by Dale Kubler is simply the best in-memory, data base manager on the market. It allows you to create files with up to ten categories per 'page', up to 40 characters per category and 200 characters total per page. Data from the keyboard is entered directly on a screen display of one entire page. Once entered, you can sort or search your entire data base by any category and have the information desired displayed on the screen. **INFORMATION SYSTEM** provides a thorough editing mode allowing changes by line without rewriting an entire file. Program your own printouts to almost any form you desire for line or serial printers. Screen prints from anywhere in the program are also available. **INFORMATION SYSTEM** creates either disk or cassette files depending on the version you use. Four versions are supplied with the program tape. From mail lists to recipes, for only $24.50, this program is the ideal information manager.

EXERCISER is for everyone. This program allows you to set your own physical fitness goals, then chart and analyze your progress toward these goals. Further, you may program an exercise regimen, then have the computer 'coach' you through your exercise routines. This system will allow you to use your computer to reinforce your effort to attain physical health. **EXERCISER** is really two programs in one. One measures your progress in jogging, swimming and bicycling and the other is for setting calisthenic regimens. It has long been known that to effectively structure an exercise program, it is necessary to think in terms of goals which can be met over a period of time. Whether you are training for the Boston Marathon or just wish for a minimum level of fitness, **EXERCISER** is designed to help you attain your goals. The price for this exceptional program is just $12.50.

TBS has other great software for your TRS-80. **BASIC TOOLKIT, SYSTEM DOCTOR & TERMINAL CONTROL** are systems utilities. **BUSINESS MAIL LIST, DATA BASE MANAGER, CHECK REGISTER ACCOUNTING SYSTEM & ANALYSIS PAD** are strong applications for business. Don't forget the **LIBRARY 100**; 100 programs for only $49.50. **TBS** also has **DISK HEAD CLEANERS** for **TRS-80** and **APPLE** and **GRAN MASTER DISKETTES**, the best on the market.

TBS is **YOUR COMPANY**, and to you we pledge to produce quality software at a price you can afford. The above products are available **NOW** at Computer Stores and Associate Radio Shack Stores nationwide or directly through us. For more information please contact us at the numbers below.

TBS™ ✓6

THE BOTTOM SHELF, INC.
(404) 939-6031 • P.O. Box 49104-M • Atlanta, GA. 30359

word processing, math tests and ecology study programs.

A source package of 41 programs sells for $19.95. Also available are cassettes for $4.95.

Write Cybermate for details: RFD #3, Box 192A, Nazareth, PA 18064.

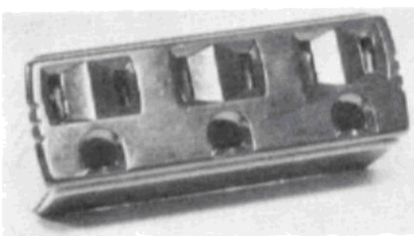
Payroll System for Business

The 32K Comprehensive Payroll Program, for up to 115 employees, is now available from the V R Data Corp.

The random access program determines federal withholding, FICA, state and local taxes. It also deducts sick pay from FICA, produces monthly, quarterly and year to date reports and can be custom tailored for individual payroll periods—weekly, bi-weekly, semi-monthly and monthly.

The system is programmed to print checks and stubs to guarantee accurate records. The price is $99.95.

V R Data Corp., 777 Henderson Blvd., Folcroft Industrial Park, Folcroft, PA 19032.



The Lightning Buster from TBS Inc.

Electronic Safety

TBS Inc. announced the release of a new product called the Lightning Buster. This device protects your computer from power surges up to 1000 amps. The Lightning Buster is a three plug adapter that goes into any three prong 125V, 15A wall outlet. The Lightning Buster is an inexpensive means of assuring the safety of electronic investments.

The price is $14.50. For more information see your local TBS dealer or write The Bottom Shelf, Inc., P.O. Box 49104, Atlanta, GA 30359.

Tiny PASCAL

Supersoft, Champaign, IL has released the only authorized version of the famous Chung/Yuen Tiny PASCAL for the TRS-80. The Tiny PASCAL package contains the editor, compiler, and the run time support routines, as well as a complete operating system.

Tiny PASCAL on cassette requires a Level II 16K computer, but no disc drive. The entire system is in 8080 machine code, but source to the compiler is also supplied with the tape. The compiler source is written in Tiny PASCAL.

Tiny includes the complete subset of standard PASCAL: recursive procedure/functions (7 levels deep); WHILE CASE; FOR (LOOP); REPEAT/UNTIL; READ; WRITE; IF THEN ELSE; 'PEEK' and 'POKE'; and complete graphics for the TRS-80.

Tiny PASCAL is available from Supersoft, PO Box 1628, Champaign, IL 61820, for $40.

Mailing Software

A new software product called the Business Mailing System has just been released by TBS Inc. It is designed for large scale business users with a TRS-80, at least 32K of memory, a printer and two disk drives. The Business Mail System allows the user to store up to 150,000 names on a single file composed of multiple diskettes. The Business Mail System sorts the entries into zip code order and alphabetical order within the zip code. As new entries are made, the file is expanded automatically by the computer.

The Business Mail System allows you to use one through four labels at your discretion. It provides for the printing of either three or four line addresses. The mail list also allows you to program names you wish to print out by using ten exclusive and non-exclusive codes.

The Business Mail System cost $125. For more information write The Bottom Shelf, Inc., PO Box 49104, Atlanta, GA 30359.

Disk Based Box-Jenkins

Box-Jenkins, a technique used for sales, price, interest rate and production forecasting, has been implemented for TRS-80's with disk capabilities and at least 32K of memory.

The current model, according to Applied Economic Analysis, gives the user the option of letting the computer select the best set of parameters to be used for forecasting. In this way even those who are unfamiliar with the Box-Jenkins procedure are able to use the technique.

The $97 package comes in a three ring binder with diskette and 40 pages of explanation and documentation.

Applied Economic Analysis, 4005 Locust Ave., Long Beach, CA 90807.

Data Base Manager

Information System (In Memory) is a new 16K program for the TRS-80.

According to TBS, Inc., the program functions as an in-mem data base manager and is operator programmable. Up to ten fields, with 40 characters per field and 200 total characters are allowed. The number of records held in memory depends on the number of fields you create, the field lengths, and the amount of memory your computer has.

Data base can be stored by any field with a high speed machine language sort. IN MEM creates either disk or tape files. A complete editing mode is provided allowing changes by line instead of entire record.

Information System allows you to program your printouts to any format you desire. The program is ideal for small mailing lists, inventories or anything that you would normally file with index cards. The price is $24.50.

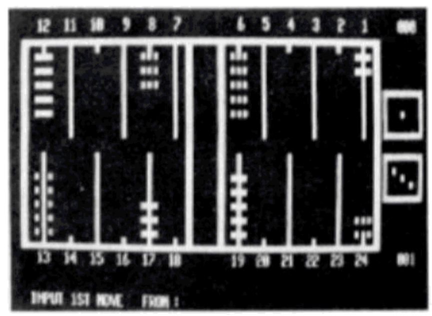
The Bottom Shelf, Inc., P.O. Box 49104, Atlanta, GA 30359.

New Mini Disk

The MF-80 mini-floppy disk drive is compatible with all Radio Shack TRS-80 software and hardware. It consists of the MPI B51 drive and a heavy duty power supply enclosed in a silver case.

The B-51 drive features 40 tracks, dust tight door and diskette eject. It has a clutch which prevents crimping of the center hole of the diskette. The MF-80 is priced at $359.

Cost Effective Computer Services, 728 S. 10th St. Suite #2, Grand Junction, CO 81501.



Classic Backgammon

Backgammon for the TRS-80

The Classic Game Series, introduced by The Software Association, includes three programs for the 16K Level II TRS-80.

BACK-40, a backgammon program, includes computer or player opening depending on dice roll, computer and player doubling and scoring of all regular, gammon, and backgammon endings. Points are numbered to make move inputs simple and all moves are checked for legality.

BACK-40 is priced at $14.95 with complete instructions. The other games in the series, Z-CHESS and DR. CHIPS (an "ELIZA" type program) are $17.95 and $14.95 respectively.

The Software Association, P.O. Box 58365, Dept. KM, Houston, TX 77058.

Does a product interest you? Mention 80 Microcomputing when you query!

✓ Reader Service—see page 147

From a two man leather goods enterprise to a multi-million dollar business, the development of Tandy Corp.

The Tandy Story

Dave Tandy, founder of Tandy Co.

Lewis Kornfeld, President of Radio Shack Corp., a division of Tandy Corp.

By Chris Brown

The roots of Tandy Corp. lie deep in the dusty plains of western Texas, where, over sixty years ago, two friends pooled their resources and laid the foundations for this Fort Worth success story.

"By the book; always keep your emphasis on net profitability." That is the philosophy old Dave Tandy espoused and it persists to this day in the boardrooms and executive offices of Ft. Worth. The result is a company which controls nearly all phases of its operation, from manufacturing to retailing.

In the consumer electronics field their operation is unique and may possibly make Tandy/Radio Shack the healthiest domestic electronics firm facing the rugged economic sailing of the 1980s.

Selling Leather

In 1918 Dave Tandy and Norton Hinckley bought a modest inventory of shoe leather and other repair supplies, and sold them to cobblers throughout Texas under the name of Hinckley-Tandy Leather.

Business was good and by 1927 they opened their first branch store in a small town, Beaumont, TX. Five years of lean operation taught Tandy a valuable marketing lesson and he relocated to more progressive Houston.

Dave Tandy remained a salesman all of his life and he passed on this enthusiasm for business to his son, Charles, who built the Tandy Corporation of today, serving as its first president and chairman.

World War II had interrupted Charles Tandy's Harvard Business School studies but his tour of duty made him aware of the interest in leather hobbies that existed within the armed forces.

After the war he returned to Hinckley-Tandy, but he wasted no time in capitalizing on what he perceived as a lucrative craft market. In 1950 he opened two stores specializing exclusively in leather crafts. To complement the retail operation, he also began a mail order sales business.

Tandy's long range goal was a chain of stores serving the hobbyist/craftsman and his instincts proved correct. His retail and mail order operations returned 100% percent on his investment in the first year. This combination of retail outlets backed up by mail order sales was the key element in Tandy Corp.'s future success.

First Catalog

The first Tandy catalog, only eight pages, was sent to respondents to a two-inch ad that appeared in the April, 1950 issue of *Popular Science* magazine. The catalog offerings, aimed at the do-it-yourselfers, brought an encouraging response. Dave and son, Charles, had struck a chord in the American mentality. The era of the hobbyist was at hand.

The Tandys' ideas for the future of the company increasingly diverged from those of Norton Hinckley, their old partner. The inevitable split came in 1950.

It was amicably agreed that the Tandys would pursue the leather craft market, while Hinckley would continue in the shoe findings trade.

In the vangard of the emerging do-it-yourself movement that swept America in the wake of consumer shortages and post-war economic slow downs, Tandy's sales climbed to over $2.9 million within two years. They opened fifteen stores during those first two years and began to develop a cadre of management talent.

In 1952, Tandy made its first of many acquisitions, a failing handicrafts manufacturing firm in New Jersey. The acquisition gave Tandy access to the huge East Coast market and gave them an in-house manufacturing facility.

Tandy had opened 16 more retail stores, but in each case the philosophy was the same: Back up retail sales with direct mail advertising and place new retail outlets in those areas where large mail order markets already existed.

In the five years between 1950 and 1955, Tandy's sales rose to $8 million but to protect the company's ownership from estate prob-

Line art courtesy of Newcomen publication: Tandy Corporation: "Start on a Shoe String", *1968*

Radio Shack TRS-80 Model II with Line Printer III and an external disk system.

lems, Tandy allowed itself to be acquired by a Boston tannery company that had come upon hard times. Terms of the sale provided options for Tandy stockholders to buy shares of the parent company over a four year period.

From 1955-1958 Tandy remained a division of the newly established General American Industries, but further unprofitable acquisition by the new parent destroyed their relationship.

It became obvious that only two of six General American divisions were making money. A struggle for control ensued with Charles Tandy emerging victorious after persuading a key European stockholder to withhold proxy votes. By 1960 Charles Tandy had eliminated the last of the unprofitable divisions in what is referred to by the Tandy management as the "clean up year."

Tandy Meets Radio Shack

Changing its name to Tandy Corporation and its address—to Ft. Worth—in 1961 the company operated 125 stores in 105 American cities. Though many were leased premises that shared space with the local drug store, five and dime or post office, a foothold was established in major American marketing areas.

1961 was also a year of acquisition for Tandy. With the lessons of the rapid expansion of 1956 still fresh in mind, Tandy cautiously went shopping in the corporate marketplace. Five companies in varying states of health were acquired, each related to the do-it-yourself industry.

In 1962 Radio Shack was a small, struggling electronics firm in Boston, MA. Nine retail outlets in the Boston area and a small mail order operation generated $9 million a year in sales. Most of its customers were old time ham radio types whose passion for electronic circuitry could only be sated by a ready supply of components.

Realizing Radio Shack's compatibility with his retail/mail order operation, Charles Tandy did not hesitate to take control.

Tandy controlled Radio Shack's management by April 1963; owned 66% of its common stock by September 1. By June 1965 Charles Tandy owned 85 percent of Radio Shack Boston. Applying the Tandy philosophy to the Radio Shack operation, he soon reduced inventory from 4000 items to 2500. He replaced slow moving items with high volume, fast turn-over ones.

When domestic vendors were unwilling or unable to respond to his needs he went elsewhere—often to the Far East. Realistic and Archer became company trademarks and developed a reputation as inexpensive and functional alternatives to more exotic brand names.

Two years after acquiring Radio Shack, Charles Tandy had turned a $4 million loss into a profit. With sales approaching $20 million, 40 percent of the Tandy total, Radio Shack was becoming the most lucrative asset in the Tandy stable of companies.

Charles Tandy's visionary perception of the electronic hobby market was confirmed. In 1969, 5000 employees worked for Tandy and sales approached $100 million.

Throughout the 70's Tandy grew steadily, consumer electronics becoming a greater market than even Charles Tandy had estimated. Tandy became a household word in electronic America.

Tandy Today

In 1976 all non-electronic activities of Tandy were spun off to Tandy stockholders as separate corporations. This gave Tandy ready capital which could then be invested in other areas, the TRS-80, for example.

Despite suffering significant losses in the personal communications market (CB) in recent years, Radio Shack has consistently shown a 45.8 percent pretax return on investment. (IBM, an acknowledged model of the well-managed American corporation, rarely exceeds a 39 percent figure.)

Net income per share rose 30 percent in fiscal 1978 for Radio Shack with sales gains following that pace.

Radio Shack now operates four manufacturing plants in the Far East while opening new retail outlets at the rate of two per day for the last five years. Radio Shack's 7000 outlets (5200 domestic) posted an 18 percent sales gain nationally in fiscal 1978.

In what direction does Radio Shack seem to be heading? Most indicators point to an increasing commitment to computers. A stock repurchasing plan is currently underway and this usually indicates that a trade-off is in the works; in this case, stock might be offered for increased manufacturing capability.

Radio Shack's ability to meet the demand for the TRS-80 system has fallen woefully short of their own projected goals.

Further, an acknowledged world-wide shortage of the IC chips used in computer systems may have been aggravated by the tactics of some of the larger computer corporations. Allegations of over-buying on the part of IBM and other large system manufacturers have surfaced.

Radio Shack is acutely aware of possible competition with these industry giants and their rumored intrusion into the future minicomputer market. However, the unique structure of Radio Shack which markets and sells its products through its own retail outlets gives it an inherent advantage over some of the larger computer manufacturers.

Radio Shack marketing priorities for the foreseeable future are the small businessman, the educator and the recreational or hobbyist user.

This attitude does not bode well for most home hobbyists. In the words of Tandy Corporation vice-president, Lewis Kornfield, president of the Radio Shack division, "Our impressions to date would indicate the hobbyist, while vocal and visible, is not the mainstream of the business."

Post sales customer support has been aimed at the business and educational user. A network of 55 regional repair locations has been established to meet the needs of the non-technical user.

In addition, WATS line service is available to both user and Radio Shack store personnel. Calls have been coming in at a brisk 20,000 per month; indicating the dependency of the non-technical user on the manufacturer.

In view of Radio Shack's current marketing priorities it seems doubtful they will make inroads into design technology but will instead stay with a proven system while constantly chipping away at the cost.

Radio Shack's insistent refusal to carry other lines of software has been nullified by the wide-spread popularity of the TRS-80 system. A huge satellite industry has sprung up around the TRS-80 system, dedicating itself to fulfilling the needs of the more gung-ho TRS-80 users. The variety and complexity of non-company produced peripherals, available to the TRS-80 owner is unmatched in any other system currently in use in America. The TRS-80 user enjoys more hardware and software options than any other microcomputer owner today.

Conclusion

Radio Shack is a company facing the 1980's with a corporate perspective on American business nearly seventy years old. It is a company in control of most facets of its operation from material acquisition to retail sales.

But there are ominous rumblings in the distance. The giants of the computer industry are stirring out of their lethargic sleep as they sense the money to be made in the new microcomputer market. Will IBM, Digital, or Wang Laboratories be able to sell a $2000 system? Can they compete with Radio Shack's proven ability to market and sell on the retail level? Can they overcome the advantage Radio Shack has by simply being first?

The stage is set for the home computer epic. How the scenario will develop is the stuff of much speculation at computer club meetings and in corporate boardrooms. Whatever happens, Radio Shack will be in the thick of it. They have started a revolution in information management for the little guy, and have too much at stake to back out now.

Early cover from a Tandy Company catalog, appealing to do-it-yourselfers.

In the season of good cheer let your 80 do the greeting!

Merry TRSMAS

Colin B. Taylor
Route 1, Box 122
Killington VT 05751

All of us are familiar with conventional holiday greeting cards, but what computer hobbyist can resist an electronic version showing off his favorite computer? Imagine the effect of a new computer displaying a holiday message, from beneath the tree, to wide-eyed children on Christmas morning!

What kind of program should be written and what should it do? A simple message could be written and displayed on the video screen, nice, but not too interesting. A timer could be added to flash the message on and off and create some action. Graphics can be effective but somehow a stationary display tends to get boring.

A Moving Greeting

This program, written for the Radio Shack TRS-80 with Level II BASIC and 4K RAM, contains all the above. The graphics create a Christmas tree, with blinking lights, located near a window, through which a snowstorm is visible. The scene includes a title and a greeting.

I had to overcome a number of practical problems before the program worked. Locating and manipulating the selected graphic blocks was one and making the snow fall and the tree lights blink simultaneously, but at different rates was another.

Let's go through the program step-by-step.

Lines 20 through 70 clear the screen, print the banner and the title. The subroutine in lines 1100 through 1130 provides a full line of "%" symbols for the top and bottom borders of the title. (This could have been implemented as the STRING$(64, "%") function, if sufficient string space was allocated with a CLEAR 64 statement). Lines 40 and 50 provide the borders for each end of the printed title. Line 70 contains the title.

Line 90 arbitrarily defines all numeric variables as integers.

Lines 100 through 130 draw the sides of the tree. Starting at the top point, the I loop sets the horizontal and vertical coordinates of each point making up the angled sides. The A variable determines the horizontal coordinate of the left side and the B variable does the same for the right side. The Y variable determines the vertical coordinate of the specific graphic blocks to be illuminated.

Lines 140 through 160 draw a horizontal line at the bottom of the tree. The vertical stem extending from the bottom of the tree is drawn by lines 170 through 210 and lines 220 through 263 draw the two horizontal lines of different lengths that are the base of the tree.

Lines 265 through 290 draw the upper diagonal tree trim. The subroutine at lines 1000 through 1030 illuminates five graphic blocks in a horizontal line while the loop between lines 270 and 290 sets the location and number of times the subroutine is called. Similarly, the lower diagonal tree trim is drawn by lines 295 through 320.

Lines 330 through 350 print the greeting. These lines may be modified to allow the program to be personalized.

Line 360 continues the program to line 2000.

Lines 2000 through 2030 draw the horizontal parts of the window frame and lines 2040 through 2090 draw the vertical. The added steps in the latter section widen the vertical lines to match the proportions of the horizontal lines.

Lines 2100 through 2130 assign values to the four indexed string variables containing

Program Listing

```
20  CLS
30  GOSUB 1100                     '### PRINT BANNER ###
40  PRINT@ 64,"%%";
50  PRINT@ 126,"%%";
60  GOSUB 1100
70  PRINT@ 76,"C H R I S T M A S  E V E  S C E N E";
90  DEFINT A-Z
100  FOR I = 0 TO 30               '### DRAW TREE SIDES ###
110  A = 32 - I : B = 32 + I : Y = 11 + I
120  SET(A,Y)
125  SET(B,Y)
130  NEXT I
140  FOR X = 2 TO 62               '### DRAW BOTTOM OF TREE ###
150  SET(X,41)
160  NEXT X
170  FOR Y = 42 TO 45              '### DRAW STEM ###
180  FOR X = 31 TO 33
190  SET(X,Y)
200  NEXT X
210  NEXT Y
220  FOR X = 27 TO 37              '### DRAW BASE ###
230  SET(X,46)
240  NEXT X
250  FOR X = 28 TO 36
260  SET(X,47)
263  NEXT X
265  X = 24                        '### DRAW TOP DIAGONAL TRIM ###
270  FOR I = 20 TO 22
280  GOSUB 1000
290  NEXT I
295  X = 14                        '### DRAW LOWER DIAGONAL TRIM ###
300  FOR I = 30 TO 36
310  GOSUB 1000
320  NEXT I
330  PRINT@ 811,"M E R R Y";      '### PRINT GREETING ###
340  PRINT@ 871,"C H R I S T M A S";
350  PRINT@ 1008,"FROM DAD";
360  GOTO 2000
400  DATA 32, 13, 26, 26, 42, 31, 15, 37, 38, 37, 24, 22, 38, 32, 13, 34
410  DATA 36, 40, 29, 15, 32, 29, 47, 30, 22, 35, 32, 10, 29, 24, 39, 21
420  DATA 36, 25, 56, 38, 32, 36, 34, 16, 19, 27, 45, 35, 8, 39, 27, 19
430  DATA 39, 28, 17, 32, 45, 39, 31, 18, 28, 30, 50, 33, 19, 40, 36, 19
440  DATA 44, 26, 52, 36, 26, 38
```

the snowfall patterns. Each of these lines contains a maximum of 20 spaces with any combination of snowflake symbols and spaces. (A little experimenting with these lines may make a more interesting display).

Lines 2140 through 2220 are more complex and are needed to make the snow fall outside the window. Variable M indexes one string at a time to determine each specific type of snow and variable L defines the PRINT @ locations for the snowfall.

If the snowfall PRINT @ location is outside the window frame, line 2160 will suppress printing and line 2180 will suppress the erase function in line 2190. Each snowfall string is printed on a line erasing the previous one so it appears the snow is falling.

Line 2195 branches out of the snowstorm routine to line 450 which blinks the tree lights. The data file in lines 400 through 440 contains 35 coordinate pairs that locate each tree light.

A counter in line 450 is tested in line 455 which branches to lines 520 through 530 to restore the data pointer when the file has been read.

Line 460 reads the light location coordinates. The light is extinguished in line 470 and after the time delay in line 480, is lit again in 490.

After a delay in line 500, the program returns to the snowfall routine at line 2200 and continues to alternate between the snowfall and light blinking routines until the BREAK key is depressed.

As you can see, for all the complexity of action the program is not too elaborate. I hope my explanation of the graphics clarifies any questions. I found the use of the graphics layout sheet in the Level II manual extremely helpful in setting up the display layout. ■


```
450  E = E + 1                    '### BLINK LIGHTS ###
455  IF E = 36 GOTO 520
460  READ F, G
470  RESET(F, G)
480  FOR I = 1 TO 60 : NEXT I
490  SET(F, G)
500  FOR I = 1 TO 20 : NEXT I
510  GOTO 2200
520  RESTORE
525  E = 0
530  GOTO 450
1000  FOR C = 1 TO 5            '### DRAW SEGMENT OF TRIM SUBROUTINE ###
1010  SET(X + C, I)
1020  NEXT C
1025  X = X + C
1030  RETURN
1100  FOR I = 1 TO 64           '### BANNER OUTLINE SUBROUTINE ###
1110  PRINT "%";
1120  NEXT I
1130  RETURN
2000  FOR X = 72 TO 117         '### DRAW WINDOW ###
2010  SET(X, 11)
2020  SET(X, 27)
2030  NEXT X
2040  FOR Y = 11 TO 27
2050  SET(72, Y)
2060  SET(73, Y)
2070  SET(116, Y)
2080  SET(117, Y)
2090  NEXT Y
2100  A$(1) = " .    .    .    . "  '### PRINT SNOW STORM ###
2110  A$(2) = "   .    .    . "
2120  A$(3) = " .  .      . "
2130  A$(4) = "   .    .   . "
2140  FOR M = 1 TO 4
2150  FOR L = 293 TO 613 STEP 64
2160  IF L>549 THEN 2180
2170  PRINT@ L, A$(M);
2180  IF L - 64<292 THEN 2200
2190  PRINT@ L - 64, "                    ";
2195  GO TO 450
2200  NEXT L
2210  NEXT M
2220  GO TO 2140
2230  END
```

Can't remember what programs are where? Mumford Microsystems may be able to help.

Disk Directory

Richard K. Riley
PO Box 2227
Augusta ME 04330

The best thing since sliced bread has just come down the pike. If you have a TRS-80 32K disk system, you should run, not walk to the nearest mailbox with a check for $20.70 made out to Mumford Micro Systems for their program "Disk Directory."

Keeping Track

When I finally got my disk from Radio Shack, I immediately put all my taped programs on it so I would never have to type "CLOAD" again.

But how do you keep track of all those programs on disk? Do you type DIR and then copy the information from the screen? That gets old, and it never seems up to date. I tried that and kept 5 × 8 index cards between each of my 50 disks in a plastic 5 × 8 card file box. This system works, but if you want a particular program, you have to paw through all the disks and cards to find it.

Along comes my salvation. The Mumford Micro Systems "Disk Directory" is a very efficient record keeper. The Mumford sorts data alphabetically either by file or by disk, holds 280 files in a 32K system, or 850 in a 48K system and either displays the files on screen or sends them to your printer.

Now for the big surprise. The Mumford reads the information off the disk directory. All you do is insert your disk, give it an identifying name or code, hit enter, and it loads the disk system (eg. TRSDOS, NEWDOS, FORMAT, or whatever), the file names (with any suffix and protection) and then tells you how many free granules are left.

What's the Secret

How do they do it? The program (it's in BASIC) is heavy in peeks and pokes, with a USR(0) and several lines of numeric data statements. (These people

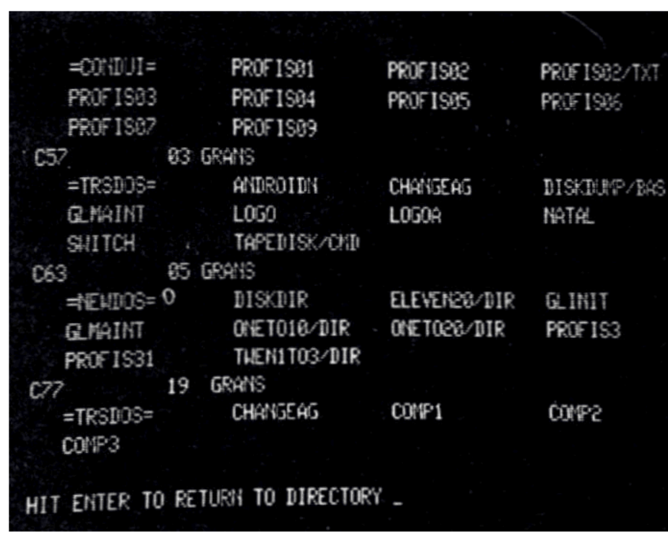


Photo 2. Listing by Disk

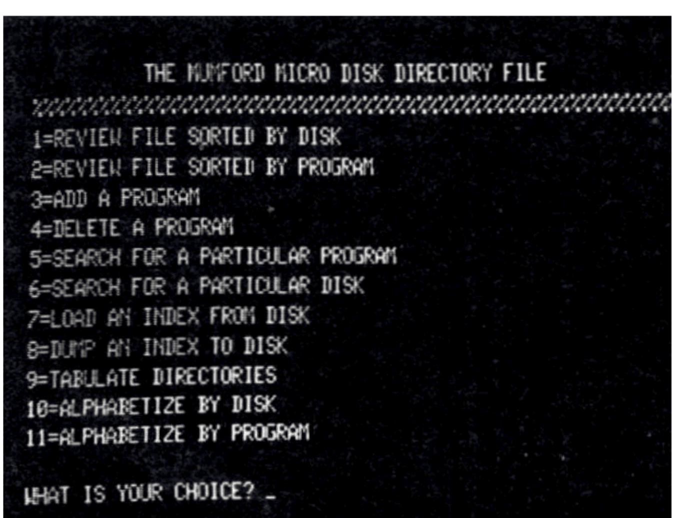


Photo 1. Mumford Micro DISK Menu

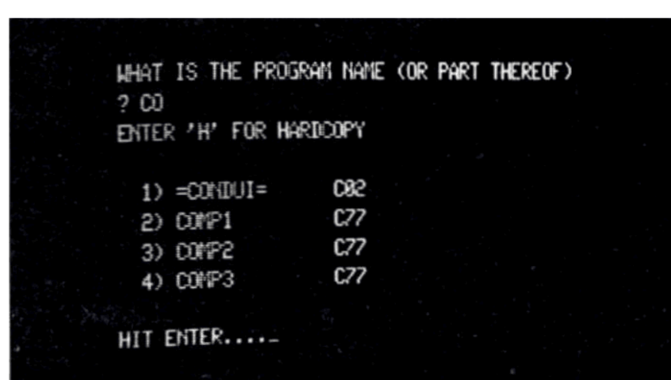


Photo 3. Search for program by portion of name.

```
03 GRANS       69
05 GRANS       63
06 GRANS       71
09 GRANS       61
10  GRANS      73
10  GRANS      74
12  GRANS      70
14  GRANS      80
15  GRANS      65
15  GRANS      67
17  GRANS      75
19  GRANS      77
28  GRANS      76
30  GRANS      79
39  GRANS      68
41  GRANS      72
67  GRANS      78
=DPF=          69
=FILEJU=       74
=FORMAT=       76
=FORMAT=       78
=NEWDOS=       65
=NEWDOS=       68
=NEWDOS=       63
=NEWDOS=       61
=NEWDOS=       70
=NEWDOS=       73
=TRSDOS=       67
=TRSDOS=       71
=TRSDOS=       72
=TRSDOS=       75
=TRSDOS=       77
=TRSDOS=       79
=TRSDOS=       80
ALLSTAR1       73
ALLSTAR1       65
ALLSTAR2       73
ALLSTAR2       65
ALPHA          70
ALPHA2         70
ASI            73
BACKUP/CMD     69
BASIC/CMD      69
BUDGET/BAS     69
CATMAINT       67
CATMAINT       80
CFINDER/BAS    69
```

Example 1.

know much more about the TRS-80 systems than I ever dreamed about.)

The best I can understand, Mumford is able to poke a subroutine that will call the DOS "DIR" program while retaining control in their own program. The program reads the screen through peek statements and stores the information in an array. It does the same thing with the "FREE" command and, again, look at the screen for an answer.

When you load all your disks in this file, you can sort, save, modify or list your files ten different ways (Photo 1).

I number my disks sequentially as I buy them. I label each by its disk number and subject (Table 1).

GA = Games
BU = Business
US = US Naval Reserve
CO = "Conduit" translations
UT = Utility programs
HO = Home
DE = Demonstrations

Table 1.

For example, a disk coded 11HO is the first side of the sixth disk I bought, and it stores my home programs on it: records; finances; etc. This allows me to keep the same numbers on the disk at all times, no matter what its contents and change the suffix only if I change its programs.

I number the front of the disk odd and the back even. Yes, I use both sides of my disks, the poor man's approach to floppy disks. I punch another two read holes and a write protect notch in the disk envelope (not necessary on the Percom-type drives). I don't know if there is any danger in this method, but I have operated disks for almost a year on both sides and since my buffered mod I have never lost a program.

I compiled my files by suffix (all the home disks together) and wrote the files to disk. A friend who has a 48K system let me load all my files together, sort them and now I have a master list of over 400 programs by disk number and name that will either display or print (Photo 2 and Examples 1 & 2).

Searching by String

Looking in the file for particular programs or disks is nice, but I didn't use this feature at first. After all, it is easier to look at the printed lists.

I re-read the instructions and found that the program will search by string section. For example, I could find all the programs with a /PCL in it. I have used this to find all the duplications (backups and progressive saves) of programs. I can find all the versions of the COMP program this way (Photo 3).

This program does have side benefits. The sort routine is very fast, but it still takes a long time to sort 280 records. The Mumford has a nice subroutine that "buzzes" the expansion interface cassette relay when the sort is completed. The sound is noticeable but not annoying and allows you to watch TV or clean up the computer room without glancing back at the screen to see if the sort is completed.

```
61             09   GRANS
   =NEWDOS=           CHECKBAL       LSTATS2        PORKBARL
   PSYCHO             STATS2
63             05   GRANS
   =NEWDOS=           DISKDIR        ELEVEN20/DIR   GLINIT
   GLMAINT            ONETO10/DIR    ONETO20/DIR    PROFIS3
   PROFIS31           TWEN1TO3/DIR
65             15    GRANS
   =NEWDOS=           ALLSTAR1       ALLSTAR2       CUST/DAT
   DISKDUMP/BAS       EDT48723/CMD   INV/DAT        INV1/DAT
   INV2/DAT           INV3/DAT       SWITCH
67             15    GRANS
   =TRSDOS=           CATMAINT       GL             GLBALSHT
   GLINIT             GLMAINT        GLMAINT2       GLTXPOST
   INCOME             TXREPORT
68             39    GRANS
   =NEWDOS=           D61T67         DISKDIR
69             03   GRANS
   =DPF=              BACKUP/CMD     BASIC/CMD      BUDGET/BAS
   CFINDER/BAS        CHECKING/BAS   CKLDR/BAS      DATA/DAT
   FLLDR/BAS          LOADER/BAS     PF/BAS         SAVINGS/BAS
```

Example 2.

The relay it uses selects between cassette 1 and cassette 2 in the interface. Since I got my disk system, I have not used the dual cassettes and only use the one input in my CPU.

The Mumford prompted me to get my act together and put all the games on certain disks, the business programs on others, a welcome bonus.

The address is Mumford Micro Systems, Box 435, Summerland CA 93067. ■

Play tunes on your cassette in minutes—without any extra hardware!

Music Maestro!

Roger L. Pape
7545 Marble Drive
Liverpool NY 13088

Music generation has become a popular activity for computer hobbyists. A variety of hardware peripheral boards are marketed specifically for this purpose. Owners of Radio Shack TRS-80s with Level II BASIC can be generating music on their systems after only a few minutes of programming, as described below. No added computer hardware is required; you simply make use of the cassette output port already available. But, rather than the raspy sounds generated by a CSAVE, a simple machine-language routine generates tones of respectable quality.

Existing Hardware Mods

The audio section can consist of whatever you have available. Lacking anything else, you can record the output on cassette and then play it back off-line. On the other hand, to listen to the sounds directly as they are generated, connect high-impedance earphones to the cassette output plug. (An inexpensive crystal earplug provided with transistor radios works great!) Since I am using an old TV set for a video monitor (instead of the $200 Radio Shack monitor), I simply added a jack tied to the audio portion of the TV set, which was still intact, and played the sounds through it.

The TRS-80 uses a direct recording approach for the cassette I/O. The output circuitry is sheer simplicity, as shown in Fig. 1. The data output word is fed into a data latch with a resistor network tied across the outputs of the lowest two bits. When the lowest bit is set (01 out), the output voltage is high. When the next bit is set (02 out), the output voltage is low. When both are reset (00 out), the voltage is at a mid-level. Radio Shack's specs call for a 0.8 volt peak-to-peak output level at 1k Ohm. To generate a tone, just toggle between any two of these levels at a controlled rate.

Incidentally, the next higher data bit (D_2) controls the relay, which turns the cassette drive on and off. Outputting 04 (or any value with bit 2 set) to the port closes the relay, turning the cassette on. Resetting the bit turns it off.

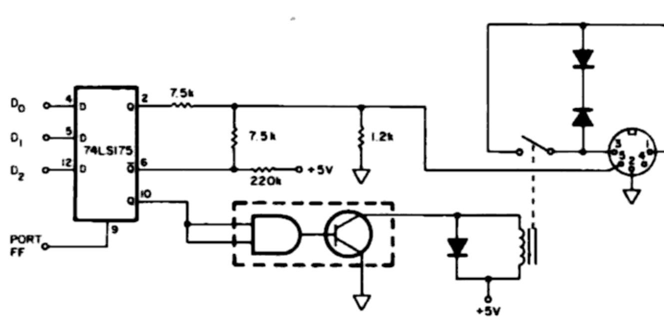


Fig. 1. TRS-80 cassette output circuitry.

Generating Simple Tunes

To convince myself that music generation was feasible, I wrote the short machine-language routine shown in Fig. 2, which can quickly be POKEd into upper memory. Since my objective was to have a routine callable by BASIC, I used the USR function linkage. Details of the USR function are given in the Radio Shack Level II BASIC manual. A single 16-bit argument is passed from the BASIC program to the routine in locations 4121_H and 4122_H (lower eight bits in the lower memory location). The Level II ROM includes a routine starting at location $0AF7_H$ that fetches the argument and returns it in the HL register pair.

For simplicity, the lower eight bits (in L) were used to control the length of a half cycle (i.e., the pitch); while the upper eight bits (in H) were used to specify the number of cycles (i.e., the duration of the note). After setting an output level, the value from L is counted down for a half cycle, after which the level is toggled. This process is continued until the value of H is counted down to zero.

Fig. 3 gives a BASIC driver program with a sample tune. The machine-language routine of Fig. 2 is POKEd into the last available memory space (depending on whether you have a 4K or 16K system) with the starting address POKEd into locations 16526_{10} and 16527_{10} to provide the linkage for the USR function call. The numbers required to generate the notes of an octave were determined as follows.

All 1s in the lower bytes (255_{10}) are used for the lowest note (coincidentally, this results in about 220 Hz or the A below middle C). For an equally tempered scale, the 12 half-step intervals in an octave are equally divided, giving a frequency ratio between adjacent notes a half-step apart equal to the twelfth root of 2, or 1.05946. To maintain a constant duration of the notes as the pitch changes, the number of cycles per note (upper eight bits) must be increased as the length of a cycle decreases. In other words, the product of the two bytes should be constant. The resulting in-

```
NOTE    CALL 0A7FH      ;PUTS USR ARGUMENT IN HL
L1      LD A,05H        ;SET FOR + CASSETTE OUTPUT (DRIVE ON)
        OUT (0FFH),A
        LD B,L          ;PUT LOWER 8 BITS IN B
        DJNZ $          ;COUNT DOWN FOR DELAY
        LD A,04H        ;SET FOR 0 CASSETTE OUTPUT (DRIVE ON)
        OUT (0FFH),A
        LD B,L          ;SAME DELAY
        DJNZ $
        DEC H           ;HOLD NOTE LONGER?
        JR NZ,L1
        RET
        END
```

Fig. 2. Z-80 assembly listing a simple Tone Generation routine.

teger value is the 2's complement equivalent of the combined 16 bits.

The notes of the tune are stored in the final DATA statements with 1 through 13 corresponding to the notes between the lower A and one octave higher. A zero is used to designate a rest. The program reads the indices in sequence and passes the corresponding integer value from the N% array to the Tone Generation routine by calling the USR function.

Memory space must be reserved for the machine-language routine at initialization. When the system is turned on and MEMORY SIZE? appears, enter 32746 (or less) for a 16K system or 20458 for a 4K system. Load the program and RUN. You should have no problem recognizing the tune. Just for the fun of it, you might replace the note-reading statement (line 100) with:

100 K = RND(13):X = USR(N%(K)) : GOTO 100

to generate an "abstract" tune.

An Improved Routine

Although the approach described above does a surprisingly good job of generating simple tunes, it is limited in the range of notes that can be generated and by the difficulty in changing note lengths. Therefore, the obvious step is to develop a routine with more flexibility.

The basic approach I adopted was to provide three values to the Tone Generation routine: the note number within an octave, the octave number and the note length in 1/16ths. The three fields in the 16-bit USR function argument are shown in Example 1.

The half-cycle count for each of the 12 tones in the lowest octave is stored internally as a table of 16-bit values, and the note number is used as an index in this table. To increase the tone one octave, the length of a cycle is divided by 2 (i.e., the count is shifted right 1 bit). The octave number controls the number of right shifts. The duration of a note is determined by accumulating half-cycle counts until the upper eight bits reach a value corresponding to the length of a 1/16 note and repeating the process for the specified number of 1/16 notes total duration. The number matched for the basic 1/16 note length determines the tempo.

Fig. 4 shows the Z-80 assembly-language listing of the Tone Generation routine that evolved. One of the more challenging aspects was to develop position-independent code so that the routine could be easily loaded into any area of memory. The relative jumps in the Z-80 instruction set simplify the problem; however, referencing an internal table presents a slight problem.

Rather than use an absolute memory reference, the objective is to latch onto the value of the program counter within the routine and add the offset to the table. That is the purpose of the CALL in the first statement. A short two-instruction sequence is provided near the bottom of the Level II ROM as follows:

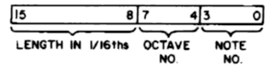
```
000A;     POP HL
          JP (HL)
```

A subroutine CALL places the value of the PC immediately following the CALL statement on the stack for a return linkage. Then, the routine at $000A_H$ places this value in the HL register pair and returns via the JP (HL) instruction. The note table was placed where it is to simplify the calculation of the offset into the table. This feature added only four bytes and a few microseconds to the routine.

The remainder of the routine is relatively direct. Note that the RRD instruction provides a convenient means for extracting the 4-bit fields from the argument. No internal checks are provided for out-of-range values (e.g., note number greater than 12). I felt that the calling program could easily provide the necessary checks. The data bytes output to the cassette port are 01 and 00 so that the cassette drive will be turned off. If you really want to use the cassette recorder for the output, the values must be changed to 05 and 04 (or disconnect the REMOTE plug).

The Program

For a routine of this size, it is probably advisable to use an assembler and generate a SYSTEM file with the object code. On the other hand, it is more convenient to have a single, self-contained program in BASIC. For those who prefer a single file (or don't have an assembler), Fig. 5 gives a BASIC program that POKEs the Tone Generation routine into high mem-



Example 1.

ory just as the previous program did.

To simplify the process of transcribing notes into DATA statements, letter designations are used to specify the various notes in an octave (i.e., A through G with # used for sharps and * for flats). Octave numbers can range from 0 to 8 with 3 corresponding to the octave containing middle C. Letters are also used to designate note lengths with S representing a sixteenth note, E an eighth note, Q a quarter note, H a half note and W a whole note.

Therefore, a section of the program (statements 43 to 75) interprets the note designators and packs the corresponding values in a 16-bit integer, which is stored in array M%. The size of this array (statement 40) is limited by the remaining memory space. For a 16K system it can be increased considerably.

The program continues to scan and store the notes until a blank character is read for a note. The tune is then played by sequencing through the values in the M% array. More than one tune can be stored in the DATA statements, separating them with a blank note.

Several features were added to the program to make it more

```
1 REM FOLLOWING SECTION PUTS TONE GENERATION IN HIGH MEMORY
2 REM A1=79 FOR 4K OR 127 FOR 16K
3 A1=127
10 POKE 16526,235:POKE 16527,A1:A=256*A1+235
20 READ B:IF B>=0 POKE A,B:A=A+1:GOTO 20
30 DATA 205,127,10,62,5,211,255,69,16,254,62,4,211,255
31 DATA 69,16,254,37,32,239,201,-1
40 REM FOLLOWING NUMBERS REPRESENT ALL NOTES IN ONE OCTAVE
49 REM LOW 8 BITS ARE PITCH, UPPER 8 BITS ARE DURATION
50 DIM N%(13)
60 FOR J=1 TO 13:READ N%(J):NEXT J
70 DATA 32767,-30735,-28701,-26666,-24374,-21825,-19276
71 DATA -16726,-13664,-10601,-7281,-3705,-129
99 REM FOLLOWING READS NOTE NUMBERS AND GENERATES TONE
100 READ K:IF K>0 X=USR(N%(J)):GOTO 100
110 IF K=0 FOR J=1 TO 100:NEXT J:GOTO 100
119 REM TURN OFF CASSETTE IN CASE IT'S USED
120 OUT 255,0
199 REM PUT NOTE NUMBERS IN DATA STATEMENTS AFTER THIS POINT
200 DATA 1,1,3,1,6,5,0
201 DATA 1,1,3,1,8,6,0
202 DATA 1,1,13,10,6,5,3,0
203 DATA 11,11,10,6,8,6
998 DATA -1
999 END
```

Fig. 3. BASIC program for music generation.

interesting: the ability to change pitch and to change tempo. The pitch can be changed any number of half-step intervals (up or down) by simply adding this offset to each note number and correcting for octave changes (statements 41 and 61-65). The tempo is changed by poking different values into the byte location used for testing for a 1/16 note length (statements 85-89). With these two simple additions, you begin to appreciate the appeal of computer-generated music.

Conclusion

Again, you must allocate sufficient space at the top of memory for the machine-language routine. Since this version requires 114 bytes of memory, the MEMORY SIZE should be set at 32653 for a 16K system or 20365 for a 4K system.

Although the sample tune in the program is somewhat lengthy, be sure to try it. It's a familiar contemporary tune. After you tire of this melody, try entering your own favorites. Soon you'll be thinking of modifications to the program to add new features. That's what it's all about. ■

```
NOTE    CALL 000AH      ;RETURNS PC IN HL (FOR POSITION INDEP)
        JR SETIX        ;SKIP OVER TABLE
TBLN    DEFW 05AAH      ;TABLE OF NOTE LENGTHS (HALF CYCLE)
        DEFW 055AH      ; FOR 1.7741 MHZ CLOCK
        DEFW 050CH
        DEFW 04C4H
        DEFW 0480H
        DEFW 043FH
        DEFW 0402H
        DEFW 03C8H
        DEFW 0392H
        DEFW 035EH
        DEFW 032EH
        DEFW 0300H
SETIX   PUSH HL
        POP IX          ;IX NOW POINTS 2 LOC ABOVE TABLE
        XOR A           ;CLEAR A
        LD D,A          ;CLEAR D
        LD HL,4121H     ;POINT HL TO LOWER BYTE OF USR ARG
        RRD             ;GET NOTE NUMBER (BOTTOM NIBBLE)
        JR Z,REST       ;0 IS A REST NOTE
        LD B,01H        ;SET B FOR + CASSETTE OUTPUT
        LD E,A          ;CALCULATE POSITION IN TABLE
        SLA E           ;2 BYTES PER ENTRY
        ADD IX,DE       ;ADD OFFSET
        LD E,(IX+0)     ;PUT BASE HALF CYCLE LENGTH
        LD D,(IX+1)     ;  IN DE
        RRD             ;GET OCTAVE BITS (NEXT NIBBLE)
S1      JR Z,ADJ        ;SHIFT LOOP FOR CHANGING OCTAVE
        SRL D
        RR E
        DEC A
        JR S1
REST    LD B,A          ;CLEAR B (A STILL ZERO)
        LD E,02H        ;ANY LENGTH IS OK FOR REST
ADJ     DEC DE          ;SUBTRACT 2 TO ACCOUNT FOR
        DEC DE          ;  OVERHEAD OUTSIDE COUNT LOOP
        PUSH DE         ;SAVE ADJUSTED COUNT FOR HALF CYCLE
        POP IX          ;IN IX REGISTER
        INC HL          ;UPPER BYTE OF USR ARG IS NOTE LENGTH
        LD C,(HL)       ;STORE IT IN C
GEN     LD HL,0000H     ;CLEAR HL
GEN1    LD A,B          ;OUTPUT B
        OUT (0FFH),A    ;  TO CASSETTE PORT
        PUSH IX         ;SET COUNT
        POP DE          ;  IN DE
        ADD HL,DE       ;AND ACCUMULATE COUNT IN HL
        XOR A           ;CLEAR A FOR TESTING
D1      DEC DE          ;COUNT DOWN HALF CYCLE
        CP E            ;PREVIOUS INSTR DOESN'T SET ZERO FLAG
        JR NZ,D1
        CP D
        JR NZ,D1
        OUT (0FFH),A    ;OUTPUT 0 TO CASSETTE PORT


        PUSH IX         ;REPEAT ABOVE FOR SECOND HALF CYCLE
        POP DE
D2      DEC DE
        CP E
        JR NZ,D2
        CP D
        JR NZ,D2
        LD A,10H        ;THIS VALUE CONTROLS TEMPO
        CP H
        JR NC,GEN1
        DEC C           ;DECREMENT NOTE LENGTH COUNTER
        JR NZ,GEN       ;AND GO BACK IF MORE
DONE    RET
        END
```

Fig. 4. Z-80 assembly listing of a more flexible Tone Generation routine. There is a minor error in this listing, but it does not affect the operation of the program. The routine in ROM, which is called in the first line to fetch the program counter from the stack, starts at location $000B_H$ instead of $000A_H$. The contents of location $000A_H$ is 40_H, which is equivalent to a Z-80 instruction code of LD B, B. This is essentially a "no operation" and does not cause any problem. Perfectionists will want to change the first line to: CALL $000B_H$.

```
1 REM FOLLOWING SECTION PUTS TONE GENERATION IN HIGH MEMORY
2 REM A1=79 FOR 4K OR 127 FOR 16K
3 A1=127
10 POKE 16526,142:POKE 16527,A1:A=256*A1+142
20 READ B:IF B>=0 POKE A,B:A=A+1:GOTO 20
30 DATA 205,10,0,24,24,170,5,90,5,12,5,196,4,128,4,63,4,2
31 DATA 4,200,3,146,3,94,3,46,3,0,3,229,221,225,175,87,33
32 DATA 33,65,237,103,40,24,6,1,95,203,35,221,25,221,94,0
33 DATA 221,86,1,237,103,40,10,203,58,203,27,61,24,247,71
34 DATA 30,130,27,27,213,221,225,35,78,33,0,0,120,211,255
35 DATA 221,229,209,25,175,27,187,32,252,186,32,249,211,255
36 DATA 221,229,209,27,187,32,252,186,32,249,62,16,188,48
37 DATA 224,13,32,218,201,-1
39 REM CONVERT NOTE DESIGNATORS TO INTEGER INDEX
40 DIM M%(150):MN=150
41 INPUT"SHIFT PITCH HOW MANY HALF TONES";S
42 K=0
43 READ N$,O,L$:IF N$=" " GOTO 80
44 N=0:L=1:IF O<0 OR O>8 O=3
45 IF N$="A" N=1:GOTO 60
46 IF N$="A#" OR N$="B*" N=2:GOTO 60
47 IF N$="B" N=3:GOTO 60
48 IF N$="C" N=4:GOTO 60
49 IF N$="C#" OR N$="D*" N=5:GOTO 60
50 IF N$="D" N=6:GOTO 60
51 IF N$="D#" OR N$="E*" N=7:GOTO 60
52 IF N$="E" N=8:GOTO 60
53 IF N$="F" N=9:GOTO 60
54 IF N$="F#" OR N$="G*" N=10:GOTO 60
55 IF N$="G" N=11:GOTO 60
56 IF N$="G#" OR N$="A*" N=12
59 REM SHIFT PITCH AND CHECK FOR OCTAVE CHANGE
60 IF N=0 GOTO 70
61 N=N+S
62 IF N>12 N=N-12:O=O+1:GOTO 62
63 IF N<1 N=N+12:O=O-1:GOTO 63
64 IF O<0 O=0
65 IF O>8 O=8
69 REM CONVERT LENGTHS TO MULTIPLES OF 1/16
70 IF L$="E" L=2:GOTO 75
71 IF L$="Q" L=4:GOTO 75
72 IF L$="H" L=8:GOTO 75
73 IF L$="W" L=16
74 REM GENERATE AND STORE COMBINED VALUE
75 K=K+1:M%(K)=N+16*O+256*L:IF K<MN GOTO 43
80 IF K=0 STOP
85 INPUT"WHAT TEMPO(S,M,F)";T$
86 T=256*A1+240  193
87 IF T$="S" POKE T,20
88 IF T$="M" POKE T,16
89 IF T$="F" POKE T,8
99 REM FOLLOWING STATEMENT PLAYS TUNE
100 FOR J=1 TO K:X=USR(M%(J)):NEXT J:GOTO 42
199 REM PUT NOTES IN DATA STATEMENTS FOLLOWING THIS POINT


200 DATA D,3,E,D,3,E,D,3,E,G,3,H,D,4,H,C,4,E,B,4,E,A,4,E
201 DATA G,4,H,D,4,Q,C,4,E,B,4,E,A,4,E,G,4,H,D,4,Q
202 DATA C,4,E,B,4,E,C,4,E,A,4,Q,R,0,Q,D,3,E,D,3,S
203 DATA G,3,H,D,4,H,C,4,E,B,4,E,A,4,E,G,4,H,D,4,Q
204 DATA C,4,E,B,4,E,A,4,E,G,4,H,D,4,Q,C,4,E,B,4,E,C,4,E
205 DATA A,4,Q,R,0,Q,D,3,E,D,3,S,E,3,Q,E,3,E,C,4,E,B,4,E
206 DATA A,4,E,G,3,E,G,3,E,A,4,E,B,4,E,A,4,E,E,3,E,F#,3,Q
207 DATA D,3,E,D,3,S,E,3,Q,E,3,E,C,4,E,B,4,E,A,4,E,G,3,E
208 DATA D,4,Q,A,4,H,D,3,E,D,3,S,E,3,Q,E,3,E,C,4,E,B,4,E
209 DATA A,4,E,G,3,E,G,3,E,A,4,E,B,4,E,A,4,E,E,3,E,F#,3,E
210 DATA R,0,Q,D,4,E,D,4,S,G,4,E,F,4,S,E*,4,E,D,4,S
211 DATA C,4,E,B*,4,S,A,4,E,G,3,S,D,4,H,R,0,Q
212 DATA D,3,E,D,3,E,D,3,E,G,3,H,D,4,H,C,4,E,B,4,E,A,4,E
213 DATA G,4,H,D,4,Q,C,4,E,B,4,E,A,4,E,G,4,H,D,4,Q
214 DATA C,4,E,B,4,E,C,4,E,A,4,H,D,3,E,D,3,S,G,3,H,D,4,H
215 DATA C,4,E,B,4,E,A,4,E,G,4,H,D,4,Q,G,4,E,F,4,E,E*,4,E
216 DATA B,5,H,A,5,Q,G,4,E,R,0,Q,G,3,E,G,3,E,G,3,Q
998 DATA " ",0," "," ",0," "
999 END
```

Fig. 5. Improved BASIC program for music generation.

SMALL SOFTWARE SYSTEM TRS-80 PRODUCTS SMALL SOFTWARE SYSTEM

RSM-2: AN ADVANCED TAPE MONITOR FOR 16K TRS-80'S - $26.95
RSM-2D: THREE MONITORS FOR TRS-80 DISK SYSTEMS - 29.95
RSM-2 RELOCATOR: PUT RSM-2/2D ANYWHERE IN MEMORY - 9.95
RSM RS-232-C: PRINT WITH RS-232-C FROM RSM-2/2D - 9.95

22 commands to control your TRS-80 Z-80 processor! Examine ROM's, test RAM, program in machine language, read/write machine language tapes, and much more! SYMBOLIC DUMP will disassemble memory into Z-80 mnemonics! HEX and two ASCII memory dumps; EDIT, MOVE, EXCHANGE, VERIFY, FILL, ZERO, TEST, or SEARCH memory, read/write SYSTEM tapes, enter BREAKPOINTS, PRINT with TRS232 or Centronics, and read/write disk sectors directly! Radio Shack uses RSM; see page 4-17 of your disk manual! RSM-2 tape loads at top of 16K LEVEL I or II; RSM-2D disk includes 3 versions for 16K, 32K and 48K.

ADVENTURE! $14.95 each, 3 for $39.95, all 5 for $64.95.

Machine language versions of Adventure, the current rage of the big time-sharing computers! 5 versions, 1-Adventureland, 2-Pirate's Adventure, 3-Mission Impossible, 4-Voodoo Castle, 5-The Count. A challenge that can take weeks to solve! LEVEL-II 16K.

BARRICADE: MACHINE LANGUAGE ACTION GAME FOR TRS-80'S - $14.95

Break through 5-wall Barricade with high-speed ball and keyboard controlled paddle! Trap the ball among the walls and watch it destroy the 100 blocks! Select 96 different options to challenge experts and beginners. 3 scores with the best of each saved to be challenged by other players. NOBODY can achieve the maximum WEIGHTED SCORE of 33,000! 4K LEVEL-I and II.

AIR RAID: MACHINE LANGUAGE TRS-80 SHOOTING GALLERY! - $14.95

Shoot down high speed aircraft with a ground based missile launcher! Aircraft explode dramatically when hit, sometimes destroying other nearby planes! Score is tallied for each hit or miss, and the highest score is saved to be challenged by other players. Hours of fun for you, and a super demonstration program for entertaining friends! 4K LEVEL I and II.

BASIC-1P - LEVEL-1 BASIC WITH PRINTING! - $19.95

Run any LEVEL-1 BASIC tape (12K or less) in 16K LEVEL-II TRS-80 without conversion! Plus LLIST and LPRINT for TRS232, RS-232-C or Centronics printers. LPRINT ON and LPRINT OFF prints anything that you see on the screen! All LEVEL-I abbreviations and functions supported.

*** CALIFORNIA RESIDENTS ADD 6 PER CENT SALES TAX ***

*** SMALL SYSTEM SOFTWARE * P.O. BOX 366 * NEWBURY PARK, CALIF. 91320 ***

TRS232 PRINTER INTERFACE - $49.95 (+$2.00 shipping)

Assembled and tested output port for TRS-80 printing. Use any RS-232 or 20-mil current loop ASCII printer. Expansion interface not required. Use with LEVEL-II BASIC, CP/M, BASIC-1P, ELECTRIC PENCIL, RSM-2/2D or your own programs! Standard cassette software included, or order new "FORMATTER" (below) for enhanced printer control.

TRS232 "FORMATTER" SOFTWARE PACKAGE - $14.95

Page and line length control, form feed function, printer pause, 9 baud rates, "smart" line termination, indented continuation lines, built-in keyboard debounce, software control of screen printing, etc.

THE ELECTRIC PENCIL FOR TRS-80 DISK SYSTEMS - $150.00
THE ELECTRIC PENCIL FOR TRS-80 TAPE SYSTEMS - 99.95
PENCIL RS-232-C: ADAPT TAPE PENCIL FOR RS-232-C - 9.95

Write text, delete, insert, or move words, lines or paragraphs, save text on tape (or disk), then print formatted copy with our TRS232 or Centronics printer (RS-232-C with disk version). Right justification, page titling and numbering, transparent cursor and repeating keyboard. Lowercase entry and display with modification. Tape version runs on Level I or II 16K.

CP/M OPERATING SYSTEM WITH TRS232 SOFTWARE - $145.00

SMALL SYSTEM SOFTWARE/LIFEBOAT ASSOCIATES version of CP/M. Includes TRS232 and RS-232-C software, lower-case support, debounce, DCV-2 and other unique utilities. CP/M Editor creates and modifies all files. Assemble directly from disk, placing HEX and PRINT files back onto disk! Includes DDT (Dynamic Debugging Tool), PIP (Peripheral Interchange Program), and more! CP/M is a trademark of Digital Reasearch, Inc.

DCV-1: CONVERT SYSTEM PROGRAMS TO DISK FILES - $9.95

Execute Adventure, Barricade, Air Raid, RSL-1, ESP-1, T-BUG, etc., from disk, even if they interfere with TRSDOS! After using DCV-1, your program loads from disk into high memory, moves itself to its correct address, then jumps there and executes! New version works with TRSDOS 2.2.

OTHER TRS-80 PRODUCTS

ESP-1: $29.95 Editor, assembler, and monitor using INTEL 8080 mnemonics.
RSL-1: 14.95 Draw patterns, then play Conway's LIFE in machine language.
LST-1: 8.00 A disassembled listing of LEVEL-1 BASIC with some comments.

*** SMALL SYSTEM SOFTWARE * P.O. BOX 366 * NEWBURY PARK, CALIF. 91320 ***

"TRS-80 is a registered trademark of TANDY CORP." ✓30

SECURITY for TRS-80 DISC DRIVE OWNERS

- **BUSINESSMEN** - **Ensure the privacy of your Corporate Files**
- **PROGRAMMERS** - **Protect Programs/Data Files**
- **RS232 USERS** - **Foil wiretaps with Super Cipher**

CIPHER **$39.95** for security at the Confidential level. Cipher is a sophisticated cryptographic system which is impervious to all Master DOS passwords. Codes of up to 11 alphanumeric characters accepted. Cipher is supplied with its own self-chaining disk operating system, but will accept DOS 2.1, 2.2 or 2.3 files.

SUPER CIPHER **$99.95** for security at the Top Secret level. Super Cipher accepts codes of up to 256 alphanumeric characters. This is the code for special business or military applications.

AUTHORS — DISTRIBUTORS — SYSTEM HOUSES

- **DOUBLE AND TRIPLE YOUR SALES**
- **CURTAIL COPYRIGHT VIOLATIONS**
- **PREVENT ILLEGAL DISC DUPLICATION**
- **THWART PROGRAM EXAMINATION DURING EXECUTION**

We will customize your program so that it will execute, but can not be examined regardless of any effort to disable the protection. Our technique is impervious to both Master DOS passwords and alien DOS systems. Send us your program together with a $40 encipherment fee. If we consider the cryptographic problem acceptable, we will encipher your program and return the encrypted version to you. Otherwise your $40 fee will be promptly refunded. Examine and test our diskette - complete with its own copyrighted DOS system - to see if the level of encipherment is adequate. If you are satisfied we will then supply duplicates in lots of fifty at a fixed charge per diskette.

P.O. Box 516 • Troy, Idaho 83871 • **SOLARIS PRESS** • Specify 32K or 48K • (208) 835-5391 ✓36

The University of Loyola runs a computer education course, using 25 TRS-80 systems.

Night School

Dr. Antonio M. Lopez, Jr.
Dept. of Math Sciences
Loyola University
New Orleans LA 70118

As more personalized computers—with their latest in technological advances—become available in today's marketplace, it is becoming more evident that the human element is not keeping pace. When Radio Shack announced its TRS-80, I was one of the first to make my way down to my local Radio Shack, only to find a store manager with a product he did not know how to sell. When I asked to see the product, his response was, "Do you know how to program this thing?" That was the beginning of what I hope will be a long and profitable friendship for both of us. Let me explain what I mean.

Loyola University is fortunate to be in the forefront of microcomputer use in the New Orleans area. We have a variety of systems—an Altair, a couple of KIM systems, two Processor Technology SOL systems, two Apple II systems, a Commodore PET and two TRS-80 Level II BASIC systems. These microcomputer systems see heavy use daily at Loyola, which offers two degree programs in the department—a BS in computer science and a BS in mathematics. When they are not being used in advanced course work, these systems are used in research, with Heuristics' Speech-lab on our own homemade plotter, or they are down for upgrading. There is no time available to run a large-scale continuing education program for the general public off these systems; however, if I had 25 systems of the same kind. . . .

After some good old-fashioned bartering, I was able to convince both the district manager of Radio Shack and the director of continuing education that it would be in the common interest to hold a six-day workshop on the TRS-80 Level I BASIC unit, which retails for under $600. Radio Shack would provide 25 systems and 25 store managers; Loyola would provide the instruction and be allowed to open the workshop to the public for a nominal fee of $65. This cooperation on the local level provided 36 paying customers the first time the course was offered in June. In July, when it was offered again, we had 48 paying customers. The director of continuing education has slated the course again for later this year.

The education level of the audience varies greatly in these workshops—from a nine-year-old boy to a 67-year-old, retired furniture-store owner; from CPAs, doctors and lawyers to homemakers, hobbyists and the just plain curious. The common denominator is that they know nothing about computers, but they want to learn.

Workshop Format

The format for the workshop is a combination of lectures and supervised labs (Fig. 1). It is intended that the course run for two consecutive weeks on Monday, Wednesday and Thursday nights, but I do not feel that this is absolutely necessary as long as enough "think time" is allowed between sessions. A good deal of information is covered in the lecture periods, more than the average person can comprehend immediately. At the end of each lecture period the participants are handed an exercise set which is divided into two sections—problems that all are expected to do and problems to do if they are bored. The lab periods are scheduled for the first hour of the sessions for two reasons. First, the participants have had time to think about the exercises and are now eager to try their solutions. Second, once someone starts using

	Mon.	Wed.	Thur.	Mon.	Wed.	Thur.
FIRST HOUR	LECTURE	LAB	LAB	LAB	LAB	LAB
SECOND HOUR	LECTURE	LECTURE	LECTURE	LECTURE	LECTURE	LECTURE

Fig. 1.

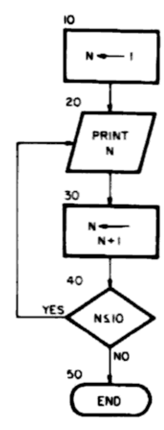


Fig. 2.

a microcomputer system, it is difficult to tear him away. If you want to go home at a reasonable hour, you do not dare schedule the lab period for the last hour. For this workshop, I use two student assistants, usually a math major and a computer-science major. During the lab periods, we circulate from station to station answering both programming and technical questions and giving hints on how to solve problems that are being encountered. We endeavor to install a "try-it" attitude among the participants.

A two-hour lecture the first night enables me to cover a variety of topics. I like to point out to my audience why we have chosen the TRS-80 to work with:

1. Radio Shack has supplied us with the 25 systems.

2. The TRS-80 requires no engineering know-how to assemble and run. In fact, it can be set up and programming begun in 15 minutes.

3. The TRS-80 is expandable with a line of products to fit varying needs and there is no worry about interface and compatibility.

4. The TRS-80, including the Level I BASIC unit, is capable of handling business-oriented problem solving.

5. Finally, the TRS-80 is locally stocked. Should a power unit or a CPU fail, we have 35 Radio Shack dealers in New Orleans—surely one will have a replacement part. I do not have to ship my unit halfway across the U.S. for repair.

Next, we talk about problem solving with the TRS-80 and divide our endeavors into the two classical areas of decision making and actual data manipulation. The next two nights we concentrate on the TRS-80 as a tool for decision making; the following two nights are dedicated to the TRS-80 in the data-processing environment. Our problem-solving procedure depends heavily on flowcharts and the ability of the participant to see his solution as a sequence of "atomic" BASIC instructions.

It is at this point that I encounter the biggest audience awakening—"You mean you have to do all that to get it to print out the numbers 1 through 10?" (Fig. 2).

With this all behind us, we still have time to examine some of the special function keys and the two existing modes of the TRS-80 BASIC Level I unit—the calculator mode and the program mode. To the novice it is incredible that you can do arithmetic calculations "over" your program without "damaging" the program.

Lectures and Programs

By now, my two hours are just about up and my audience is really ready for action. It is very important to start with a program that is not too difficult; yet, not too easy for the entire group. I have found that a compound-interest problem is just about the right level (Fig. 3). It illustrates all the basic flowchart symbols, plus the idea of a loop. The scenario for this problem is that you wish to deposit a certain sum in a savings account that pays 5¾ percent per annum, compounded quarterly. How much will you have in the bank after 20 years? I also introduce the idea of blocks of the flowchart actually becoming lines of code in BASIC—the block numbers match the line numbers (Fig. 4).

By the second lecture session, confusion seems to be giving way to confidence and a desire to "try it" and see what happens. I have very little time to dwell on this newfound knowledge, but I do emphasize that those who do not wish to learn to program their own personal computers are faced with lines of software that include such greats as: Backgammon, Blackjack, Quick, Watson, and the Home Recipe Program. Personal computers like the TRS-80 are *not* toys unless you want them to be toys. My lectures will now center around different types of programs.

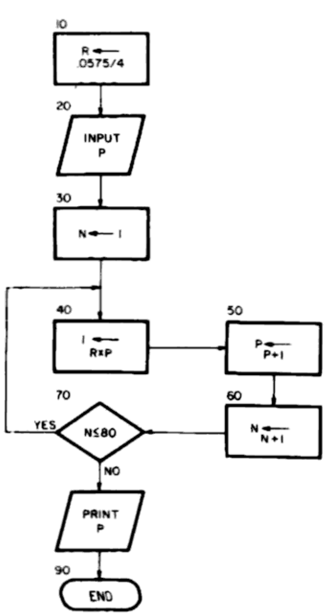


Fig. 3.

```
10      LET R =  0575/4
20      INPUT "WHAT IS THE PRINCIPAL ";P
30      LET N = 1
40      LET I = R * P
50      LET P = P + I
60      LET N = N + 1
70      IF N <= 80 THEN GOTO 40
80      PRINT "YOUR DEPOSIT IS NOW ";P
90      END
```

Fig. 4.

```
10   INPUT "WHAT IS THE COST OF YOUR ASSET ";C
20   INPUT "WHAT IS THE USEFUL LIFE IN YEARS ";N
30   INPUT "WHAT IS THE EXPECTED SALVAGE VALUE ";S
40   I = N
50   A = (C-S)/N
60   PRINT TAB(20);"DEPRECIATION SCHEDULE"
70   PRINT "YEAR";TAB(10);"STRAIGHT LINE";TAB(30);"SUM OF YEARS DIGITS"
80   Y = 1
90   D = (2*I*(C-S))/(N*(N-1))
100  PRINT TAB(2);Y;TAB(13);A;TAB(35);D
110  IF Y = N THEN END
120  I = I-1
130  Y = Y+1
140  GOTO 90
```

Fig. 5.

```
10      INPUT "WHAT IS YOUR CLIENT'S NAME "; A$
20      INPUT "WHAT IS THE AMOUNT OF THE LOAN "; L
30      INPUT "WHAT IS THE NUMBER OF MONTHLY PAYMENTS ";N
40      READ R
41      DATA 0.0079
42      REM .0975/12  CURRENT MONTHLY INTEREST RATE
50      A = R + 1
60      T = 1
70      FOR I = 1 TO N
80      T = T*A
85      NEXT I
90      P = (L*R*T)/(T-1)
100     PRINT A$;" YOUR MONTHLY PAYMENT IS $";P
110     END
```

Fig. 6.

```
10      INPUT "WHAT IS A NUMBER BETWEEN 1 AND 100 ";N
20      FOR I = 1 TO N
30      R = RND(0)
35      NEXT I
40      T = 0
50      FOR K = 1 TO 10
55      CLS
60      A = RND(10)
70      B = RND(10)
80      C = A+B
90      PRINT TAB(5);A
91      PRINT TAB(4);"+";B
92      PRINT TAB(4);"-----"
100     INPUT "     ";D
110     IF C = D THEN GOTO 130
120     PRINT "NO !!!     ";A;" + ";B;" = "; C
125     GOTO 160
130     FOR X=46TO83:SET(X,27):SET(X,44):NEXTX
131     FOR Y=27TO44:SET(46,Y):SET(47,Y):SET(82,Y):SET(83,Y):NEXTY
132     FOR X=54TO59:SET(X,30):SET(X,33):NEXTX
133     FOR X=68TO73:SET(X,30):SET(X,33):NEXTX
134     FOR Y=31TO32:SET(54,Y):SET(55,Y):SET(58,Y):SET(59,Y):NEXTY
135     FOR Y=31TO32:SET(68,Y):SET(69,Y):SET(72,Y):SET(73,Y):NEXTY
136     FOR Y=34TO35:FOR X=62TO65:SET(X,Y):NEXTX:NEXTY
137     SET(52,37):SET(53,37):SET(74,37):SET(75,37)
138     SET(54,38):SET(55,38):SET(72,38):SET(73,38)
139     SET(56,39):SET(57,39):SET(70,39):SET(71,39)
140     SET(58,40):SET(59,40):SET(68,40):SET(69,40)
141     FOR X = 60 TO 67: SET(X,41):NEXTX
150     T = T+1
160     FOR I = 1 TO 2500: NEXT I
165     NEXT K
170     PRINT "YOUR SCORE IS "; T*10;"%"
180     END
```

Fig. 7.

The second program that we discuss is a depreciation schedule. In fact, I want to depreciate an asset in two ways—a straight-line depreciation and a sum-of-years digits depreciation. This program serves as a means of teaching a number of points. Since I am not an accountant, I pick up my favorite accountant text and look up the formula for straight-line and sum-of-years digits depreciation. They are: Let C be the cost of the asset, S the salvage value and N the useful life in years. Then, the straight line depreciation is: A = (C-S) / N and the sum of year digits depreciation is shown in Example 1. I = N for the first year; I = N-1 for the second year; . . . I = 1 for the *N*th year.

$$D = (2 \times I \times (C - S))/(N \times (N + 1))$$

Example 1.

The stage is now set for more instruction in looping techniques.

Finally, in order to introduce a need for the TAB function (which Level I BASIC supplies), I have the participants "dress up" their output with a heading and subheading (Fig. 5).

Since the audience is varied, the next problem is oriented to capture the attention of any real-estate agents that might be in attendance. If a young couple is interested in purchasing a home today, one of the first things that enters their minds is, "How much is the loan going to cost us per month?" What we have here is an amortization problem. Again, looking up the formula for the periodic payment, we see that if L is the amount of the loan, N is the number of monthly payments and R is the monthly interest rate, the payment P is shown in Example 2.

$$P = (L \times R \times (R + 1)^N)/((R + 1)^N - 1)$$

Example 2.

An immediate problem with Level I BASIC is that it does not have an exponentiation key. However, this serves as another teaching point—if you are clever, you can program around many of your system's shortcomings.

I hasten to point out, especially to the *ham radio operators*, that the trigonometry functions can also be programmed into

```
10      INPUT "HOW MANY DATA ELEMENTS ";N
20      FOR I = 1 TO N
30      INPUT "ENTER DATA ELEMENT "; A(I)
35      NEXT I
40      H = A(1)
50      L = A(1)
60      FOR I = 2 TO N
70      IF H < A(I) THEN H = A(I): GOTO 85
80      IF L > A(I) THEN L = A(I)
85      NEXT I
90      W = (H-L)/3
100     S=0:A=0:B=0:C=0
140     FOR I = 1 TO N
150     S = S+A(I)
160     IF A(I) < L+W THEN A = A+1: GOTO 210
180     IF (L+W <= A(I)) * (A(I) < L+2 * W) THEN B = B+1: GOTO 210
190     IF A(I) >= L+2 * W THEN C = C+1
210     NEXT I
215     CLS
220     PRINT "THE HIGHEST VALUE IS ";H
230     PRINT "THE LOWEST VALUE IS  ";L
240     PRINT "THE AVERAGE VALUE IS ";S/N
250     PRINT AT 256, L
260     PRINT AT 320, L+W
270     PRINT AT 384, L+2*W
280     PRINT AT 448, H
290     FOR I=1TOA:X=39+I:SET(X,12):SET(X,13):SET(X,14):NEXTI
300     FOR I=1TOB:X=39+I:SET(X,15):SET(X,16):SET(X,17):NEXTI
310     FOR I=1TOC:X=39+I:SET(X,18):SET(X,19):SET(X,20):NEXTI
320     END
```

Fig. 8.

```
10      CLS:PRINT "LOAD DATA TAPE -- PRESS PLAY AND RECORD"
20      FOR I = 1 TO 5000: NEXT I
30      CLS: PRINT TAB(20);"CREATING A MAILING LIST":GOSUB 1000
40      PRINT "ENTER CUSTOMER'S NAME XXXXXXXXXXXXXXXX"
41      INPUT "                          ";A$:PRINT
50      PRINT "ENTER CUSTOMER'S ADDRESS XXXXXXXXXXXXXXXX"
51      INPUT "                              ";B$
60      CLS:GOSUB 2000: PRINT# A$;",";B$:CLS
70      GOSUB 1000
80      PRINT "ENTER CITY XXXXXXXXXXXXXXXX"
81      INPUT "             ";A$:PRINT
90      PRINT "ENTER STATE CODE AND ZIP XX    XXXXX"
91      INPUT "                              ";B$
100     CLS: GOSUB 2000: PRINT# A$;",";B$:CLS
110     INPUT "ARE THERE MORE ENTRIES -- 1 - YES, 2 - NO ";Q
120     IF Q = 1 THEN GOTO 30
130     IF Q = 2 THEN END
140     PRINT "PLEASE ENTER EITHER 1 FOR YES OR 2 FOR NO"
150     GOTO 110
1000    PRINT:PRINT "AN ENTRY IS LIMITED TO 16 CHARACTERS"
1010    RETURN
2000    PRINT:PRINT "***** WRITING TO DATA TAPE *****"
2010    RETURN
```

Fig. 9.

the system. (Some of these routines are given in the back of the Level I BASIC manual.)

Although looping techniques were introduced from the beginning, I wait until the amortization problem to introduce the FOR NEXT instruction. The amortization program also receives the personal touch by the use of string variables in requesting the client's name and then, after the calculations are done, addressing the answer to the client. Since the interest rate does not vary daily (yet!), I can use the READ and DATA instructions to eliminate the need for the input of the rate each time the program is run (Fig. 6).

Probably the hardest lecture to present is that of the third night—the last night we consider the TRS-80 as a tool for decision making. My objective is to introduce the RND function, the logical operator* (AND) and the TRS-80 graphic functions CLS, SET, RESET, PRINT AT, as well as the idea of an array.

Since there are usually educators participating in these workshops, I can get their attention with a computer-assisted instruction program, which I will call an addition quiz. This program is designed for young children. It gives a ten-question problem set that a child can answer. If the child's response is correct, a "happy face" is drawn on the screen; if the response is incorrect, the correct answer is displayed and the child is allowed to study it for a few seconds. Although the concept of this program is basically simple, the tedium of the graphics seems to bore the participants. They like the results very much; however, they do not want to code the drawing (Fig. 7). The coding for the "happy face" is found inclusively in

```
10      CLS: PRINT "LOAD DATA TAPE INTO RECORDER -- PRESS PLAY"
20      FOR I = 1 TO 5000:NEXT I
30      CLS: PRINT TAB(17);"MAILING LIST RETRIEVAL"
40      PRINT: PRINT: INPUT# A$, B$
50      PRINT A$: PRINT B$
60      INPUT# A$,B$: PRINT A$;", ";B$
70      FOR I = 1 TO 5: PRINT: NEXT I
80      INPUT "ARE THERE MORE RECORDS TO BE RETRIEVED 1--YES, 2--NO";Q
90      IF Q = 1 THEN GOTO 30
95      IF Q = 2 THEN END
100     PRINT "PLEASE ENTER EITHER 1 FOR YES OR 2 FOR NO":GOTO 80
```

Fig. 10

lines 130 to 141.

On many occasions, we handle a large amount of data and we want to statistically analyze it. It is not necessary to give each piece of data a different name. Years ago, mathematicians had the same problem; they solved it by talking about sets and subscripted variables. If we have a set of three objects we can denote this by A = [a_1, a_2, a_3], where a_1 is the first element of the set, a_2 is the second element, and a_3 is the third element. We can do the same thing with our TRS-80 Level I BASIC unit.

Now we want to write a program to accept N data points; find the highest value, the lowest value and the average value; and draw a bar graph of the data grouped into three categories. This program illustrates a brute-force sorting technique. It again makes use of the graphic functions and introduces the instruction of PRINT AT and the logical operator* (AND) (Fig. 8).

The last two lectures show the TRS-80 Level I BASIC in the classical data-processing environment. The major function of data processing is the establishment of files of data, the retention of this data and the processing of it to produce meaningful information. Two facilities on the TRS-80 Level I BASIC—the tape recorder and the instructions INPUT# and PRINT#—allow us to enter the world of data processing.

The scenario for the next program is an eye-catcher. Suppose you are a Radio Shack store manager. You foresee that a customer who comes in and buys a $600 4K Level I BASIC unit will soon want to upgrade

```
10      CLS: PRINT "PLEASE ANSWER THE FOLLOWING QUESTIONS:":PRINT
20      INPUT "WHAT IS EMPLOYEE'S NAME ";A$ :PRINT
30      INPUT "WHAT IS THE SOCIAL SECURITY NUMBER XXX-XX-XXXX";B$:PRINT
40      INPUT "WHAT IS THE EMPLOYEE'S DATE OF BIRTH (MMDDYY) ";B:PRINT
50      INPUT "HOW MANY DEPENDENTS ";D
60      CLS
61      PRINT "SALARY TYPE: 1> SALARIED"
62      PRINT "             2> HOURLY":PRINT
70      INPUT "WHICH CATEGORY IS IT ";T
71      IF (T=1) + (T=2) THEN GOTO 80
72      GOSUB 200: GOTO 61
80      IF T = 1 THEN INPUT "WHAT IS WEEKLY SALARY ";W:GOTO 100
90      INPUT "WHAT IS HOURLY WAGE ";W
100     CLS
101     PRINT "SEX:      1> MALE"
102     PRINT "          2> FEMALE":PRINT
110     INPUT "WHICH CATEGORY IS IT ";S
111     IF (S=1) + (S=2) THEN GOTO 120
112     GOSUB 200: GOTO 101
120     CLS
121     PRINT "MARTIAL STATUS:   1> SINGLE"
122     PRINT "                  2> MARRIED"
123     PRINT "                  3> DIVORCED":PRINT
130     INPUT "WHICH CATEGORY IS IT ";M
131     IF (M=1) + (M=2) + (M=3) THEN GOTO 140
132     GOSUB 200: GOTO 121
140     CLS:PRINT "***** WRITING TO DATA TAPE *****"
141     PRINT# A$;",";B$;",";T;",";B;",";S;",";M;",";D;",";W
150     PRINT: INPUT "MORE EMPLOYEES 1--YES, 2--NO";Q
151     IF Q = 1 THEN GOTO 10
152     IF Q = 2 THEN GOTO 160
153     GOSUB 200:GOTO 150
160     T = 0
170     PRINT# A$;",";B$;",";T;",";B;",";S;",";M;",";D;",";W
171     FOR I = 1 TO 20
172     CLS: PRINT "REMOVE DATA TAPE"
173     FOR J = 1 TO 100: NEXT J
174     NEXT I
180     END
200     PRINT:PRINT "***** INVALID RESPONSE *****"
201     PRINT "PLEASE CHECK YOUR ANSWER":PRINT
202     RETURN
```

Fig. 11.

the system as he or she becomes more experienced. It would be beneficial to you, then, to keep a list of your TRS-80 customers and inform them periodically of product innovations, sales of equipment and changes or revisions.

What better to do with the TRS-80 you have on display in your shop than store a mailing list on tape? The information you wish to store is the individual's name, the street address, the city, the state code and the zip code.

Doing this brings a couple of immediate problems to mind. First, the TRS-80 Level I BASIC unit has only two string variables, A$ and B$. It cannot compare these to themselves or any other string. Second, these strings are limited to 16 characters each. Hence, our records will be 32 characters maximum, and we will make repeated use of A$ and B$. Furthermore, we must personalize this program by telling the user to load the new data tape and press play and record, by recalling each CTR-41 set at the position on which the program terminated the last time it was run. Finally, we want to be able to retrieve these records, so we write a quick little retrieval program incorporating some of the "personalizing" techniques we used in the creation program (Fig. 10).

The fifth and final formal lecture is devoted to the establishment of master files and the procurement of reports based on these files. These topics lead to some very interesting and fundamental concepts in file processing. I briefly touch upon the question of inserting, deleting, and updating records. The idea of a father-son tape system is explored with the realization that the TRS-80 4K Level I BASIC system will have to be upgraded. We leave this realm, a little dissatisfied but still capable of doing quite a bit of data manipulation. It is with these programs that I introduce the encoding and decoding of information for human readability, the editing of input in the creation program so that the data captured is "machine-pure" and using a dummy record to detect an end of file.

NAME	A$	Limited to 16 Characters
SOCIAL SECURITY NUMBER	B$	XXX-XX-XXXX
SALARY TYPE	T	1—SALARIED
		2--HOURLY
DATE OF BIRTH	B	MMDDYY
SEX	S	1--MALE
		2--FEMALE
MARITAL STATUS	M	1-SINGLE
		2--MARRIED
		3--DIVORCED
DEPENDENTS	D	
WAGE	W	

Table 1.

time that we are limited to 16 characters per entry, by marking out what 16 characters are such that we do not have to count the characters in our entry and by deciding when to quit.

This program (Fig. 9) serves to instruct the workshop participant in the use of subroutines in repetitive-type processing at different locations within a program. It also serves as a "springboard" for talking about blocking and the interblock gap between file records. Furthermore, additions to this file may be made by running this program again with the counter on the

My scenario for this set of problems is that I want to develop a personnel master file for all my employees. The information I want to store is the individual's name, social-security number, salary type, date of birth, sex, marital status, number of dependents and the weekly or hourly wage. The coding of the data is handled as in Table 1.

In order to create a master file with "*machine pure*" *data, then* edit checking must be performed at each point of data entry. The use of the logical operator + (OR) is very valuable at this time. The dummy record will have a zero in the T variable (Fig. 11).

Once the master file is created on tape, I want to create my detail file for the video display unit. The scenario for this problem is that I want a list of all salaried employees, their social-security number and their marital status. Since all my records are of the form A$, B$, T, B, S, M, D, W, I must interrogate T and M and print out A$, B$ and the correctly interpreted marital status (Fig. 12).

Although I have not said or written anything about program documentation in this paper, I do stress this to the participants. Since the classes are scheduled far enough apart for thinking about a problem, they are also set far enough apart for forgetting about how one solved a previous problem. The need for program documentation can easily be demonstrated throughout the seminar.

The final lecture is not really a lecture but an audience-participation discussion with some more advanced microcomputer systems that I have been able to borrow. I like to view the future . . . talk about the 11 megabyte hard-sector disks that are now being marketed by International Memories, Inc.[1] We discuss the chip and how it has revolutionized our environment, and I like to speculate on what bubble memory will do for microcomputer systems when it is offered commercially. (Rockwell International Corporation has now introduced bubble memory with volume production to start in 1980.[2]

Finally, in a very cumbersome yet functional way, I use a TRS-80 16K Level II BASIC system with line printer to give a demonstration of word processing. Each participant gets a letter with his or her name and address and proper salutation, thanking him for having attended the workshop and informing him of Loyola University's intention of continuing to offer higher-level seminars in the near future.

Evaluation

To date, this program has been very successful. I have detected the slow but sure movement away from the Level I BASIC units immediately; that is, beginners, because of the delays in shipping (possibly), are ordering the Level II units without ever having experienced the Level I unit. This leaves a gap in our educational structure because the opening pages of the Level II reference manual read: "We've prepared this reference manual with the assumption that you—the user—already have considerable experience with programming in BASIC. Our Level I user's manual was written for the total beginner—and has been greeted with wide acclaim. We freely admit this manual has not been written from the same perspective."[3] ■

```
10      CLS: PRINT TAB(22); "SALARIED EMPLOYEES":PRINT
11      PRINT "NAME",TAB(22);"SSN";TAB(38);"MARTIAL STATUS"
20      INPUT# A$,B$,T,B,S,M,D,W
30      IF T = 0 THEN GOTO 100
40      IF T = 2 THEN GOTO 20
50      IF M = 1 THEN GOTO 90
60      IF M = 2 THEN GOTO 80
70      PRINT A$;TAB(22);B$;TAB(41);"DIVORCED":GOTO 20
80      PRINT A$;TAB(22);B$;TAB(41);"MARRIED":GOTO 20
90      PRINT A$;TAB(22);B$;TAB(42);"SINGLE":GOTO 20
100     END
```

Fig. 12.

1. "Hardware," *Datamation*, June 1978, page 254.
2. Alexander Auerbauch, "Recent Advance Making Computers Practical," *The Times-Picayune*, New Orleans, Sunday, September 24, 1978, Section 1, page 32.
3. *Level II BASIC Reference Manual*, Radio Shack, 1978, Fort Worth.

A look at some interesting (and some not so interesting) programs.

Software Review

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

Evaluating software is a lot like evaluating television programs. My wife and I seldom turn on the TV anymore because there isn't much there that interests us, and yet millions of people sit staring at their TVs night after night. Does that mean that their tastes are better than ours or vice versa?

I don't think either alternative is necessarily true. It's just that our tastes are different. The same can be said about personal computer software. I really like some of the programs described below, while others really don't do anything for me at all. I hope that my descriptions and opinions will give you enough information to allow you to make your own decisions.

Software

Name: Line Renumbering
Vendor: Software Associates, PO Box 2248, Springfield VA 22152
Price: $14.95
Purpose: To renumber BASIC program lines and to allow the loading of more than one program at a time (merging).
Documentation: Excellent
Loading: OK
Implementation: The tape contains 16K, 32K and 48K versions, which will all work with Level II or disk. This is a machine-language program that resides in high memory. It can be called from BASIC at any time by using the SYSTEM command. Three renumber commands are provided: N allows you to pick the starting program line number and the line increment. For instance, N 100 10 would renumber the program starting at 100 in increments of 10. Of course, all GOTOs and GOSUBs are appropriately renumbered also. M puts a partition at the end of the in-memory BASIC program to allow another program to be added to it. As many programs as you have room for can be loaded. R removes all partitions and merges all in-memory programs.
Suitability: This is the type of applications software that the personal computer owner needs. It is a shame that this was left out of Level II BASIC.

Name: Doctor and Fetch
Vendor: Omicron Software, PO Box 2547, Sepulveda CA 91343
Price: Doctor, $10; Fetch, $7
Purpose: To demonstrate artificial intelligence
Documentation: Good
Loading: Touchy
Implementation: Both programs are intended to demonstrate the field of artificial intelligence. Doctor is based on the program Eliza written by Joseph Weizenbaum. It takes the part of a psychotherapist and asks you questions about your problems. The theory behind the program is good, but this implementation gets lost quickly if the program doesn't receive exactly the answers that it is looking for.

Fetch allows you to make declarative statements, which are stored, and then you can ask questions relating to the stored information. For instance:

```
?CARTER IS A PRESIDENT.
OKAY.
? A PRESIDENT IS A LEADER.
OKAY.
? IS CARTER A LEADER?
YES.
```

Table 1. GSF subroutine summary.

GSF#	Page#	Arg#	Mode	Description
1	5			*Invert Graphic Video*
		1	Address	Graphic data to be inverted
		2	Integer	Number of bytes to be inverted.
			Return Value	0
2	15			*Read Tape Data Block*
		1	Address	Location where data to be placed.
		2	Integer	Maximum number of bytes to be read.
		3	Integer	Tape Block ID number.
			Return Value	−1 Tape block ID does not match.
				−2 Data read error.
				>0 Number of bytes read.
3	15			*Write Tape Data Block*
		1	Address	Location of data to be written.
		2	Integer	Number of bytes to be written.
		3	Integer	Tape Block ID number.
			Return Value	Address last byte written +1.
4	9			*Duplicate Memory Serially*
		1	Address	Locations of start of data.
		2	Integer	Number of bytes to be duplicated.
			Return Value	Address last byte duplicated.
5	5			*Scroll Screen Up*
		None		
			Return Value	0
6	5			*Scroll Screen Down*
		None		
			Return Value	0
7	5			*Scroll Screen Left*
		None		
			Return Value	0
8	5			*Scroll Screen Right*
		None		
			Return Value	0
9	9			*Duplicate Memory Incrementally by 64*
		1	Address	Location of start of data.
		2	Integer	Number of bytes to be duplicated.
			Return Value	Address last byte duplicated.
GSF#	**Page#**	**Arg#**	**Mode**	**Description**
10	13			*Compress Data*
		1	Address	Data to be compressed.
		2	Address	Where compressed data is to be placed.
		3	Integer	Number of bytes to be compressed
			Return Value	Number of bytes in compressed area.

? IS REAGEN A LEADER?
I DON'T KNOW.
Which only proves that even computers can get into arguments about politics.

Suitability: Both programs are interesting and also demonstrate some possible future computer applications. I do, however, feel that they are overpriced, and I think that most buyers will feel the same. Both programs on one tape for about $7 to $8 would be more realistic and would probably net the vendor more money in the long run.

Name: Generalized Subroutine Facility (GSF)
Vendor: RACET Computes, 702 Palmdale, Orange CA 92665
Price: $24.95
Purpose: To provide utility subroutines that can be called from BASIC.
Documentation: Excellent
Loading: OK
Implementation: GSF provides a number of machine-language utility subroutines that can be called from a BASIC program. These subroutines are loaded into protected high memory and are available to the calling programs at all times. Some of the utilities provided include: Display Screen Control, Draw Horizontal and Vertical Lines, Duplicate Memory, Move Data, Compress and Uncompress Data, Read and Write Tape Data and In-Core Sort. See Table 1.

Each routine is numbered and is called with the Level II USR statement. As an example, consider the "Scroll Screen Left" utility, which is USR(7). Each time that USR(7) is used, i.e., K = USR(7), whatever is on the screen will move one position to the left. Scroll right, up, down and reverse video are all included in this group. The Draw Line utilities ask for the starting point and the length of a line, which appear almost instantly. Outstanding graphics are possible with these routines, which operate many times faster than the normal BASIC POKE and SET statements.

I found the Compress and Uncompress Data and the Read and Write Tape Data routines particularly useful. Recently I wrote a BASIC program to allow me to draw pictures on the screen using the Level II graphics characters. It worked out just fine. Then I added a routine to SAVE and LOAD the pictures on cassette tape. I stored the 1024 picture elements in an array and output the array to tape 64 elements at a time. Total recording time was almost 5 minutes. I improved on this by converting the array to 16 strings and recording those. Time to convert and record was about 1 minute. This was before I received GSF, which can do the same thing in a few seconds!

The other utilities are too complex to explain rationally here in a few words, but the manual does an excellent job. Each subroutine is carefully explained with a sample listing of a BASIC program using that routine. What makes learning even easier is that all of the example BASIC programs are also recorded on the GSF tape with a menu to provide selection. This is also an effective display of the type of graphics that can be displayed. I wish that all software gave the buyer this kind of demonstration.

Suitability: I only found two minor areas of complaint: the tape label says that protected memory should start at 29950, and the manual says 30000. It works great at 30000. Also, no error checking is done as far as addresses are concerned. If you tell it to move into an area occupied by your program or data, it will do so. If you tell it to draw a line longer than can be contained on the screen, it will do that too, even if it destroys data at the same time.

GSF comes in 16K, 32K and 48K versions. If you buy the 16K version and later increase the size of your system, you can get a 32K or 48K version for $5. I personally feel that all three versions should be included on the original tape. I don't think that the vendor will make much at that $5 price because it will be an inconvenience to the buyer. In any case, I definitely recommend GSF as a good buy for anyone who is seriously writing his own programs.

11	13			*Uncompress Data*
		1	Address	Data to be uncompressed.
		2	Address	Where uncompressed data to be placed.
			Return Value	Number of bytes in uncompressed area.
12	7			*Draw Vertical Line*
		1	Integer	Row number for vertical line.
		2	Integer	Column number for vertical line.
		3	Integer	Length of vertical line.
			Return Value	0
13	7			*Draw Horizontal Line*
		1	Integer	Row number for horizontal line.
		2	Integer	Column number for vertical line.
		3	Integer	Length of vertical line.
			Return Value	0
14	11			*Move Data*
		1	Address	Location of data to be moved.
		2	Address	Location where data is to be moved.
		3	Integer	Number of bytes of data to be moved.
			Return Value	Address last byte + 1 (Arg#3 + Arg#1).
15				*Fetch GSF Argument*
		1	Integer	GSF argument # to be fetched.
			Return Value	Integer argument saved by GSF.
16				*Fetch Memory Word*
		1	Address	Address of memory location fetched.
			Return Value	Integer value at memory location.
17	17			*In-Core Sort—Multiple Variable Mode*
		1	Address	Pointer to sort key string.
		2	Integer	Start index for sort.
		3	Integer	End index for sort.
			Return Value	0 Sort completed successfully.
				1 Null Argument #1
				2 Missing variable.
				3 Array specified not found.
				4 Array found not single dimension.
				5 Array too small.
18	17			*In-Core Sort—Character String Mode*
		1	Address	Pointer to array to be sorted.
		2	Integer	Start index for sort.
		3	Integer	End index for sort.
		4	Address	Sort key parameter list.
			Return Value	0 Sort completed successfully.
				1 Argument #4 array not integer.
				2 Argument #4 array multi-dimension.
				3 No substrings specified.
				4 Substring location 0 specified.

Name: Music Composer/Editor
Vendor: PFDC Software, 784 Goucher Street, Gretna LA 70053
Price: $20
Purpose: To compose and play music through a radio placed next to the TRS-80 keyboard unit
Documentation: Very good
Loading: Touchy
Implementation: Once you get the hang of it, music is very easy to enter into memory. The basic requirements are octave, note length and note. A quarter note at middle C would be entered as 2 ON C. The octave number does not have to be repeated until you move to a new octave. The same goes for the note length. Rests can be inserted where required. If you can read music at all, you will be able to enter notes quickly. Daisy, along with a random 25 note piece, is provided as a part of the program.

Many other features, such as listing, tape saving and loading, editing and playing, are provided. Each song can be titled, the tempo changed, the key changed, portions repeated and much more . . . too much to try to explain here. After a song has been entered from the keyboard or tape, it can be played, changed, replayed, retitled, saved on tape or erased. A radio placed near the keyboard and tuned for the best sound transmits the song when played on the computer. I tune my radio to 700 kHz on the AM dial.

Suitability: The music that results from this program is quite good; it is obvious that a great deal of thought went into its creation. However, there are some drawbacks. Only one note can be played at a time. . .sort of like playing the piano with one finger. Whenever a song is not playing, great amounts of computer noise emerge from the radio. I find this annoying and have to reduce the volume, which I forget to turn back up when I'm ready to play a song. All in all, this is a nice novelty demonstration program; I'll let you decide whether it is worth the price.

Conclusion

I'm happy to say that every one of the tapes reviewed above came in a plastic box. That makes for better, cleaner storage. I have modified my CTR-41 tape recorder for automatic level control while reading tapes. Now most tapes, whether my own or commercially produced, load at 7 on the volume control. Whenever I receive a tape that won't load at 7, I try different volume control settings until I get a good load. I immediately CSAVE the program back on the tape. From then on it will load at level 7.

At the present time I have received 50 TRS-80 tapes for review. Of these, I have been unable to load eight no matter what I do. I have requested replacements and will try again.

Robert Elliott Purser says on the back cover of his excellent magazine, "List of TRS-80, PET and Apple computer cassettes," that, ". . .95 percent of the programs listed in this magazine should never have been offered for sale." I don't know if I would agree with the 95 percent part, but there is an awful lot of software being sold that are word-for-word copies of magazine or book programs. Some are poor and some are good but most are overpriced and could have been copied by the user at no cost. None of the software reviewed here falls into that category.

With the enormous amount of software that is being offered for the TRS-80, it is difficult to know what you are getting by reading the advertisements. I hope this and similar reviews make the selection process a little easier and more effective. ■

✓ Reader Service—see page 147

✓ Reader Service—see page 147

With an impact printer, Braille can be generated by your computer.

Braille

Alfred J. Bruey
201 S. Grinnell St.
Jackson MI 49203

This article describes a program to generate and print Braille on an impact printer that has been adjusted to make a heavy impression. To read the output, just turn the sheet over so the reversed impressions become properly oriented raised dots. This program will run on either the Level II TRS-80 or the PET with no modifications.

The Development of Braille

As most of you already know, Braille is a form of written communication for the blind. It consists of patterns of raised dots that can be felt with the fingertips. These dots represent letters, numbers and special characters.

In the early 19th century, French Army officer Charles Barbier devised a method of sending messages that could be read in the dark. The messages were coded with a series of raised dots and dashes. Barbier demonstrated his system to Louis Braille, a young Frenchman who had been blind since the age of three. Braille found Barbier's code too complicated for general use, so, at the age of 15, Braille started work on a system of his own.

Before he died at age 43 in 1852, Braille, who was a teacher of the blind, an organist and a cellist, had devised a system to represent mathematical and musical notation. He also devised a way for the blind to write in Braille by using a sliding template on the paper with a stylus to punch the dots into the paper.

The Braille Notation

Braille notation is based on a cell containing six dots. For ease of reference, the dots are numbered from 1 to 6 as follows:

1 . . 4
2 . . 5
3 . . 6

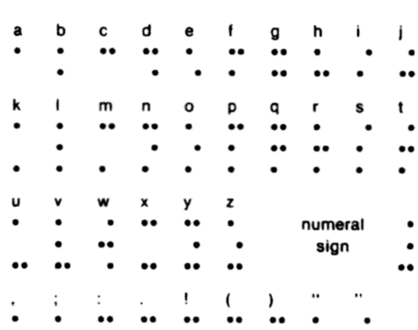


Fig. 1. The Braille alphabet.

As you can see from the first row of Fig. 1, the letters a through j are formed using only the dots 1, 2, 4 and 5. The letters on the second row, k through t, are formed by adding dot three to the characters above them in the first row. The third-row letters, u, v, x, y and z, are formed by adding dot 6 to the second-row characters k, l, m, n and o. The letter w is formed independently of the rules above. (The letter w is rarely used in French, and the symbol for w was not included in Braille's original alphabet.)

The digits 1, 2, 3, 4, 5, 6, 7, 8, 9 and 0 are formed by placing the numeral sign

in front of the pattern for the letters a, b, c, d, e, f, g, h, i or j, respectively. Similarly, capital letters are represented by placing a dot 6 before the letter symbol. Symbols for some punctuation marks are also shown in Fig. 1.

Printing Braille with a Computer

It should be obvious that Braille can be printed on a computer printer by using only the period symbol, but this would be of little use to the blind since they wouldn't be able to see the dots. A step in the right direction would be to have the printer adjusted to print a heavy impression of a dot so that it could be felt from the back of the paper. This method has a shortcoming: Turning the paper over to feel

Program listing.

```
5   DIM DA$(27,7),OT$(3)
10  DATA A,"•"," "," "," "," "," "
20  DATA B,"•","•"," "," "," "," "
30  DATA C,"•"," "," ","•"," "," "
40  DATA D,"•"," "," ","•","•"," "
50  DATA E,"•"," "," "," ","•"," "
60  DATA F,"•","•"," ","•"," "," "
70  DATA G,"•","•"," ","•","•"," "
80  DATA H,"•","•"," "," ","•"," "
90  DATA I," ","•"," ","•"," "," "
100 DATA J," ","•"," ","•","•"," "
110 DATA K,"•"," ","•"," "," "," "
120 DATA L,"•","•","•"," "," "," "
130 DATA M,"•"," ","•","•"," "," "
140 DATA N,"•"," ","•","•","•"," "
150 DATA O,"•"," ","•"," ","•"," "
160 DATA P,"•","•","•","•"," "," "
170 DATA Q,"•","•","•","•","•"," "
180 DATA R,"•","•","•"," ","•"," "
190 DATA S," ","•","•","•"," "," "
200 DATA T," ","•","•","•","•"," "
210 DATA U,"•"," ","•"," "," ","•"
220 DATA V,"•","•","•"," "," ","•"
230 DATA W," ","•"," ","•","•","•"
240 DATA X,"•"," ","•","•"," ","•"
250 DATA Y,"•"," ","•","•","•","•"
260 DATA Z,"•"," ","•"," ","•","•"
270 DATA" "," "," "," "," "," "," "
500 FORI = 1TO27:FORJ = 1TO7:READDA$(I,J):NEXTJ:NEXTI
```

```
544 OT$ = (1) = "  ":OT$(2) = "  ":OT$(3) = "  "
550 PRINT "ENTER CHARACTER STRING. NO MORE THAN 13 CHARACTERS
    :"
560 INPUT TX$
570 J = LEN(TX$)
580 IF J>13 THEN TX$ = LEFT$(TX$,13)
590 L = 1
600 FOR I = 1 TO 27
610 IF MID$(TX$,L,1) = DA$(I,1) THEN 630
620 NEXT I
625 PRINT "ILLEGAL CHARACTERS ":GOTO 544
630 OT$(1) = DA$(I,2) + OT$(1)
640 OT$(2) = DA$(I,3) + OT$(2)
650 OT$(3) = DA$(I,4) + OT$(3)
660 OT$(1) = DA$(I,5) + OT$(1)
670 OT$(2) = DA$(I,6) + OT$(2)
680 OT$(3) = DA$(I,7) + OT$(3)
690 OT$(1) = "  " + OT$(1)
700 OT$(2) = "  " + OT$(2)
710 OT$(3) = "  " + OT$(3)
720 L = L + 1
730 IFL>LEN(TX$)THEN 760
740 GOTO 600
760 PRINT OT$(1)
770 PRINT OT$(2)
780 PRINT OT$(3)
785 PRINT
790 PRINT"MORE? ENTER A Y OR N ";
800 INPUT AN$: IF AN$ = "Y" THEN 544
810 STOP
```

the raised dots would make the text backwards. We can correct this by *printing* the Braille backwards, so that when the paper is turned over, it will read correctly from left to right.

That's the method that this program uses. Braille text entered with INPUT statements is converted to reversed Braille and printed. An impact printer must be used for this application so that the pressure can be set to make tangible indentations in the paper. A character printer would probably give better results than a dot matrix printer, but you can try whatever you have.

The Program

This program is offered only as an example of what can be done. As written here, it is only capable of converting to Braille the 26 letters of the alphabet and the space character and printing them in reverse on the video screen. You can add more characters by using more DATA statements and changing the number 27 in lines 5, 500 and 600 to the new number of characters.

Since it takes seven characters to represent each character in the table, you will soon run up against the PET limit of 255 for a maximum subscript value. You can break the data up into two or more arrays and modify the search routine.

The 13s in lines 550 and 580 are necessary because of the screen width restriction on the PET. The numbers can be changed to 21 on the TRS-80 and to an even larger number if the Braille text is sent to a printer. Each Braille character requires three print positions, two for the character and one for the space between characters.

An example of backward Braille, the letters a through m, is shown in Fig. 2. If you wish to demonstrate this program printing regular, nonreversed Braille, just reverse the order of the operands on the right side of the equations in lines 630 to 710. For example, change line 630 to

```
OT$(1) = OT$(1) + DA$(1,2)
```

This program is simple to run. Just type RUN and enter the character string that you want to convert when you're prompted for it. TRS-80 users will have to use the CLEAR command to reserve some string space.

Extensions

There is a lot of room for experimentation with this method, and this program is only presented as a starting place. For example, it should be relatively easy to connect a printer and a typewriter in conjunction with a keyboard and a microcomputer so that a blind person could type a document for a class or company while at the same time generating a Braille copy for his own records.

This program could also be changed to print forward as described above, and the printed output can be used as input to a scanning device, which could create embossed output.

Although I agree that it's fun to hunt the Wumpus or land your simulated rocket on the moon, there's no reason that more useful things can't be done with microcomputers. I hope this program will help someone take a step in that direction. ■

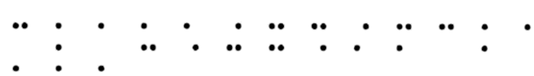


Fig. 2. The letters a through m printed in reverse.

Find out if you can read minds with this aid to ESP research.

Telepathy

John Warren
Assistant Professor
Dept. of English
East Carolina University
Greenville NC 27834

Extra sensory perception (ESP)—to most people these words conjure up images of crystal balls, black cats and bad science-fiction movies. However, many reputable psychologists feel that ESP is real and that a large portion of the population can call upon supernormal powers at one time or another. Two of these powers are telepathy and precognition.

Telepathy is the ability to read the thoughts of others... sometimes even the ability to see through their eyes. Precognition involves a knowledge of things to come. This foreknowledge may extend for seconds, weeks, months or years.

Nearly everyone can claim a "psychic experience." For example, the man who inexplicably stopped just seconds before a brick crashed into the sidewalk in front of him, or the woman who "just knew" what the man in the elevator was thinking. Unfortunately, scientists can't use these occurrences as evidence. The human subconscious is too effective in sorting out and acting upon cues that the conscious mind just slides over.

To build a body of evidence to study, scientists had to come up with a series of stimuli that were simple, incapable of being easily misunderstood, easy to handle and inexpensive. Among the first instruments tested were ordinary playing cards, but the shapes were ambiguous and there was too much confusing information—such as suit, number and color. Instead of abandoning cards entirely, the scientists produced their own designs. These had only four distinct symbols (a fifth was added later)—circle, cross, square and star. These are known as Zener cards and are accepted tools for detecting certain kinds of ESP powers.

The Program

This program simulates a deck of Zener cards but has the advantage that it acts as an electronic bookkeeper. At the end of the run, the readout contains not only a raw score but a record of each trial.

Another person is needed for the telepathy segment. The subject should turn so that he can see neither the tester nor the computer screen. Some psychologists suggest that the subject close his eyes and try to make his mind as blank as possible.

When the image appears on the screen, the tester should look directly at it, concentrate on the shape for about ten seconds and ask the subject what form he thinks is on the screen.

To minimize verbal cues, the question should be the same each time. A single word or sound is sufficient to prompt the subject. After getting a response, the tester enters it into the computer, which goes automatically to the next image.

In the precognition segment, the computer acts as the tester. The subject sits at the terminal and enters what he feels that the image will be. This is a valid test of prediction since the random number generator is not activated until a response has been entered.

At the beginning of the run, the user can elect to have the computer terminate the run after a given number of trials, or by entering 0 when prompted, he can terminate the run at any time with an escape command instead of the requested input.

At present, the DIM statement on line 100 limits the program to 100 trials. If a greater number of trials is desired, simply increase the number after A$ *and* B$.

Interpreting Data

Pure chance will yield a score of about 25 percent. A consistently higher or *lower* score could be a strong indication of active ESP ability. In statistical circles, an abnormally low score is considered just as significant as an abnormally high one since both are deviations from the

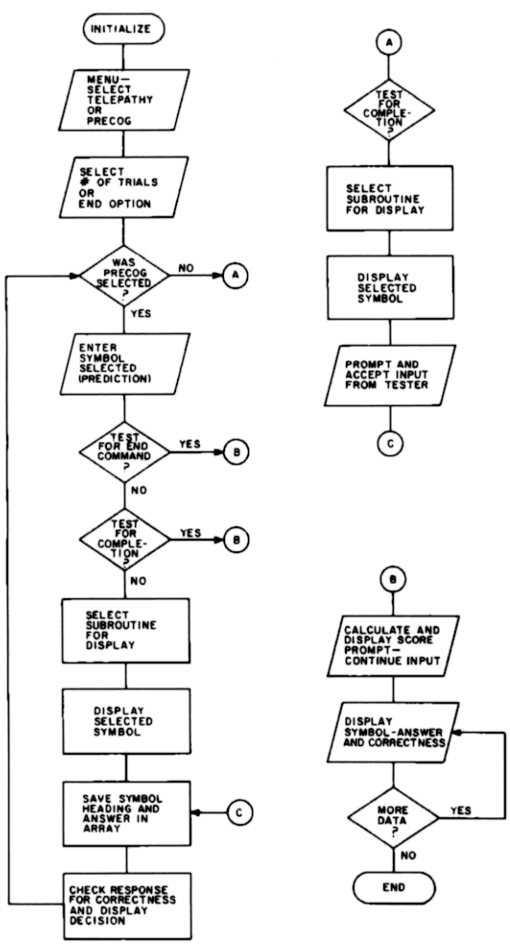


Fig. 1. Program flowchart.

ESP program.

```
100 CLEAR700: CLS: CT=0: MI=0: DIMB$(100), A$(100)
120 PRINTCHR$(23): PRINT@16, "E S P  TEST": PRINT@256, "PRECOGNITION ------------ 1": PRINT: PRINT"TELEPATHY --------------- 2"
140 INPUTA
150 PRINT"TO LIMIT RUN TO A SELECTED      NUMBER OF TRIALS   ENTER THAT     NUMBER"
160 INPUT"TO USE 'END'  ENTER '0'"; TR
170 IFA=2THEN290
180 CLS: PRINT@448, "TYPE WHAT THE SYMBOL WILL BE (SQUARE, CIRCLE, CROSS OR STAR)"
190 IFTR>0GOSUB3000
200 INPUTA$
210 IFA$="END"THEN2000
290 IFTR>0GOSUB3000
300 CLS: CT=CT+1: B=RND(4): ONBGOSUB10100, 10200, 10300, 10400
320 RESTORE: IFA=1THEN FORN=1TO1000: NEXT: GOTO1000
340 PRINT@0, "ASK THE SUBJECT WHAT YOU ARE SEEING AND TYPE HIS RESPONSE": INPUT"(CIRCLE, SQUARE, CROSS OR STAR)"; A$
350 IFA$="END"CT=CT-1: GOTO2000
360 GOSUB1000: GOTO290
1000 A$(CT)=A$: B$(CT)=B$: CLS: IFA$=B$PRINT"CORRECT RESPONSE""
1010 IFA$<>B$ MI=MI+1: PRINT"INCORRECT RESPONSE"
1050 FORN=1TO500: NEXT: IFA=1THEN180
1060 RETURN
2000 CLS: PRINT"SCORE="; 100-(MI/CT*100); "PERCENT   PRESS ENTER TO REVIEW ANSWERS"
2005 INPUTZ$
2010 FORX=1TOCT: PRINT"TRIAL#"; X, : PRINT"SYMBOL="; B$(X), "RESPONSE="; A$(X);
2020 IFA$(X)=B$(X)PRINT"     CORRECT": GOTO2040
2030 PRINT" "
2040 NEXT
2999 END
3000 IFCT=TRTHEN2000
3010 RETURN
9999 END
10000 REM SUB FOR IMAGES
10100 B$="STAR"
10110 Y=12: FORX=60TO90: Y=Y+1: SET(X, Y): NEXT
10120 Y=43: FORX=30TO59: Y=Y-1: SET(X, Y): NEXT
10130 Y=43: FORX=30TO100STEP4: Y=Y-1: SET(X, Y): NEXT
10140 Y=43: FORX=86TO18STEP-4: Y=Y-1: SET(X, Y): NEXT
10150 Y=24: FORX=18TO98: SET(X, Y): NEXT
10160 RETURN
10200 B$="CROSS": Y=19: FORX=45TO75: SET(X, Y): NEXT
10201 X=60: FORY=10TO45: SET(X, Y): SET(X+1, Y): NEXT
10210 RETURN
10300 B$="CIRCLE": CLS
10310 READ X, Y: IFX=0ORY=0THEN RETURN
10320 SET(X, Y): SET(127-X, 47-Y): SET(127-X, Y): SET(X, 47-Y)
10325 GOTO10310
10330 DATA 63, 7, 62, 7, 61, 7, 60, 7, 59, 7, 58, 7, 57, 8, 56, 8, 55, 8, 54, 8, 53, 8, 52, 8, 51, 8, 50, 8
10340 DATA 49, 8, 48, 9, 47, 9, 46, 9, 45, 9, 44, 10, 43, 10, 42, 10, 41, 10
10350 DATA 40, 11, 39, 11, 38, 12, 37, 12, 36, 13, 35, 13, 34, 14, 33, 14
10360 DATA 32, 15, 31, 16, 30, 17, 29, 18, 28, 19, 28, 20, 27, 21, 27, 21
10370 DATA 27, 22, 27, 23, 27, 24, 0, 0, 0, 0
10400 B$="SQUARE"
10410 X=30: FORY=15TO40: SET(X, Y): SET(X+60, Y): NEXT
10420 Y=40: FORX=30TO90: SET(X, Y): SET(X, Y-25): NEXT
10430 RETURN
```

predicted norm.

Just what the practical benefit would be from a negative ESP seems rather vague. However, these people would do well to avoid gambling or "playing hunches."

The user should carefully analyze the incorrect answers on the readout. Studies have indicated that some persons with what is presumed to be strong ESP potential will set up a pattern of wrong answers. For example, every time a cross was displayed, a particular subject would respond "star." His response to a star would consistently be "circle."

This is just as valid an indication of ability as being able to pick the correct answer; although, again, the practical usefulness is questionable.

This program is presented for amusement only. It is a great way to introduce people to the TRS-80 and is a wonderful icebreaker at parties. However, strict laboratory conditions are required to remove the intervening variables that interfere with scientific testing. It is highly unlikely that home conditions would be completely suitable. ■

Your 80 can help you through the decision making jungle—if you can decide to use the program!

Decisions, Decisions

Stephen Walton
PO Box 147
Village Station
New York NY 10014

If making decisions were easy, there would be no mystique surrounding the process. As it is, few choices are really clear-cut, and we need all the help we can get.

I'm an admirer of Phil Feldman and Tom Rugg's utility-value decision program ("Pass the Buck," p. 90, *Kilobaud* No. 7, July 1977) and have enjoyed running it for both serious and trivial problems. Like this program, it forces you to make your decision a little bit at a time.

But where Feldman and Rugg had you identify and weigh the factors relevant to a decision then evaluate each alternative in the light of each factor, my program, Cross-Preferences, simply asks you to decide between alternatives two at a time, until all pairs have been tested and the program can then rank alternatives by frequency of preference.

Concept

Three arrays are used: B records the number of times each alternative is preferred as

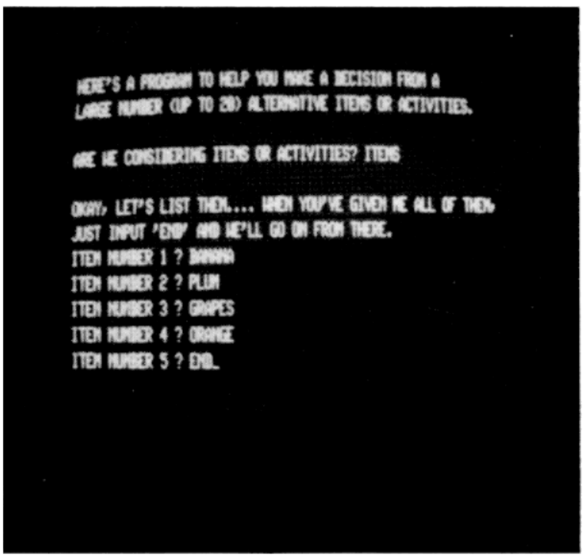


Photo 1: Beginning of a run.

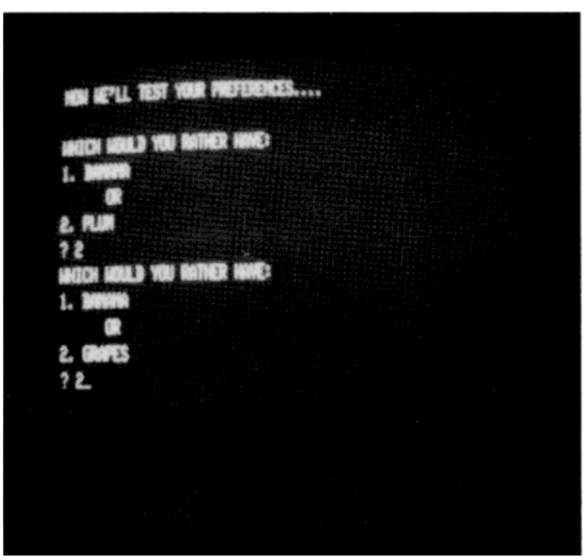


Photo 2: Testing user preferences.

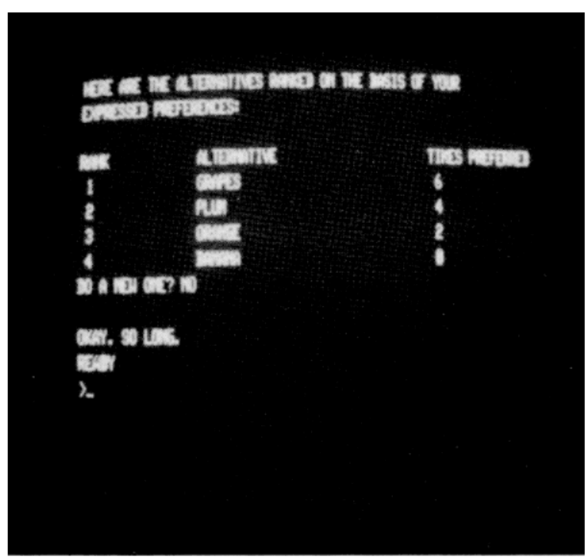


Photo 3: A decision is made

they're tested in pairs. Two-dimensional P contains, after ranking, the numerical identities of the alternatives and the number of times each was chosen over another. A$ holds alternatives' names.

The number of alternatives is limited to 20 to keep video display of the list manageable; the limit could be higher or indefinite. After the alternatives have been identified as either "items" or "activities" (to make later program responses a bit smoother) and they've been listed and the list approved, the pairs are tested.

With a list of seven or fewer alternatives, each one is tested against each other one twice, the second time with their order reversed. The double testing provides a kind of verification: if one alternative is preferred the first time and the other the second, the results of testing them against each other will cancel out in the final rankings.

For as many as seven alternatives A, the number of pair tests (microdecisions?) is $A^2 - A$, for a maximum of 42 tests. With eight to 20 alternatives, each pair is tested only once, and the number of tests is $(A^2 - A)/2$. That's still a lot of microdecisions to make on a 20-alternative list, but not as unwieldy as 380 would be.

The decision to make seven the breakpoint for double-vs-single testing was entirely arbitrary. Change line 605 in the program listing to go single above a different number or just eliminate line 605 for double testing of any size list.

When tests are complete, each alternative will have been preferred anywhere from zero to $2A - 1$ times, and it's this value that's stored in array B at that alternative's number. The program now does a fast "destructive sort" of B to load P with those values, largest first, and the ranked alternatives' numerical identities.

The contents of P are then printed in order, along with the alternatives' names obtained from A$. A sample run using a trivial example—"What kind of fruit do you want for dessert?"—is shown.

Implementation

The program was written in Radio Shack Level II BASIC and requires 2302 bytes, including 500 cleared for strings. Conversion to another string-handling BASIC should present no difficulty.

On the TRS-80 (and probably on most other machines as well), it's good to be terse in naming each alternative—or the printing of the ranked list may be thrown off. Try to keep each alternative to two or three words and you'll print cleanly at the end.

Use and Improvements

Like the Feldman and Rugg program—and like old-fashioned coin-tossing—Cross-Preferences can make you look at what's already on your mind when you're faced with a decision. Even if you disagree with the program's conclusions, you'll probably find that they help you clarify your thoughts.

Most of us have been exposed, at one time or another,

```
10 REM *** CROSS-PREFERENCES ***
20 REM  A DECISION-AID PROGRAM BY STEPHEN WALTON
30 REM
40 REM
80 CLEAR 500
90 CLS
100 DIM A$(20),B(20)
110 PRINT"HERE'S A PROGRAM TO HELP YOU MAKE A DECISION FROM A"
120 PRINT"LARGE NUMBER (UP TO 20) ALTERNATIVE ITEMS OR ACTIVITIES."
130 PRINT
140 INPUT"ARE WE CONSIDERING ITEMS OR ACTIVITIES";T$
150 R$=LEFT$(T$,1)
160 IF R$="I" V$="HAVE"
170 IF R$="A" V$="DO"
180 IF R$<>"I" AND R$<>"A" THEN 140
190 PRINT:PRINT"OKAY, LET'S LIST THEM....  WHEN YOU'VE GIVEN ME ALL OF THEM,"
200 PRINT"JUST INPUT 'END' AND WE'LL GO ON FROM THERE."
210 IF R$="I" J$="ITEM"
220 IF R$="A" J$="ACTIVITY"
230 C=1
240 PRINT J$;" NUMBER";C;
250 INPUT Q$
260 IF Q$="END" THEN 508
270 A$(C)=Q$
280 C=C+1
285 IF C=21 PRINT"SORRY, 20 IS THE LIMIT.":GOTO 490
290 GOTO 240
490 INPUT"PRESS 'ENTER' TO CONTINUE";Z$
500 CLS:PRINT"HERE'S YOUR LIST:"
505 C=C-1
510 FOR I=1 TO C
520 PRINT"     ";A$(I)
530 IF I=12 AND C>12 PRINT"THERE'S MORE...  PRESS ENTER TO SEE IT";:INPUT Z$
540 NEXT I
550 INPUT"IS THE LIST CORRECT";Z$
560 IF LEFT$(Z$,1)="N" PRINT"WE'LL DO IT OVER.":GOTO230
570 IF LEFT$(Z$,1)<>"Y" THEN 550
600 CLS:PRINT"NOW WE'LL TEST YOUR PREFERENCES....":PRINT
605 IF C>7 THEN 624
610 FOR I=1 TO C
620 FOR J=1 TO C
622 GOTO 630
624 FOR I=2 TO C
626 FOR J=1 TO I-1
630 IF J=I THEN 720
640 PRINT"WHICH WOULD YOU RATHER ";V$;":"
650 PRINT"1.  ";A$(I)
660 PRINT"       OR"
670 PRINT"2.  ";A$(J)
680 INPUT R
690 IF R=1 B(I)=B(I)+1
700 IF R=2 B(J)=B(J)+1
710 IF R<>1 AND R<>2 THEN 640
720 NEXT J,I
800 DIM P(C+1,2),P$(C+1)
810 FOR N=1 TO C:B(N)=B(N)+1:NEXT N
820 FOR K=1 TO C
830 L=0:V=0
840   FOR N=1 TO C
850  IF B(N)>V THEN L=N:V=B(N)
860   NEXT N
870 P(K,1)=L:P(K,2)=V
880 B(L)=0
890 NEXT K
892 CLS:PRINT"HERE ARE THE ALTERNATIVES RANKED ON THE BASIS OF YOUR"
894 PRINT"EXPRESSED PREFERENCES:":PRINT
900 PRINT"RANK","ALTERNATIVE",,"TIMES PREFERRED"
910 FOR N=1 TO C
915 IFN=12 AND C>12 PRINT"PRESS 'ENTER' TO SEE THE REST";:INPUTZ$
920 PRINTN,A$(P(N,1)),,P(N,2)-1
930 NEXT N
960 INPUT"DO A NEW ONE";Z$:IF LEFT$(Z$,1)="Y" THEN RUN
970 PRINT:PRINT"OKAY. SO LONG."
980 END
```

Program listing.

to vocational-preference tests, which work the same way as this program. You can readily construct your own "What do I want to be when I grow up?" test by giving appropriate inputs for the list of alternatives.

The principle can be extended to cheap-and-dirty personality tests such as the WQFOT (Walton's Quick Fundamental-Objectives Test). In this, the alternatives, specified as items, are: power, fame, wealth, love and pleasure. I've tried the WQFOT on several persons and obtained widely varying results.

With a graphics-capable micro such as the TRS-80, you can go on to graph a profile showing the importance of each of these aims for the testee and even comparing such a profile with an updated "normal" profile (using cumulative averages for all the people you've tested), if you permanently embed the WQFOT list—or your own—in a version of the program. (Some suggest that "accomplishment" should be added to the WQFOT list, while others maintain that accomplishment is implicit in the attainment of any of the objectives already listed. You decide.)

One improvement I'll definitely include in the next version of Cross-Preferences is a procedure for recognizing ties and identifying them as such in the print of the ranked list. You could also arrange for recognition of preferences as strong or of the don't-much-care variety, with numerical or verbal indications given in the last print. Extension to alternatives-lists of any length was mentioned above. Provision for changing an incorrect list without completely redoing it would also be nice.

A combination of this program with Feldman and Rugg's is another possibility. You would probably take the Cross-Preferences data first and display it. Then, while running the utility-value module, you would ask what weight should be assigned to relative preference as a factor (the user can always eliminate it by assigning it zero weight), positioned last on the list of factors. It would be handled differently from the other factors, inasmuch as the user would make no new evaluation of the alternatives based on it—rather, the data already obtained would be used, with the weighting previously assigned to relative preference as a factor, to contribute to the conclusions drawn.

This program is available on cassette for TRS-80 from the author for $4.95 postpaid. ■

✓ *Reader Service—see page 147*

Teach yourself the ins and outs of Level I with the aid of these program modules.

Beyond Blackjack

Robert M. Thorson
12991 Pierce Road
Saratoga CA 95070

Are you one of the thousands who have recently purchased a TRS-80 from Radio Shack? By now you've probably played a lot of blackjack using the packaged cassette program that came with your TRS-80, and perhaps now you're trying to teach yourself Level I BASIC with Radio Shack's excellent *User's Manual*.

Short Graphics Programs

Here are some tips that I use in my high school electronics classes to speed up the learning of BASIC and to hold the interest of the students at the same time. I have found that *short* graphics programs are effective teaching tools because the results of changes in the programs are immediately visible. The programs themselves are interesting enough to challenge the learner to take an active role in trying these different changes and to think up variations of his own.

After the student has learned the main instructions, statements and commands from the *User's Manual*, we introduce the random number function RND(***), which will generate random whole numbers between 0 and the number inside the brackets, and the SET(X,Y) statement, which will light the video screen at a spot determined by the horizontal and vertical coordinates X and Y. To illustrate this we use the short program shown in Example 1.

Line 20 generates random X and Y coordinates and lights a spot on the screen at that point. Line 30 causes this action to be repeated continuously. Results? A screen that is rapidly filling with stars.

Now let's take the program in Example 1 and expand it a little. For this we'll need to use a couple more graphic statements. They are RESET(X,Y), which will darken the video screen at the point whose coordinates are (X,Y), and POINT(X,Y), which checks graphic location (X,Y) . . . if the point is lit, a 1 is returned; if the point is off, a 0 is returned. PRINT AT begins printing at a specified location on the display.

Now let's see if you can make some changes. The program now causes the spot to reflect off the edges at the same angle it entered. Can you make it reflect at a different angle? Don't look at Hint 1 until you've tried. How about changing the size of the spot (Hint 2)? Could you confine the pattern to a small portion of the screen (Hint 3)?

As the program is written now, the spot keeps going until you hit the BREAK key. How could you change the program so it would stop by itself? We'll give you a little tip: Set up a counter and use an IF-THEN statment. If you can't do it, see Hint 4.

Our last program drew a symmetrical pattern. We asked you to change the program so the spot would be a different size and reflect at a different angle (Hints 1 and 2). Now can you write a program that would run through this revised program and then go back and run the original program? (See Hint 5.) Next let's change things so the screen will go blank for a while between the two programs you just wrote and so the programs will repeat. (See Hint 6.)

Now we'll write a program that will also fill the screen with stars, only this time it will do so by checking points at random on the screen (line 40) to see if the point is lit (1) or dark (0). If the point is lit, we'll turn it off (line 50); if the point is dark, we'll turn it on (line 90).

To keep track of how many points are lit up we will set up a counter (line 10), and every time we turn on a point we'll add 1 to the counter (line 100) and when we RESET we'll subtract 1 (line 60). Line 70 and 110 will keep printing out the total in our counter and, as an added bonus, will show an asterisk whenever a point is being turned on.

Will this program fill the screen completely with dots? No, it will reach an equilibrium point when the screen is about half full (see Example 2).

Now change the program so

```
10  CLS
20  SET(RND(120), RND(40))
30  GOTO 20
```

Example 1.

```
10  CLS: A = 0
20  X = RND(120)
30  Y = RND(40)
40  IF POINT(X,Y) = 0 THEN 90
50  RESET(X,Y)
60  A = A - 1
70  PRINT AT 897, "POPULATION = "; A
80  GOTO 20
90  SET(X,Y)
100  A = A + 1
110  PRINT AT 897, "*POPULATION = "; A
120  GOTO 20
```

Example 2.

```
5 CLS
10  X = RND(59): Y = RND(40)
20  A = 1: B = 1
30  X = X + A: Y = Y + B
40 IF X = 59 THEN A = -1
50 IF X = 1 THEN A = 1
60 IF Y = 40 THEN B = -1
70 IF Y = 1 THEN B = 1
80 SET(2*X,Y):SET(2*X + 1,Y)
90 GOTO 30
```

Example 3.

```
10  CLS
20  X = RND(120): Y = RND(40)
30  A = RND(6): B = RND(3)
40  X = X + A: Y = Y + B
50  IF X<7 THEN A = RND(6)
60  IF X>113 THEN A = -RND(6)
70  IF Y<4 THEN B = RND(3)
80  IF Y>36 THEN B = -RND(3)
90  SET(X Y)
100  GOTO 40
```

Example 4.

```
10  CLS
20  X = 32: Y = 24
30  SET(2*X, Y + 2): SET(2*X + 1, Y + 2)
40  A = 1: B = 1
50  FOR I = 1 TO 20
60  FOR J = 1 TO I
70  X = X + A: Y = Y + B
80  SET(2*X, Y + 2): SET(2*X + 1, Y + 2)
90  NEXT J
100  A = -A
110  FOR J = 1 TO I
120  X = X + A: Y = Y + B
130  SET(2*X, Y + 2): SET(2*X + 1, Y + 2)
140  NEXT J
150  B = -B
160  NEXT I
```

Example 5.

dots will appear only on the left half of the screen (Hint: Change line 20.)

Now let's look at the next program (Example 3). See if you can figure out what is supposed to happen here? Once again we start out at a random spot on the screen (line 10). Line 20 adds 1 to both the horizontal and the vertical coordinates, and this spot is turned on at line 80. You'll notice *two* SET statements in line 80. This was done to give us a nice square spot. Leave the second SET statement off and you'll see the difference.

Line 90 sends us back to do it all over again, but once again a 1 is added (line 30) to the horizontal and vertical coordinates, shifting the spot in a diagonal direction. Lines 40 through 70 form fences that cause the spot to bounce off the edges. We end up with a line moving diagonally around the screen generating a geometric pattern.

Testing Your Programming Ability

Let's see if you can write a program that will cause the spot to draw a line around the screen as in the previous program, but this time we want the spot to move at random speeds and reflect off the sides at random angles, and, while we're at it, let's make the sides vary randomly, too. *After* you've written your program, compare it with the program in Example 4. Notice that with A and B generated at random, the distance the spot moves with each step will also be random. Lines 50 through 80 cause the sides to vary. Try changing the numbers inside the RND brackets and see what effect that has on the program.

Your final test is to write a program that will draw a spiral that forms a large diamond on the screen. We'll give you one hint: You'll need two FOR-NEXT loops. Compare you results with the program in Example 5.

As a final task, see if you can write a program that will cycle through *all* these programs with blank pauses between each part. Incidentally, a good attention-getter we have found at school is to put the video monitor in the display window and have the TRS-80 cycle through these graphic programs with a little propaganda for the electronics classes appearing on the screen between graphic displays.

If you have been able to run all these programs, you should be well on your way toward writing your own programs—and that blackjack cassette will just gather dust on your desk. ■

Change (for example) line 40 to
40 IF X = 59 THEN A = -2

Hint 1.

Change line 80 to
80 SET(2*X,Y):SET(2*X + 1,Y):SET(2*X + 2,Y)

Hint 2.

Change the value of X and Y
in lines 40 through 70

Hint 3.

Add these lines:
7 N = 1
31 N = N + 1
82 IF N<>700 THEN 90
85 END
This will cause our spot to keep going until
our counter hits 700

Hint 4.

Use the lines added in Hint 4, except change
line 85 to
85 CLS and add
89 GOTO 105
Then add 100 to each line of the original program.
For example, line 40 becomes line 140.
Make line 190 GOTO 130.

Hint 5.

Add these lines to the Hint 5 program:
87 FOR T = 1 to 1000
88 NEXT T
89 GOTO 105
90 GOTO 30
Also add 100 to each of the above lines and
add them to the program, except modify lines
182, 189 and 190 to:
182 IF N <> 700 THEN 190
189 GOTO 5
190 GOTO 130

Hint 6.

Why your cassette is so bad, and how you can improve it. Construction details next month.

Cassette Problems

Donald L. Stoner
Dick Barker
The Peripheral People
Box 524
Mercer Island WA 98040

Bill Jones wanted a copy of John Smith's master tape of TRS-80 programs. John connected a couple of cassette recorders together, ran off a copy of his tape and checked it on his "80." A spot check of the programs showed they loaded properly.

A grateful Bill Jones took the tape home and stuffed it into his CTR-41 and mashed the play button. Oops . . . the first program didn't load! Neither did the second! After much fiddling with the volume control and tone switch, Bill succeeded in getting one of the programs to load. Unfortunately, a couple of bits were dropped, and an error message was generated at line 200. Some program tape! Some friend, that John Smith!

Sound familiar? With a couple of "ifs," the Radio Shack cassette/data-storage method is a highly reliable system . . . if the CTR-41 is used and if you get the volume set just right.

Having problems with a "freebie" tape is not as frustrating as having difficulty loading a program you paid hard cash for! After waiting some time for your tape to arrive, you put it in the cassette player and try to load it, but READY never appears on the screen.

Introduction

Most commercial enterprises duplicate tapes at high speed (five to ten times the normal playing speed). It takes superb equipment and professional-quality tape to get acceptable copies. A tiny flaw in the tape oxide (that might not affect a CSAVEd tape) can ruin a high-speed duplicated tape. Even at normal speed (500 baud for Level II), the actual data pulse is only .00054" wide. At high-speed duplication, the pulse is proportionately narrower. It doesn't take much of a flaw to destroy something so tiny. Even a dust particle can cause one or more missing data bits! Reputable program suppliers, such as Instant Software, are well aware of the duplicating problems and use the finest equipment and tape.

System tapes and other programs in assembly language present another type of duplicating problem. Generally speaking, they are harder to load than programs in BASIC. More important, they are not easy to duplicate by the average TRS-80 owner because they cannot be CSAVEd without T-BUG or similar aids. Finding their location in memory is not always easy, even for the experienced TRS-80 buff.

The circuit called the Data Dubber will permit 100 percent perfect CLOADs even with tapes that contain waveform distortion, noise, hum and even *minor* dropouts. When connected between two tape recorders, the circuit allows the operator to make perfect copies of any tape, even those in assembly language!

To prove the point, during testing of the design, we made copies of copies of a chess game in assembly language that was notoriously difficult to load. By the time a third-generation copy was made, pulse jitter from speed variations became a problem. However, the third-generation copy loaded with a little coaxing. Even three-for-a-dollar tapes from the local discount house produced consistently good copies when the Data Dubber was used.

Before we tell you how to perform these miracles of science, let's see why a Data Dubber is even necessary.

The Data System

The data stream produced by the TRS-80 is shown in Fig. 1. The stream starts with a 4-second leader of zeros to synchronize everything and ready the TRS-80 for a CLOAD. The leader consists of sync pulses only. These and the following sync pulses occur every 2 milliseconds (Level II).

After the leader, a data bit may occur between sync pulses, depending on whether a 1 or a 0 is being transmitted (Fig. 2a). This system is relatively forgiving of variations in tape speed. Even if the recorder speeds or slows slightly, the next pulse is still the next pulse. This is important since it is virtually impossible to make an inexpensive tape recorder that will maintain a constant tape speed and baud rate.

However, this scheme does have one disadvantage. If a sin-

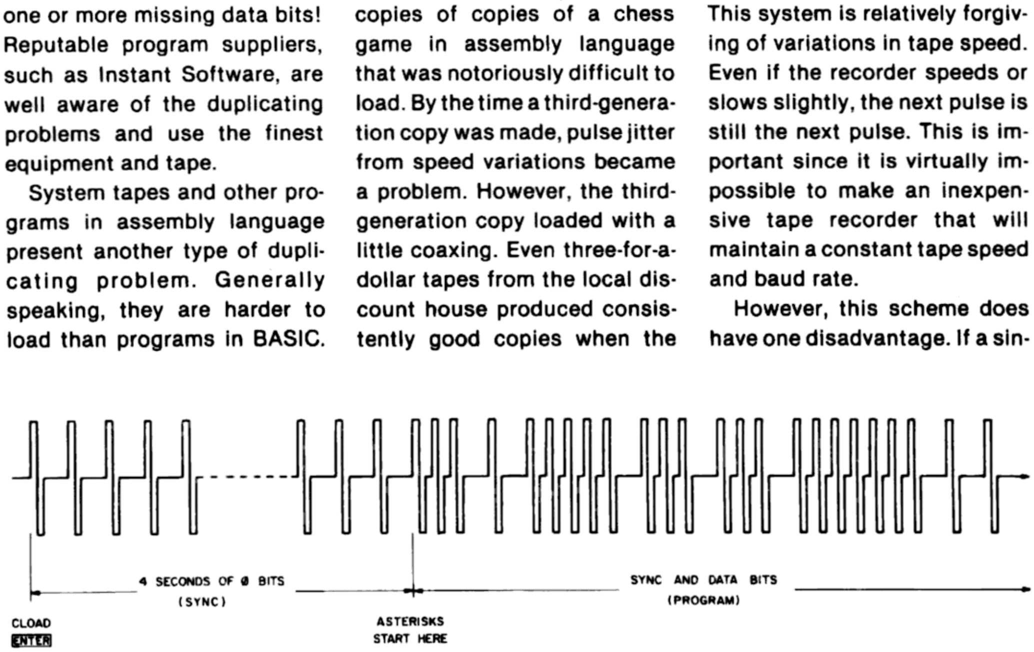


Fig. 1. TRS-80 data stream.

gle sync or data pulse is lost anywhere along the line in a TRS-80 recording, the entire transmission will become out of step. This results in a defective load and all that garbage on the screen when you try to list the program.

The sync and data pulses from the TRS-80 have a bipolarity waveform as shown in Fig. 2a. These pulses are applied to the recorder input during a CSAVE operation. This unusual waveform was selected by Tandy engineers for a specific reason. Because of frequency-response limitations, the recorder and tape cannot faithfully reproduce the waveshape shown in Fig. 2a. The poor high-frequency response of the recorder causes the leading and trailing edges to droop. Poor low-frequency response cannot maintain the flat tops of the pulses, which eventually round off. As a result, the frequency-response distortion of the cassette recorder turns the waveform into a near perfect complete cycle of audio (Fig. 2b). This is exactly what the engineers want on the tape! Crafty, those Texans.

Ideally, the playback waveform will resemble Fig. 2b. The sync pulse will allow the following data pulse (if one is present) to input memory. The successive stream of data pulses loads the memory and constitutes the program stored on the tape.

Unfortunately, the ideal single cycle of audio waveform is not always recorded on tape. Depending on a number of factors (tape characteristic, recorder bias, record level, transformer phasing, etc.), the waveform can be substantially modified from the ideal. Some commercial program tapes have so much hum and waveform distortion that it is truly a miracle that they load at all. It is also amazing how forgiving the TRS-80 is of these poor tape recordings.

The waveform in Fig. 2c was noted coming from both Panasonic and Sanyo recorders. This is not to imply there is anything wrong with these machines. However, the pulse characteristics are quite different than those of the CTR-41 supplied by Radio Shack.

Note the predominant double-positive pulses. While the TRS-80 appears to prefer positive-going pulses, it does not like to see two of them! The "bumps" above the baseline tend to double-trigger the TRS-80. Incidentally, this waveform is often seen on commercially duplicated tapes, and the TRS-80 despises it. Early versions of Radio Shack Microchess were improperly duplicated and exhibited this waveform. The two positive peaks were the same amplitude as the negative peaks! As a result, the tapes were difficult or impossible to load.

When predominantly positive or negative pulses occur, excessive input levels are often required to get the loading "stars" to appear. Sometimes reversing the phase of the waveforms will greatly improve the ability to load the program tape. Those of you who are technical types can accomplish this by simply reversing the secondary leads of the speaker transformer in the cassette recorder.

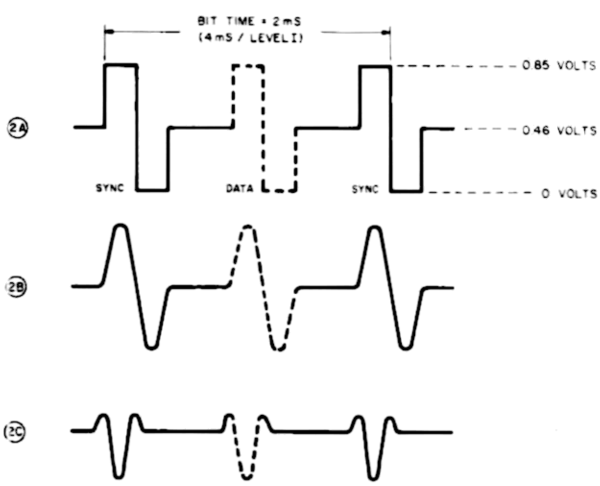
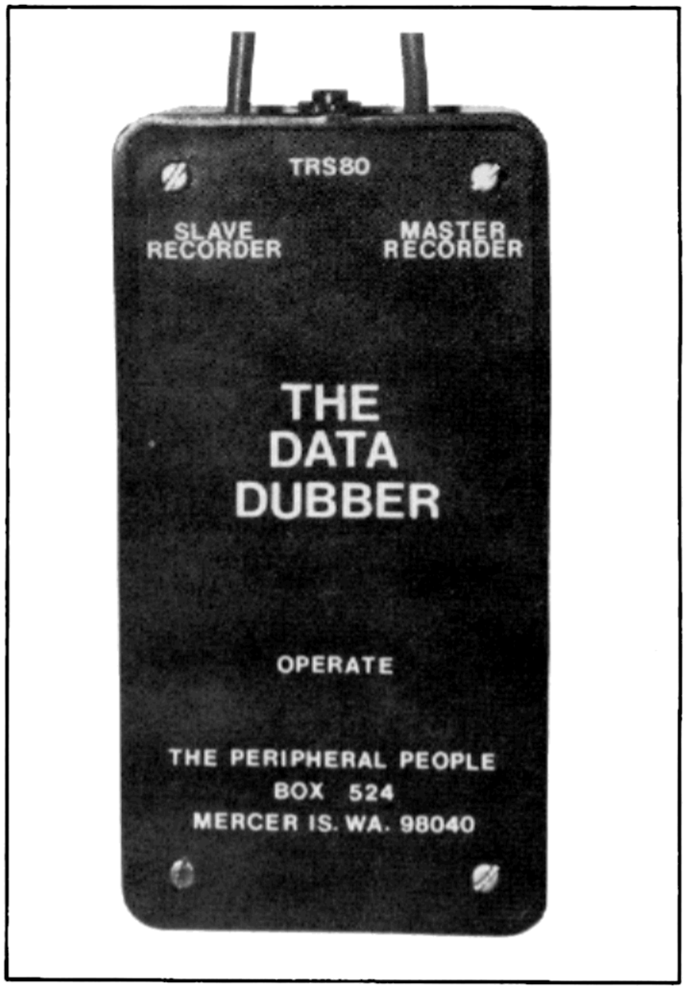


Fig. 2. Cassette-recorder waveforms.



An electronic switch turns on the Data Dubber when pulses are applied. This eliminates the usual toggle switch and eliminates battery drain when used by forgetful computerists. An LED indicates the unit is operating.

Program Tape Problems

Dropouts are a problem with all tapes. If you hear a click, empty space or a sudden volume change, it will usually mean that data has been lost. Generally, dropouts are caused by defective oxide or wrinkles on the tape. Sometimes, increasing the tension on the pressure pad in the cassette cartridge will bring the level up sufficiently to provide a satisfactory load. However, if a sync or data bit is actually missing, even the divine intervention of the Data Dubber will not help.

Power-line audio hum is another common problem with program tapes. The TRS-80 hates hum with a passion. If you hear hum when listening to a program tape, it indicates defective duplicating equipment and will cause loading difficulties. The tape should be returned to the supplier for credit or replacement.

There is another ac problem that few people realize. If your work area looks like the average computerist desk, you probably have line cords dangling all over the place. You can actually damage a cassette by leaving it on an active line cord for a day or two. With a thin diskette, partial erasure can happen in an hour or two. Even though the ac field around the line cord is tiny, over a period of time, it can demagnetize the magnetic medium. Obviously, you must keep your tapes away from any ac fields, weak or strong.

Amplitude changes are also a common source of loading difficulty. If the active program volume is low compared to most of your other tapes, the problem is probably caused by a head-alignment difference between the duplicator and your cassette. Quite often you can obtain a significant increase in volume by shimming up the cartridge with paper stickers to achieve better alignment. Try wiggling the cartridge around while it is playing to see if shimming will help.

If most of your purchased tapes can be improved by shimming, it probably means the head in your cassette, and not the duplicator, is at fault. In this case, you should invest in a head-alignment test tape. If you don't mind butchering your CTR-41, you can drill a hole just above the head-alignment screw and trim each tape for maximum volume and high pitch tone.

If the volume or tone of the tape seems to wander around, it could be caused by a number of problems. Probably the most common is oxide buildup on the record/playback head (either your unit or the duplicator). The head must be kept clean at all times. Always keep a bottle of alcohol and a pack of cotton swabs handy for head cleaning.

Slow periodic volume and tone changes can also be caused by tape weaving. This actual wandering of the magnetic field causes variations from proper alignment. Shimming may help this problem (by centering the wander), but usually, you will have to play games with the volume control to get a good load.

The Data Dubber

The Data Dubber was designed to cope with most of the situations just described. The concept of the Data Dubber is relatively simple, and a block diagram is shown in Fig. 3. Ideally, the tape recorder should be fed bipolarity CSAVE pulses. To duplicate a tape, the problem is to take the sinewaveform data of the incoming tape signal and recreate duplicates of the bipolarity pulses that produced it in the first place. This is exactly what the Data Dubber does. The idealized pulse is then fed to a second cassette recorder and copied. Thus, the Dubber can be connected between two cassette recorders to make a clean new tape.

The TRS-80 also likes to "see" bipolarity pulses. A second Dubber output provides pulses that are optimized for the TRS-80 CLOAD input. Unless there is a missing pulse on the tape, the Data Dubber will permit perfect CLOADs, usually on the first try. The playback level is not critical when the Dubber is used. You can actually run the volume up and down or flip the CTR-41 tone switch back and forth while a tape is loading, and the Data Dubber will ignore the disturbance.

Between now and the time your next issue of *80-Microcomputing* arrives, take a few minutes to listen to the pulse recordings. They should be loud, steady and have a sharp staccato sound. The leader (that steady sound at the beginning) should come on steady with no clips or pops. A sudden burst, then volume reduction, indicates the tape was recorded too loud and the recorder automatic volume control had to correct the level. Learn to recognize what a good tape sounds like so you can analyze the problems with a bad tape.

The second part of this article will tell you how to build a Data Dubber to solve all of the aforementioned problems, plus a few more that have not been mentioned. The Data Dubber is available from The Peripheral People, Box 524, Mercer Island WA 98040. ■

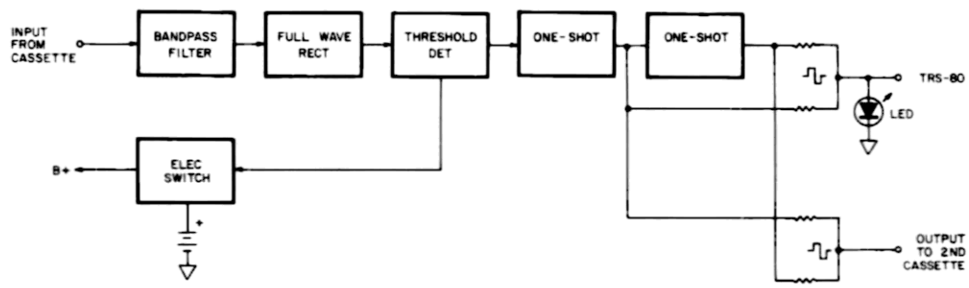


Fig. 3. Data Dubber block diagram.

NEW VERSATILITY For Your TRS-80

FROM THE ORIGINATOR OF THE TRS-80 PROJECT

FMG CORPORATION

HEADQUARTERS FOR PASCAL

And Other High Level Languages

NEW

PASCAL COMPILER

Pascal/MT™ is designed to run under CP/M. It provides an efficient development cycle plus efficiently executing object programs using a highly structured language so fewer programming errors are made. Pascal/MT has extensions to standard Pascal.

Features:
- ***Compiler executes only in 32K.***
- ***Direct I/O manipulation.***
- ***Logical Functions allow bit manipulation.***
- ***Assembly languages interface.***
- ***Object programs execute ten times faster than P-code system.***
- ***Includes real time symbolic debugger.***
- ***Accurate 16 digit BCD business arithmetic.***

Package Includes System Diskette and Instruction Book (Requires CP/M)
price.......................... **$100⁰⁰**

Same package (TRS-DOS version)
price.......................... **$125⁰⁰**

Same package (S-100 CPM version with 8" diskette) price.................. **$100⁰⁰**

HIGHLY RECOMMENDED PASCAL MANUALS

"Pascal Users Manual and Report"
Price, Manual Only................. **$9⁹⁵**

UCSD Reference Manual
Price, Manual Only................ **$25⁰⁰**

"Problem Solving Using Pascal," the beginner's book for UCSD Pascal
Price, Manual Only................ **$14⁹⁵**

"Programming in Pascal"
Price, Manual Only................ **$14⁹⁵**

"Pascal Primer"
Price, Manual Only................ **$17⁹⁵**

TEXTWRITER II

New Lower Price

Exclusively from FMG

A text formatting program that prints files created by an editing program. Contracts, personalized form letters and other documents can be printed from a stored library of standard paragraphs.

Price.......................... **$75⁰⁰**

Price (Manual only)............... **$25⁰⁰**

NEW

TEXTWRITER III

All features of Textwriter II plus ability to computer create Indexes and Table of Contents.

Price.......................... **$125⁰⁰**

CP/M® OPERATING SYSTEM

New 1.5 Version

Includes RS-232 and I/O Byte implementation, Editor, Assembler, Debugger and Utilities for 8080 and Z80 Systems.

Package Includes:
- ***CP/M System Diskette 5¼"***
- ***CP/M Features and facilities Manual***
- ***CP/M Editor's Manual***
- ***CP/M Assembler Manual***
- ***CP/M Debugger Manual***
- ***CP/M Interface Guide***

PRICE....... **$150⁰⁰** (Requires 16K and one drive min.)

Price, set of 5 (Manuals only) **$25⁰⁰**

Update for 1.4 version owners, new disc supplied................. **$20⁰⁰**

FORTRAN

New Lower Price

Comparable to compilers on large main frames and minicomputers. All of ANSI standard FORTRAN X 3.9-1966 is included except COMPLEX data type. Therefore, users may take advantage of the many applications programs already written in FORTRAN.

Features:
- ***FORTRAN Compiler***
- ***Macro Assembler (Z80)***
- ***Linker***
- ***Library***
- ***Lib Manager (Not in TRS-DOS version)***

FORTRAN, TRS-CP/M version price... **$200⁰⁰**

FORTRAN, TRS-DOS version price .. **$150⁰⁰**

Price for each (Manual only) **$25⁰⁰**

NEW

TRS-80 COMMUNICATOR

New RS232 Communication Program that allows your TRS-80 to transmit or receive programs and data files. Also makes the TRS-80 into a remote terminal. Requires Radio Shack RS232.

TRS-CP/M version **$25⁰⁰**

TRS-DOS version.................. **$35⁰⁰**

NEW

TRS-COBOL-80

FMG's Microsoft COBOL brings the world's most widely used computer programming language to the TRS-80 user. FMG COBOL-80 is comparable to COBOL systems found on minicomputers and large mainframes. Consequently, it greatly enhances the usefulness of microcomputers because it gives users access to the incredibly large numbers of programs already written in COBOL. Because COBOL-80 is a standard, COBOL programs written on other computers may be run easily on the TRS-80.

FMG TRS-COBOL-80 is based on the 1974 ANSI standard. It combines all Level 1 features and the most useful Level 2 options for the "Nucleus" and for sequential Relative and Indexing file handling facilities.

The FMG TRS-COBOL-80 system includes a compiler for translating source code into relocatable object code (which, incidentally, is compatible with the object code of our FORTRAN-80 compiler and MACRO-80 assembler), and a routine system for running the program by interpreting the object code at execution time. It also includes screen handling capabilities for easy use of interaction.

Requires 48K, CP/M and 2 Mini Disk Drives.

Price.......................... **$750⁰⁰**

UCSD PASCAL

The powerful, general purpose language system, originally developed for large, complex system, is now available from FMG for your TRS-80. This new FMG/UCSD Pascal System greatly increases the value and capability of the TRS-80.

Package Includes:
- ***Operating System***
- ***Screen Editor***
- ***Z80 Macro Assembler***
- ***Library***
- ***Pascal Compiler***
- ***Utilities and System Reference Book***

Price.......... **$150⁰⁰** (Requires 48K, 2 drive System)

available without Macro Assembler, Linker and Library (not for compiling programs) price................. **$100⁰⁰**

For recommended Pascal manuals, refer to "PASCAL COMPILER" listing.

Call or write for complete information

FMG CORPORATION

(817) 294-2510

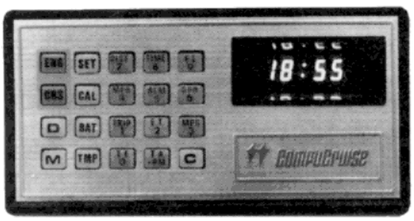
P.O. Box 16363(80) • Fort Worth, Texas 76133

High Level Languages for the TRS-80

BASIC • FORTRAN • COBOL • PASCAL

CP/M is a registered trademark of Digital Research Corp. • TRS-80 is a registered trademark of Radio Shack.

If you enjoy driving, you're going to get a COMPUCRUISE. Once you see what it can do, you just won't be able to live without it.

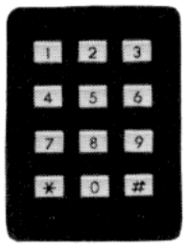


This gadget fits into most dashboards . . . no strain even in a tiny sports car like the Mazda RX-7 . . . and once you have it, every trip is like flying a 747. The darned thing tells you the time, how fast you're going, how far you've been on this trip or since the last regassing, how many miles per gallon you're getting, either at the instant or the average on the trip . . . or gallons per hour at the moment or for the trip . . . temperature outside . . . inside (or coolant temperature, if you prefer) . . . oh, it has an elapsed time for the trip, a stop watch, lap time, an alarm . . . how much further for your trip, how many gallons more the trip will take, how much longer for the trip at your present average speed . . . yes, it gives you your average speed for the trip. You prefer it in metric, no strain . . . liters remaining, etc. Did we mention that it also has cruise control either at a speed set on the control board or at whatever speed you are traveling? The Compucruise will keep you busy and entertained during any trip . . . telling you more than you will ever want to know.

The Compucruise is not difficult to install . . . though it does connect to everything except the cigarette lighter. Until you've tried computerized travel, you haven't found out how much fun driving can be. It will work on any car not having fuel injection . . . and there is a front-wheel drive accessory gadget available for only $4.40-#P001 (regularly $5.50).

The price for the Compucruise is regularly $199.95 . . . and a bargain at that price. We'll sell you one of these fantastic gadgets for $159.95 with cruise control (Model 44-#P002), and $127.95 without (Model 41-#P003). Send money . . . and start having fun!

MOM'S
MAIL ORDER MICROS
POB 427 • Marlboro NH 03455



A CAR IS STOLEN EVERY 32 SECONDS. ***Will yours be next?***

Do you want to drive a car thief completely batty? Put one of these little buggers in your car and watch the fun. You can even leave your keys in the ignition and the best of thieves won't be able to get your car started. Hot wiring? No way, not from the passenger compartment.

To start your car, you must punch in the correct four digit code . . . it's a digital combination lock. Not bad for $49.95 . . . and even better for our special price of **$39.95 (#P004)** (that's 20% less than the manufacturer's suggested retail). We're in business for the fun of it, not to make money, so we're able to pass this gadget along to you at a real bargain price. Buy one now, before we come to our senses!

20% OFF Inventory Clearance SALE

3 Parallel Keyboard (can be used with Poly-88)—Good condition. #S001-$50 each. **SPECIAL PRICE.**

1 PolyMorphic System 88 Software on 5¼"—This is system software that requires a PolyMorphic Disk Controller in a System 88 Cabinet. #D002-$100.

1 Electric Pencil for Poly 88—Word Processor. #D003-$80.

4 Processor Tech. SOL Computers—8K Memory, S-100, excellent condition. #D004-$980 each.

19 TREK-80 on cassette for SOL—This is one of the best real time space games available today; needs 8K. #D005-$11 each.

9 Electric Pencil on cassette for SOL—Word processor, needs 8K. #D006-$80 each.

1 Processor Tech ROM/RAM card—Contains 1K RAM, empty slots for 12K ROM, S-100, excellent condition. #D007-$76.

3 Processor Tech 3P + S I/O card—Three parallel ports, one serial port, S-100, excellent condition. #D008-$116 each.

9 Processor Tech Video Display Module—Memory mapped video for S-100, excellent condition. #D009-$144 each.

2 Processor Tech 16K Static RAM card—450 nanoseconds, S-100, excellent condition. #D010-$199 each.

4 Processor Tech 2K ROM card (kit)—Empty board, S-100, new. #D011-$17.20 each.

4 Processor Tech 2K ROM card—Empty board, S-100, excellent condition. #D012-$25.20 each.

19 BASIC 5 from Processor Tech—This is a simple BASIC for a SOLOS, CUTTER, or CONSOL Monitor and 8K of RAM. #D013-$11.60 each.

4 Processor Tech GAMEPAC for above BASIC—Various simple games. #D014-$11.60 each

2 Processor Tech Extended Disk BASIC—This is full disk BASIC on 8" disk for HELIOS II disk controllers with PTDOS and greater than 16K. #D015-$80 each.

1 Extended Disk BASIC on cassette—This is the same as previously mentioned for the Disk BASIC from Processor Tech. Needs more than 16K. #D017-$72.

8 Extended Cassette BASIC from ProTech—This includes all file operations, advanced functions for doing more than playing games; for SOLOS, CUTTER, and CONSOL Monitors. #D016-$22 each.

14 ALS-80 Operating system—This system requires 12K RAM from D000 to FFFF; as well as either the SOLOS or CUTTER monitor; it includes an Assembler/Editor. #D018-$11.60 each.

8 PolyMorphic Video Terminal Interface—Memory mapped video for S-100 bus, good to excellent condition. #S044-$150 each. **SPECIAL PRICE.**

1 PolyMorphic Video Terminal Interface (used)—As above. #S045-$100.00. **SPECIAL PRICE.**

3 PolyMorphic CPU—8080 CPU for S-100 with Poly monitor in ROM, excellent condition. #S046-$100 each. **SPECIAL PRICE.**

4 PolyMorphic Cassette Interface—Plugs into Poly CPU only, new. #D047-$29.60 each.

4 PolyMorphic Printer Interface—Plugs into Poly CPU only, new. #D048-$29.60 each.

4 PolyMorphic S-100 Cabinets—Nice 5 slot S-100 mainframe, good to excellent condition. #D049-$248 each.

17 Tech Design Consultants Income Tax Manager—This requires the above BASIC and is for the year 1977. #D019-$11.60 each.

1 IASIS COMPUTER-IN-A-BOOK Computer Trainer—8080 Microcomputer, comes built into training manual, excellent condition. #D020-$240

6 Novation Modem #3102A—Connects to any phone, originate only, good condition. #S021-$165 each. **SPECIAL PRICE.**

2 Novation Modem #3103A—Connects to any phone, answer/originate, good condition. #S022-$200 each. **SPECIAL PRICE.**

1 Novation Modem #43—Connects to any phone, originate only, good condition. #S023-$165. **SPECIAL PRICE.**

112 Abacus Paperweight—Hefty, brass, excellent condition. #S024-$3 **SPECIAL PRICE.**

2 Compucolor Computer 8001 (use as computer or 75 MHZ Color Monitor)—8K RAM, BASIC and DOS in ROM, good condition. #S025-$1500 each. **SPECIAL PRICE.**

2 Compucolor MiniFloppy—5¼ inch, good condition. #S026-$525 each. **SPECIAL PRICE.**

4 Compucolor 8K RAM card—Static RAM, good condition. #S027-$200 each. **SPECIAL PRICE.**

3 Compucolor Floppy Tape Drive—Uses eight track cartridges, good condition. #S028-$70 each. **SPECIAL PRICE.**

2 Bally VideoCode Cassettes—We received only four of these; they consist of two games: Speed Math and Bingo Math. #D029-$16 each.

2 MicroTerm ACT Terminal—Need a video monitor, up to 600 Baud, good condition. #S035-$200 each. **SPECIAL PRICE.**

1 Shugart MiniDisk Drives—With case and power supply, includes cable for North Star, good condition. #S036-$320. **SPECIAL PRICE.**

4 Shugart MiniDisk Drives—No cabinet, good condition. #S037-$300 each. **SPECIAL PRICE.**

4 Power Supplies for above—good condition. #S038-$30 each. **SPECIAL PRICE.**

2 Power Supplies for above—good condition. $30 each. **SPECIAL PRICE.**

7 Mountain Hardware AC Controller—Remote AC outlet control, S-100, new. #D040-$100 each.

4 Mountain Hardware remote outlet—Remote module for above, two channels, new. #D041-$72 each.

1 ICOM Dual Disk Drive—Single density, 512K storage, S-100 controller, includes CP/M ROM, good condition. #S030-$1500. **SPECIAL PRICE.**

1 ICOM PROM and 8" Disk for SOL FDOS—This disk requires an ICOM S-100 Disk Controller installed in an S-100. #D031-$160.

1 ICOM CP/M on 8" Disk for S-100—Requires an ICOM S-100 controller in an S-100 cabinet, #D032-$100.

2 ICOM FDOS-II on 8" Disk for S-100—Requires an ICOM S-100 controller in an S-100 cabinet, no documentation. #D033-$180 each.

1 ICOM FDOS-II on 5¼" Disk for S-100—Requires an ICOM S-100 Mini-Floppy Controller in an S-100 cabinet. #D034-$168.

1 Heuristics Speech Lab—S-100, used, fair condition. #S042-$100 as is. **SPECIAL PRICE.**

4 Heuristics Speech Labs—S-100, new. #D043-$151 each.

120 S-100 Edge Connector—Gold Contacts, new. #D050-$.20 each.

6 Extender Card for S-100 (kit)—New. #D051-$24 each.

Prices include 20% discount.
SPECIAL PRICE includes more than 20% discount.

TERMS: FOB Marlboro, NH USA. Limited stock; everything guaranteed as described; you pay postage on returns. PRINT orders clearly. Minimum order $10 plus $2.50 shipping and handling charge in USA only. DOUBLE THAT ELSEWHERE. Orders over $50 add 5% for shipping in USA; 10% elsewhere (we will refund excess). Orders shipped UPS or insured mail only. Send US funds by check or money order. For credit card purchases, add 3%, list AE, MC or VISA, number, and expiration date. Mail to MOM's, POB 427, Marlboro NH 03455.

Please specify catalog number and name of item when ordering.

Condition of Inventory:
New = original container   Excellent = new, but not in original container   Good = tested or used in store

☐ Yes! Sign me on as a Charter Subscriber to 80 Microcomputing for $12 (half the newsstand price) and send me your giant book of 80 Microcomputing programs—FREE!

☐ 12 issues—$12 ☐ 36 issues—$36
☐ Payment enclosed ☐ Please bill me
☐ Bill my: ☐ Master Charge ☐ Visa

Credit card# ______
Exp. date ______ Signature ______
Name ______
Address ______
City ______ State ______ Zip ______

Canada: $12 per year, U.S. funds only. All foreign subscriptions $20, one year only, payable in U.S. currency.

80 microcomputing

301K33

Subscription Dept. • Box 981 • Farmingdale NY 11737

Bring the cursor to your attention with this subroutine.

Winking Cursor

Daniel Lovy
2398 Hulett Road
Okemos MI 48864

Many computers feature a winking cursor which, some users feel, enhances input of information. This article concerns a short subroutine written for the TRS-80 Level II that will give input statements a winking cursor or winking graphics block.

To use this subroutine, you first PRINT the question that is being asked, then execute GOSUB 1000. The cursor will blink as data is input. When the ENTER key is hit, control will return to the main program with the input information stored in FI$. If the data is to be stored as a number, it will have to be converted using the VAL function (see Example 1).

Winking cursor subroutine.

```
 999 END
1000 PO = 256•(PEEK(16417) – 60) + PEEK(16416)
1010 PRINT @ PO,CHR$(95);: FOR WAIT = 1TO 35:NEXT
1020 IN$ = INKEY$
1030 IF IN$ = ""THEN PRINT @ PO," ";:FOR WAIT = 1 to 35:NEXT:GOTO 1010
1040 IF ASC(IN$) = 13 THEN PRINT @PO," ":RETURN
1050 IF ASC(IN$)<32 THEN POKE 15360 + PO,32
1060 PRINT @ PO,IN$;
1070 FI$ = FI$ + IN$:GOTO 1000
```

```
10 PRINT"WHAT IS YOUR NAME?";:GOSUB 1000
20 A$ = FI$
30 PRINT "YOUR NAME IS";A$
```

Example 1

If you prefer a winking graphics block to a cursor then change line 1010 to . . . CHR$(143). . . The rate of blinking can be changed by altering the length of the two FOR-NEXT loops in lines 1010 and 1030. ■

Connect your 80 to a Teletype 33 printer, with less than $40 worth of electronics.

Level II To Model 33

Lauren A. Colby
532 Pearl St.
Frederick MD 21701

If you own a TRS-80 computer and also own an ASCII Teletype, such as the Model 33, you can interface the computer to the Teletype and enjoy the convenience of hard-copy printouts. You will need Level II and the expansion unit to do the job.

You will also need two UARTs (universal asynchronous receiver-transmitters). These are 40-pin chips. There are two types: One works off a single 5 volt supply; the other requires two supplies— +5 volts and −12 volts. I have the kind that requires a dual supply. However, you can simplify matters if you get the kind that uses only a single 5 volt supply. Order UART# AY-51015A/1863 from Advanced Computer Products, Box 17329, Irvine CA 92713. Price is $6.95 each.

You will need to make a special connector to access the line-printer port on the expansion unit. You will then need to construct some circuitry to convert the parallel data appearing on the port to serial data for use by the Teletype machine. Finally, because the computer can't generate certain line-feed signals that the Teletype needs in order to operate properly, you will have to build some circuitry to generate these signals.

Connections to Line-Printer Port

The line-printer port consists of a 34-pin, male edge connector on the side of the expansion unit. Pin 1 will be marked. All of the odd-numbered pins are on the *same* side of the board as pin 1, so that the pin numbers on that side of the board will be 1, 3, 5, 7, etc. Pin 2 is directly opposite pin 1 on the other side of the board so that the pin numbers on that side of the board will be 2, 4, 6, 8, etc.

To make a female connector to mate with the male edge connector, go to any electronics store that sells Vector Products and purchase a PC receptacle, number R636-1DP. This is a 72-contact edge connector with 0.1-inch spacing. With a small coping saw, saw off the end of the connector between pins 18 and 20 (as marked on the Vector product, not the computer). Obtain some ribbon epoxy and mold a new "end" around the cut, so that the connector will slide over the contacts inside the expansion unit and all the pins will match up properly. Warning: The pin markings on the Vector product do not match the pin numbers on the computer expansion unit. Go by the pin numbers on the expansion unit; pay no attention to the numbers on the Vector connector.

Fig. 1 shows the pin-numbering system at the line-printer port. The eight data lines, D-0 through D-7, generate the ASCII code required by the printer. Actually, the eighth bit is a parity bit that is not used; you can hook it up as shown in the figures or just leave it dangling.

Grounded pins: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 27, 31, 33, 34.
Not connected: 19, 29, 30, 32
Data lines: D-0 pin 3 (LSB)
D-1 pin 5
D-2 pin 7
D-3 pin 9
D-4 pin 11
D-5 pin 13
D-6 pin 15
D-7 pin 17 (most significant bit)
Pin 21 is "busy" to expansion unit (logical low, not busy, ready for new data; logical high, wait).
Pin 23 is "Out of paper" to expansion unit (logical high-out of paper, wait).
Pin 25 is "Unit select" input to expansion unit (logical high—unit selected, all OK; logical low, unit not selected, wait).
Pin 28 is "fault" input to expansion unit (High, OK; Low, wait).
Pin 1 is the strobe pin. Generates a 1.5 microsecond low to strobe the printer.

Fig. 1. Line-printer port pin connections.

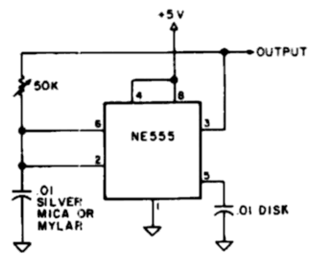


Fig. 2. Clock circuit.

Pin 21 is a handshaking line. The computer can output ASCII data at incredible speeds; however, the Teletype can only print about 100 words per minute (110 baud), so pin 21 is used by the Teletype and associated circuits to tell the computer to withhold sending any more data until the Teletype is ready for it. Pin 1 is a strobe pin used by the computer to tell the Teletype that new data is ready to be sent out. Pin 23 is used in this project for a special purpose. Its use will be described later. Pins 25 and 28 are not used; they must be left unconnected.

Parallel-to-Serial Conversion

The UARTs you will be using in this project must be clocked at 16 times the baud rate. The baud rate of a Model 33 Teletype is 110 baud, so the clock must be operated at 1760 Hz. Fig. 2 shows the clock circuit. It should be adjusted to proper frequency with a counter. My experience of about nine months has shown this clock to be suffi-

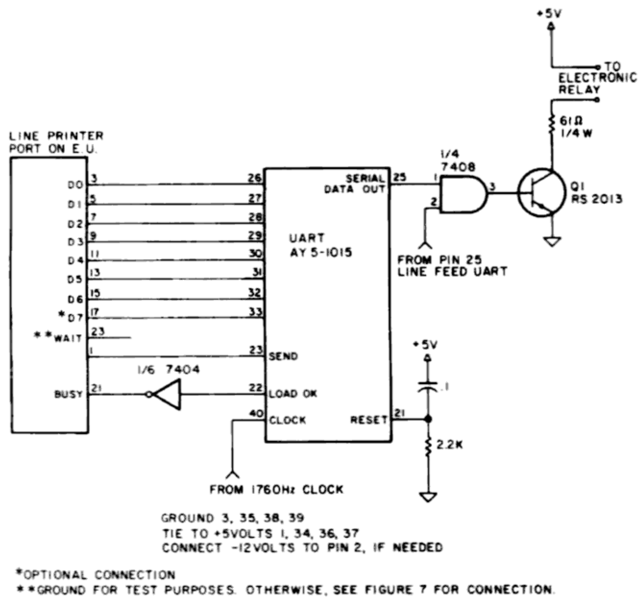


Fig. 3. Parallel-to-serial-conversion circuit.

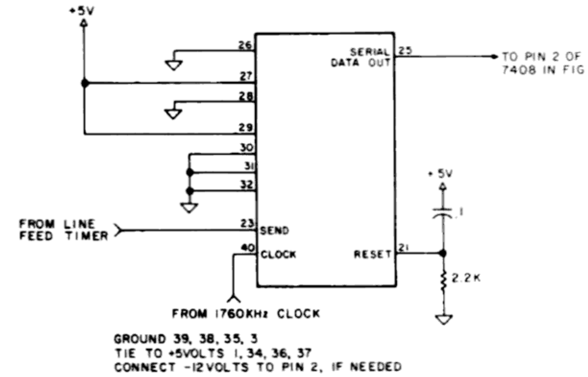


Fig. 5. Line feed UART circuit (AY5-1015 or equivalent).

ciently stable; a crystal clock is not needed. However, be sure to use a silver mica or Mylar capacitor as the timing capacitor.

Fig. 3 shows the circuit used for parallel-to-serial conversion. Parallel data enters the UART at pins 26 through 33, but the eighth bit, applied to pin 33, is a parity bit that is ignored and not used. When the computer wants to output some data, it emits a brief negative pulse to pin 23 of the UART. The UART then outputs a low at pin 22, which is inverted and applies a high to pin 21 of the line-printer port. This tells the computer that the UART is busy loading data for transmission to the Teletype. When the load is complete, UART pin 22 goes high, applying a low to pin 21 of the line-printer port and telling the computer it is OK to send more data.

Serial data is output from pin 25 of the UART and applied to an AND gate (¼ of a 7408. For testing purposes, the free input of the AND gate can be tied to +5 volts. The data will then appear at the output of the gate and activate the transistor, Q-1. The output of Fig. 3 is applied to the electronic relay, Fig. 4, which operates the Teletype's local 20 or 60 mA loop.

For test purposes, jumper pin 23 of the line-printer port to ground. With the circuits of Figs. 2, 3 and 4 constructed and operational, and the free gate input tied to the 5 volt line, as suggested, you should be able to command your TRS-80 to LPRINT different things, and it should do so. For example, if you enter the command LPRINT "THE QUICK BROWN FOX", the computer should cause the Teletype to print, THE QUICK BROWN FOX.

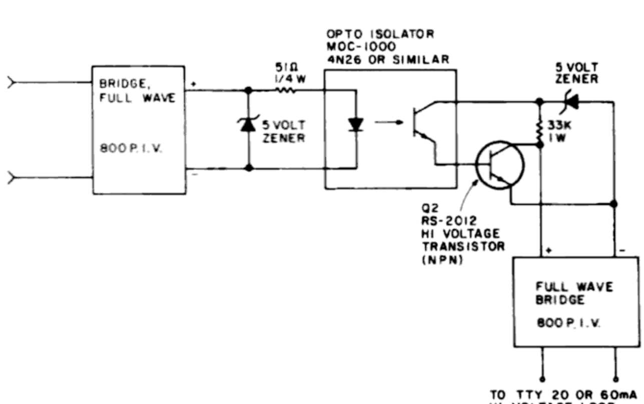


Fig. 4. Electronic relay circuit.

Generating the Line-Feed Function

After your computer has printed THE QUICK BROWN FOX, it will execute a carriage return. It will not, however, generate a line feed. To get a line feed you must enter LPRINT CHR$(138). In the LLIST mode, there is no way that you can get a line feed. Thus, the computer will just keep typing on the same line, over and over.

The problem here is that the Radio Shack printers automatically execute a carriage return and a line feed when they get a carriage-return signal. Your Teletype does not, so you need some more circuitry to generate a line feed when the computer puts out a carriage return.

To accomplish this, we use a second UART, the line feed UART, which is hardwired to output a line-feed signal (a ten in binary). Fig. 5 shows the circuit. I suggest you construct it and test it by disconnecting the jumper from the free input of the AND gate to +5 volts (previously installed for testing purposes) and connecting pin 25 of the line-feed UART to the gate input. Now, momentarily bring pin 23 of the line-feed UART to ground. Every time you do this, the printer should execute a line feed.

You now have the capacity to generate a line feed on command. Also, by removing the jumper from pin 23 of the line-printer port to ground, you can make the computer stop and wait and not send any more data until you bring the pin low, once again. If you could really move fast, you could put your computer in the LLIST mode, let it print out a line, quickly bring pin 23 of the line-printer port high (thereby stopping the computer and making it wait), bring pin 23 of the UART low, generating a line feed, and then bring pin 23 of the line-printer port low, so the computer would print out another line.

But obviously, you'd have to move fast, and it would be an awful lot of work. You need a method to do everything automatically.

To do this, it is necessary to decode the carriage-return signal from the computer. It is a binary 13. Fig. 6 shows the decoding circuit. Whenever the computer outputs a carriage return, the output will go low.

Finally, it is necessary to construct two timers (Fig. 7). Both timers are activated by the negative-going pulse from the decode line. The first timer generates a 300 ms positive pulse, which goes to the "out of paper" line and tells the computer, "Wait, we're getting ready to generate a line feed—don't do anything for 300 ms." The second timer generates a 150 ms positive pulse. When that pulse goes away, the line-feed UART sees it as a negative-going strobe pulse and generates the line feed. Finally, when the 300 ms pulse goes away, the computer gets the message, "Wait no longer, proceed to send more data."

Watch Out for Pin 23

In reading over the circuit descriptions and examining the figures, please watch out whenever there is a reference to pin 23. Through an unfortunate coincidence, there are three different pin 23s, doing three different things. Pin 23 of the data UART is a strobe that receives a negative pulse and activates the UART whenever the computer sends out a character to be printed. Pin 23 of the line-feed UART is a strobe that receives a negative pulse from the decoder whenever the computer puts out a carriage return, thereby activating the line-feed UART and generating a line feed.

Pin 23 on the line-printer port (expansion unit) is an interrupt or wait line, which causes the computer to wait whenever it goes high. This line is delineated as an "out-of-paper" line by Radio Shack. However, that particular designation has no significance in this application. It is merely a "wait" line; the computer doesn't know or care for what purpose it is being told to wait.

Construction Details and Comments

I constructed this whole project on two pieces of Radio Shack perfboard, the kind with 22-pin edge connectors. The two boards are linked together by soldering the pins of an edge connector to board A and plugging board A into board B. The ICs used were based on what I had readily available. You could probably simplify matters by substituting a 7430 eight-line decoder for the two 7420s and the OR gate (7432). That would get rid of two ICs and might make it possible to get everything on one board. However, I can't guarantee this would work, because I haven't tried it.

On each of the UARTs, note the little circuit tied to pin 21. Often, in books and articles, this pin is shown as tied to ground. Grounding this pin is a principal reason for hang-ups and erratic operation. The UART needs a brief positive pulse when the power is first turned on to initialize it.

In constructing the circuit, liberal quantities of bypass capacitors were installed, in accordance with standard TTL construction practice. Consult Don Lancaster's *TTL Cookbook* for information on bypassing techniques.

The interface should be powered by a separate, well-regulated 5 volt supply (and a −12 volt supply, if needed by your UARTs). It is risky to try to derive power from the computer or the expansion unit for any peripherals. The *Radio Shack Technical Manual* recommends against it.

I have been using this interface with no problems for about nine months. Total construction cost, not including power supplies, was under $40. ■

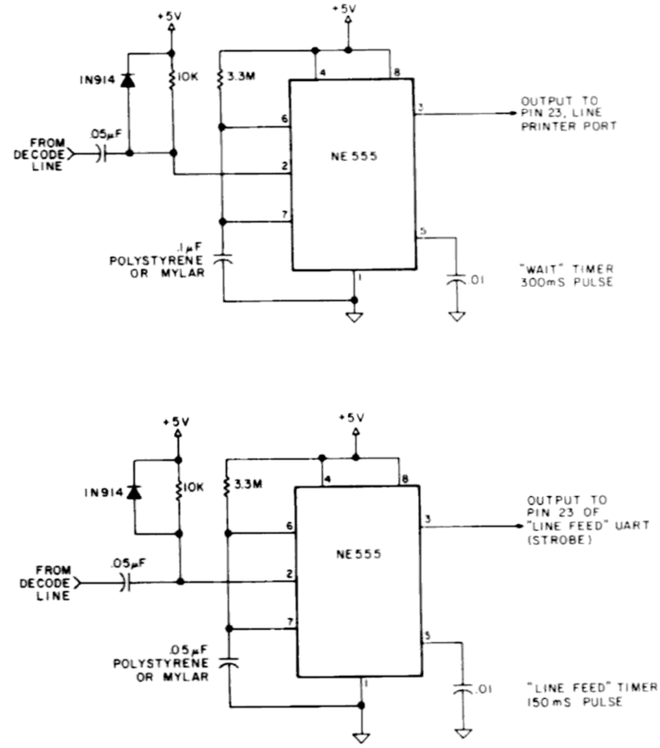


Fig. 7. Wait and line-feed timers.

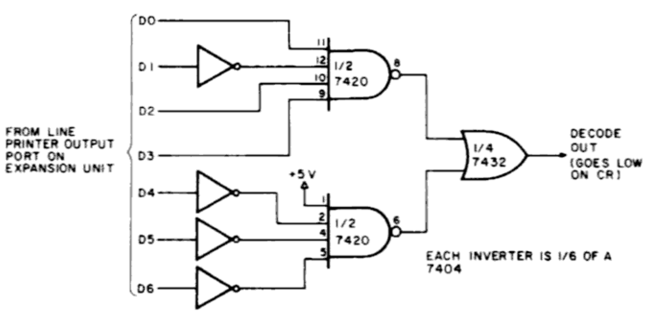
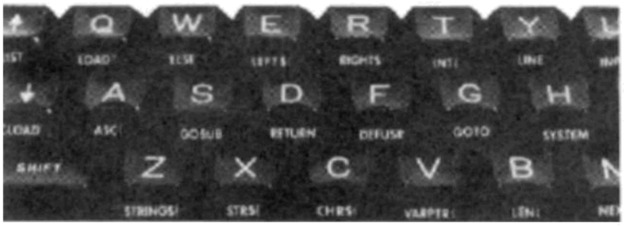


Fig. 6. Decoding circuit (decodes carriage return—binary 13).

WEB ASSOCIATES

EXCLUSIVE TO TRS-80™ USERS

TSHORT™ – THE GREATEST SOFTWARE BUY OF THE YEAR! HAVE YOU BOUGHT YOURS YET?



SHORTHAND for LEVEL II and DISK BASIC

- TSHORT™ lets you type LEVEL II and/or DISK BASIC more quickly and accurately than ever before. Save up to 90% programming time and achieve 100% accuracy.
- 31 preprogrammed statement keys.
- KUSTOM™ key, up to 64 characters — changeable anytime.
- 42 key decals (see picture). 10 are different for DISK users.
- A single, shifted-key entry types entire statement on screen.
- Fast, efficient machine language.
- Uses 580 bytes of LOW memory. i.e.: No MEM SIZE req'd.
- Comes on cassette, one side LEVEL II, the other for DOS.
- Compatible with DOS 2.1, 2.2, 2.3, NEWDOS, KBFIX, etc.
- DOS version loads to and executes from disk via TAPEDISK.
- Features self-entering commands: CONT; GOTO10; KUSTOM (Self-enter optional)
- TSHORT W/4 page instruction manual **$9.95**

OTHER PRODUCTS AND SERVICES:

TBEEP™ For LEVEL II and DISK USERS — A self-contained beeper. Alerts you with a pager-like tone when YOUR program commands it! Simply plug in-line with the "AUX" cable from your CPU and program in BASIC, i.e.: OUT 255, 1: FOR I = 1 to 300: NEXT: OUT 255, 0. (Req's 9V Batt.) $19.95

TBUFF™ – For LEVEL II CASSETTE USERS. Prevent and eliminate forever, cassette relay sticking. TBUFF is no larger than an ice cube and plugs in-line with the "REMote" cable to your cassette recorder. (Specify cassette recorder make and model) . . . . $ 9.98

TPAK™ – C-10 blank tape cassettes. AGFA 611 high quality tape — the best money can buy! We use this tape exclusively for TSHORT. Pack of 10 with box and blank labels. . . . $12.95

COMING SOON: (Write for further details.)

TBASE™ – A powerful DATA BASE MANAGER program second to none! . . . . for under $50.00

TCHAIN™ – LEVEL II chaining utility — Preserve your variables and arrays for multiple program use, or while EDITing, RUNing or CLOADing . . . . Priced under $ 10.00

TSEL™ — We'll convert your IBM SELECTRIC to a high quality printer — up control — 512 character buffer — special TRS-80 cable with control switches — complete and ready to LPRINT (cleaning, minor service included) . . . . . . Priced under $800.00

DEALER INQUIRIES INVITED
Send check or money order to:

TELEPHONE ORDERS: (714) 559-6249

SORRY, NO C.O.D.'S

✓27 **WEB ASSOCIATES** • P.O.BOX 60-QA • MONROVIA, CA 91016 • (Calif. residents add 6% tax)

TRS-80:
"TRS-80 is a registered trademark of TANDY CORP."

FINDISK-II Automatically extract user file names from disks, print disk directory labels and indexed master list of all disk files. Automatic update. Optional descriptions. TRSDOS or NEWDOS **$20.00**

SOLAR-I Passive solar analysis. Input: any latitude, orientation, slope, storage, roof shading. Output: solar angles, heat gain/loss, % solar, fuel use, hourly, monthly, yearly, in presentation format. By solar architect, easy use. Comprehensive manual. . **$30.00**

SOLAR-II As above plus lifecycle cost/optimization **$50.00**

RIA-II Real Estate Investment Analysis, business or homeowner. Input: costs, loan, tax data. Output: before/after tax cash flows, rates of return, depreciation, profit/gain of sale for any time series, complex, essential investment calculations in seconds . . . **$30.00**

All 32K. Tape or Disk (one drive—order tape)

Following min.req. 16K Level-II:

DEPRECIATE-I Calculate, sums, print depreciation schedules 1 to 500 items, plus print Tax Schedule C. Automatically update 12 facts on each item such as remaining bal & life. Prompts optimum switch DDB to SL. **$10.00**

STRUCT-1 Calculate and print beam sizes, moment, shear diagrams. Optimize beam, joist selections. Simple span/cantilever, uniform and/or point loads. Also moment transfer calculations for unusual shapes. Screen and printer graphics. **$15.00**

DOCUMAN SOFTWARE ✓88
BOX 387-D KALAMAZOO, MI 49005
(616) 344-0805 VISA/MC

IBM SELECTRIC I/O PRINTER.

TRS-80®
DIRECT INTERFACE

$695.00

- Cleaned and Functionally Checked IBM I/0 Terminal
- ASC II-RS232, Parallel
- Service and Parts Manuals
- Immediate Delivery
- Heavy Duty Packing $25.00
- Shipping Collect
- Cashier Check, Money Order

McClain & Associates, Inc.
5104 E. 65th Street
Indianapolis, Indiana 46220
(317) 842-0526
✓71

THE SMART TERMINAL

SOFTWARE PACKAGE FOR THE

TRS-80

"TRS-80 is a registered trademark of TANDY CORP."

- TRUE BREAK KEY
- AUTO REPEAT KEYS
- PROGRAMMABLE 'SOFT' KEYS
- MULTIPAGE SCROLLING DISPLAY
- FULLY CONFIGURED FROM KEYBOARD
- TWO SELECTABLE LINE DIAGNOSTIC MODES DISPLAY CONTROL CHARACTERS
- MULTI PROTOCOL CAPABILITY
- GRAPHIC CHARACTERS GENERATED FROM KEYBOARD
- TRANSMIT SCREEN, PRINT SCREEN
- TRANSMIT FILE, RECEIVE & CREATE DISK FILE
- FLEXIBLE I/O LINKAGE CAPABILITY
- DISPLAY AND KEYBOARD DRIVERS CAN BE USED UNDER DOS

• **$79.95**

✓73
MICRON, INC.
10045 Waterford Dr.
Ellicott City, MD 21043
(301) 461-2721
MC/VISA accepted

Now Available: SPOOL-80
•PRINT YOUR LISTINGS WHILE RUNNING OTHER PROGRAMS.
(DISK SYSTEMS ONLY.)
$39.95

80 may be the best TRS-80 magazine, but there are others—read about some of them . . .

Rival Publications

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

I had intended this to be a review of newsletters aimed at the TRS-80 user, but some of the publications that I have received are not really newsletters according to my definition of the word. I have, therefore, expanded the coverage of this article to include TRS-80-related publications.

As the popularity of the TRS-80 grows, so does the number of businesses designed to provide the TRS-80 market with software, hardware and publications of various kinds. In a companion article I have reviewed some of the software that is available, and I intend to continue that as a series of articles as I try out different programs.

Because of the TRS-80's popularity, articles and advertisements pertaining to the TRS-80 are appearing regularly in all of the general-interest personal computer magazines. However, these magazines can only provide a certain amount of space, since they have to satisfy the owners of many different types of computers. Various individuals and companies have obviously felt, as I do, that there was a need for specialized publications directed toward the TRS-80.

I feel that there is a need for these publications because the TRS-80 owner/programmer/hardware modifier has useful information to offer his fellow enthusiasts. Why spend hours or even days trying to solve some problem or come up with a better way of doing something when someone else has already found the answer? By the same token, when you discover something that no one else has thought of, you want to be able to pass your masterpiece along.

The true newsletter is a clearinghouse of information. The thoughts of the editor, articles on various subjects and advertisements provide helpful information. But what I find most useful are the letters from readers spelling out their solutions to various problems that they have encountered.

I have not attempted to categorize the following publications, which are listed in the order in which they happened to come to hand. I have not included prices because they seem to be in a state of change; most of these publishers are apparently new to this game, and they are just now learning the hard facts of publishing economics. Also, I have listed the interval between publications, such as monthly, bimonthly or quarterly, but some publications have had difficulty adhering to their own schedules.

TRS-80 Publications

Name: *TRS-80 Computing*
Publisher: Computer Information Exchange, Inc., PO Box 158, San Luis Rey CA 92068
Interval: Monthly—slowly getting on schedule
Format: 32 pages—8½ × 11
Comments: This is a true newsletter in that the bulk of the material published comes from its readers. I have at hand issues 1 and 2. While they seem to lean towards the hardware side of computing, there is a lot of software material, including some tutorial articles.

The first issue contains a complete set of TRS-80 schematics. These are large scale and easy to read. It is possible to have both Level I and Level II BASIC in the TRS-80 at the same time. Complete modification information, as well as a comparatively easy mod that implements lowercase without losing the graphics character set, is included.

TRS-80 owners, including me, grumble about the need to plug and unplug the cords to the cassette recorder in order to rewind the fast-forward. I have seen gadgets in magazine articles and for sale that eliminated this problem, but they cost up to $40. Issue 1 of *TRS-80 Computing* has a series of modifications for the CTR-41 recorder that took me less than 15 minutes to implement and cost me a ¼ Watt resistor. Now I never have to unplug anything; record and playback are more reliable; and I can hear the data on the tape when it is being loaded. This alone is worth a year's subscription.

An interesting article by one of the designers of the TRS-80 gives the thoughts behind the decisions that were made in its design. Another by a technician in one of Radio Shack's repair depots discusses the problems that crop up most often and what can be done about them.

Other articles and letters discuss how the TRS-80 works, how to make the computerist's life easier and the latest news from Radio Shack on what is coming in the near and distant future. Issue 2 also contains a Journal-Ledger program and a Star Trek game.

At the present time *TRS-80 Computing* does not contain many ads, but that may change as it is discovered by the small-computer business world.

Name: *TRS-80 Bulletin*
Publisher: Computer Information Exchange, Inc. (address above)
Interval: Monthly
Format: 16 pages—8½ × 11
Comments: This is a companion publication to *TRS-80 Computing.* In fact, most of the material presented comes from that magazine. However, *TRS-80 Bulletin* is given away free at computer stores. Free Subscription applications are also available.

The same comments made about *TRS-80 Computing* apply here, except that there are fewer pages and more advertisements.

Name: *Programmers Software Exchange*
Publisher: Programmers Software Exchange, 2110 N. Second Street, Cabot AR 72023
Interval: Quarterly
Price: $1.00
Format: 5½ × 8½
Comments: This is a catalog of programs, which are equally divided between the TRS-80 and the Apple, with a few for the PET thrown in. Programmers with software for sale can get their creations listed without charge. Also listed are public-domain programs that are available for a $5 copying charge each, or $49.95 for 100. The majority of software listed are games, but there are also some interesting applications programs here too.

The January-March 1979 issue contains an article on teaching the blind to type. A "call for articles" would indicate that other than just software listings are planned for future issues.

Name: *TRS Yellow Page*
Publisher: Micro Architect, 96 Dothan Street, Arlington MA 02174
Interval: Five times per year
Format: Eight pages—7 × 9
Comments: I don't know whether the November-December 1978 issue I have is the first issue or not. It is also hard to tell what direction this paper will take. Classified ads will be accepted, while the bulk of this issue is made up of advertisements for the publisher's software. Perhaps the next issue will show more clearly the publisher's intentions. Subscriptions are free.

Name: *CLOAD*
Publisher: CLOAD Magazine, PO Box 1267, Goleta CA 93017
Interval: Monthly
Comments: *CLOAD* differs from all of the other publications that I will discuss in this article. It is supplied on a C-30 cassette rather than being printed on paper. I have the October 1978 issue. *CLOAD* is strictly software-oriented and contains half a dozen programs each month. With *CLOAD,* you have no long listing to enter from the keyboard, just insert cassette and *CLOAD.*

Each feature (program) is recorded twice with Level I on one side and Level II on the other. The tape starts out with a Cover program that displays the name, date and a geometric design that keeps changing as the program runs.

Next comes a long Star Wars game that gives you a chance to fly the trench and drop a bomb down a ventilation shaft in an attempt to destroy Darth Vader and the Deathstar. Of course, while you are thus engaged, TIE fighters and laser guns alongside the trench are trying to destroy you! Add to that the fact that you have to get out quickly after you drop your bomb or the exploding Deathstar will take you with it, and you have a fast-moving, interesting game. A slow-paced mode is available for us beginners.

Graphic is a design-drawing program that is better than any that I've seen. The variety of designs is infinite; I find myself sitting for long periods of time fascinated by what is appearing on the screen.

Chase is a game that involves robots, electric fences and you. It has been around for quite a while, but this is the first BASIC version that I have seen. BASIC slows the game down considerably, especially when you elect to have more than one robot on the playing field, but it is still challenging and interesting.

If you have any money in the bank that draws daily interest, you will have a use for Passbook, which computes the daily interest on your savings. Enter the interest rate, the date and amount of each deposit, and Passbook will keep you up to date on your account.

The tape ends with a machine-language program called Machin, which fills the screen with the TRS-80 alphanumeric and graphics character sets. This is more of a demonstration routine than anything else since a simple BASIC program can be written that will accomplish the same thing.

Included with each month's tape are a few printed pages of editorial material that contain comments on CLOAD programs, TRS-80 software in general and announcements. Although program names and tape counter settings are printed on the tape label, I would like to see a table of contents either in the printed material or as part of the Cover program.

CLOAD is available by subscription, or individual issues, including all of the back issues, can be purchased. It is mailed first class. It does not contain any advertisements as yet, but would you be surprised if you loaded your tape some month and your screen announced, "This portion of *CLOAD* is sponsored by Superfluous Products, makers of the TRS-80 dust cloth and other related products?"

Mainly TRS-80

The following publications are not dedicated exclusively to the TRS-80 but contain much that will interest the TRS-80 owner.

Name: *Software Exchange*
Publisher: The Software Exchange, PO Box 55056, Valencia CA 91355
Interval: Bimonthly
Format: 32 pages—8½ × 11
Comments: This is another software-oriented magazine. The first three issues show a definite TRS-80 trend; each issue contains more TRS-80 software reviews and advertisements than the previous one. The main thrust is to review software and software-related books and periodicals. In addition, the editor and other insiders talk about software marketing, availability and implementation.

Classified advertising space is provided for both individuals and companies to offer for sale their creations or to inquire about programs of special interest in "wanted" ads. Letters from readers are also published.

Name: *Purser's List*
Publisher: Robert Elliot Purser, PO Box 466, El Dorado CA 95623
Interval: Quarterly
Format: 20 pages—8½ × 11
Comments: This is a list of software available for the TRS-80, PET and Apple computers. Edition 4 contains 15 pages listing vendors and programs. The printing is very small, so an enormous amount of software is listed. Individuals who have programs to trade are included. There is no charge to vendors or individuals who want to be listed. The list itself can be purchased in single copies or as a subscription. Advertising space is also available at a reasonable cost.

A unique feature of *Purser's List* is a photo section showing screen displays of various programs.

This is the best and most complete list of TRS-80, PET and Apple software that I have seen. Every owner of these three machines should have a copy.

Name: *On-line*
Publisher: Dave Beetle, 24695 Santa Cruz Hwy., Los Gatos CA 95030
Interval: Every three weeks
Format: 24 pages—6½ × 9
Comments: This is strictly a classified ad magazine, but it always contains quite a few ads relating to the TRS-80. It is printed with very small type and includes a large number of advertisements. It also has a very short lead time, so that ads received by the publisher up to four days before publication can appear. A long list of computer club meetings held all over the country is a feature of each issue. As space allows, the publisher includes new hardware and software announcements.

Conclusion

We will continue our objective review of TRS-80 publications next month in the second of this series of reviews. Each publication is intended to fill some need that the publisher thinks exists. It is clear in many cases that a good start has been made, while in a few cases it is too early to tell. I look for even better results as the readers and publishers become more familiar with this field. ■

realsoft™ THE PROGRAM STORE

(202) 337-4691

4200 Wisconsin Ave. NW P.O. Box 9609 Washington D.C. 20016

TRS-80
a trademark of Tandy Corporation

IF YOU'RE LOOKING FOR ANY TRS-80 SOFTWARE, GIVE US A CALL. THIS IS ONLY A SMALL SAMPLE OF OUR PRODUCTS. WE HAVE HUNDREDS OF TITLES IN STOCK. OUR PHONE IS ANSWERED 24-HOURS A DAY TO TAKE YOUR ORDER. SO ACT TODAY.

ADVENTURE 'GAME'

Adventures are like no other programs you have ever seen. They are "electronic novels" or "role-playing games." Whatever description you hear, it is inadequate until you "play" one. But here's an attempt.

In any version you move from one location to another. The locations can be as varied as caverns, islands, or rooms in a house. You take or use objects in these locations either immediately, or carry them with you for later use. For example, a key found in one place may unlock a door in another. By collecting and using these items you strive to accomplish your mission which differs with each adventure.

Also, you interact with the program using short sentences like TAKE KEYS, THROW AX, etc. Part of the fun is figuring out what words the program understands.

The "adventure" is figuring out the mystery - how to complete your mission. This makes adventure like a crossword puzzle, a mystery novel, and more. All require 16k, Level II.

adventure

from Adventure International

Scott Adams has authored seven different machine language Adventures for quick responses. They support the optional lower case hardware, have a unique screen video driver with blinking cursor for easy reading, and have over a 100 word vocabulary. They are highly recommended by 80-US, insiders, and 80 Software Critique.

1. ADVENTURELAND: Most like the original with caves, maze of pits, magic words and the dragon. $14.95
2. PIRATE'S ADVENTURE: "Yo Ho Ho and a bottle of rum..." Go from your London flat to Treasure Island searching for the pirate's buried treasure. $14.95
3. MISSION IMPOSSIBLE: "Your mission is..." Will you be able to complete your mission in time? Or is the reactor doomed? $14.95
4. VOODOO CASTLE: Count Cristo has had a fiendish curse put on him by his enemies. You're his only hope. $14.95
5. THE COUNT: Somewhere in Transylvania you awake in a large brass bed. Guard your neck. And watch for the postman. $14.95
6. STRANGE ODYSSEY: At the edge of the galaxy you find the ruins of an ancient alien culture and its treasures. $14.95
7. MYSTERY FUN HOUSE: Try to make it through the strangest Fun House before the park closes. $14.95

1&2 or 6&7 on disk for $24.95. 3,4&5 for $39.95. Other combinations available.

80-adventure

from Mad Hatter Software

Greg Hassett has created a series of four adventures. While they are written in BASIC, the response time is quick and they have about 40 rooms. Most enjoyable.

1. HOUSE OF 7 GABLES: Ring the bell, but you'll have to deal with the witch to leave. Check your score. 9.95
2. JOURNEY TO THE CENTER OF THE EARTH: Escape after you fix your earthdigger, but watch out for the bugs. $9.95
3. KING TUT'S TOMB: A true treasure hunt inside the cursed tomb. You'll be lucky to escape with your life. $9.95
4. SORCERER'S CASTLE: Kill the evil, wicked sorcerer before he captures you. Then try to leave the castle. $9.95

All four on disk for $35.00

dog star adventure

by Lance Micklus from TRS-80 Software Exchange

Aboard an enemy battlestar you try to rescue the princess, find the plans, retrieve the treasure and blast off. $9.95

treasure dungeon 2

by David White

Fight with the Savage Minataur, ghouls, trolls, and other monsters to get their guarded treasure. But know when to withdraw. $7.95

MAIL/FILE

from Galactic Software

A professional mailing list program requires support and thorough documentantion by the publisher. Galactic Software provides 30 pages of documentation in a three ring binder and updates to registered owners. And the file interfaces with their line of business packages.

The program will sort over 600 records on a single diskette in seconds! Not minutes. Not hours. Retrieval is in either alphabetic or zip code order plus other criteria. Labels are printed in either standard or unique user defined label formats. And an optional message line is available. Also, it supports both company names and international addresses.

Each record consists of name, address, phone, and category codes. With the proper codes, thousands of sublists are possible. And, editing is simple.

Mail/File is excellent for customer billing, newsletters, direct mail, collection agencies, clubs, and many other organizations. A complete package on disk for $99.00.

temple of apshai

from Automated Simulations

This first in the DunjonQuest(tm) series lets you take your hero into a magical and mythical labyrinth of over 200 rooms. You can encounter over 30 kinds of fearsome monsters who guard over 70 varied treasures. Some of the treasures are magical and can help you in exploring the underground complex, but look out for monsters and traps that spring at you from the walls and shadows of the rooms and passages you traverse. The Book of Lore fills in the background and describes the appearance of the temple as you go. You combat monsters, move, and grab treasure in real-time. Bring in characters from other fantasy role playing games, or let the innkeeper find thee a hardy fellow. Test your mettle against the servants of evil! For serious gamers. $24.95 with Book of Lore.

edtasm DISK★MOD PATCH

by Roy Soltoff from Misosys

This machine language program modifies your copy of the Radio Shack Editor/Assembler for use with your minidisk and disk operating system. You can save and load both text source and assembled object files. You can read the directory and the space used and available while in the EDTASM. You can also kill files. It is a complete disk modification for one or more drives.

Other capabilities are also added. The block move command relocates a section of text to any other area. The global change command permits, for example, changing a label throughout the text. The pagination feature provides hardcopy on 8 1/2 by 11 pages on either single sheets or continous paper. In addition, high memory can be reserved, like in BASIC, for machine language routines like printer drivers. You can also display the amount of memory remaining.

The <CLEAR> key is functional, the symbol table is sorted alphanumerically and output 5-across, the scroll up/down allows 15 lines on the screen, and the 'DEFM' assembly is improved. Lower case input is now permitted. Plus, you can branch to any address.

Upgrade your Editor/Assembler today. $19.95

ATERM

by Tom Stibolt from Acorn Software

The complete ASCII terminal program with the features you need and want: truly full duplex, completely compatible with Radio Shack's RS-232, all 128 ASCII characters from the keyboard, support of lower case if a modification is made, and BELL sound on AUX line from the computer.

You can set baud rate, parity, word length, and number of stop bits from the keyboard, even while receiving. Output to the lineprinter is buffered in the computer's remaining memory so printers slower than the RS-232 can be used. Completely compatible with Radio Shack's communications package. Level II with modem. $19.95

GSF

from Racet Computes

A collection of fast, easy-to-use machine language routines: in memory sort with multiple variables and keys, array read/write to tape; compress, uncompress and move data, screen scrolling in any direction, save screens, and more. Thorough documentation and two BASIC programs with multiple examples on the tape. Specify 16, 32 or 48k when ordering. $24.95

Credit card callers may phone us 24-hour a day. Or clip the coupon and mail your order today!

✔17

THE PROGRAM STORE
4200 WISCONSIN AVE NW
PO Box 9609 DEPT R
WASHINGTON, D.C. 20016

☐ YES, please send me these TRS-80 programs:

title	price
postage:	$ 1.00
total:	

name:

address:

city, state & code

☐ Check payable to The Program Store
☐ MASTERCHARGE mc bank code:
☐ VISA exp date:
card number:
signature:

The most common cause of 80 breakdown is the failure of the cassette relay—protect yours.

Relay Protection

Robert M. Richardson
PO Box 1065
Chautauqua Lake NY 14722

Here is a simple and inexpensive solution to what can be a very expensive problem when you use the TRS-80 microcomputer with a tape recorder other than the Radio Shack CTR-41. The contacts of the TRS-80's tape recorder remote control relay, K1 part #56-1051-10, are adequate to handle the nominal 500 mA at 6 V dc drawn by the CTR-41 tape recorder that comes as part of the TRS-80 system, but they cannot handle the larger current drawn by most other tape recorders.

Though the TRS-80 is extremely reliable and well designed, the Radio Shack computer service centers report that failure of the K1 relay contacts due to overload with recorders other than the CTR-41 is their most common servicing problem.

I use this battery-powered isolation relay (to operate a Radio Shack CTR-21 tape recorder as the second tape) unit with my TRS-80 Expansion/Interface, which controls two tape units. It works equally well with the TRS-80 barefoot. The relay contacts are rated at 1 Amp @ 125 V ac and will handle considerably more current if needed.

Total current drain from the 9-volt transistor radio battery is only 8 mA while the tape recorder is running. Battery life should approach shelf life of these batteries, which most Radio Shack stores give away free once a month to regular customers.

Layout is unimportant and may be as shabby as you wish (see Fig. 1). About the simplest and easiest configuration possible is to tape the relay to the bottom end of the 9-volt transistor radio battery with transparent tape, with approximately 2 inch leads from each end to the plug and jack, respectively. With this layout you simply plug the isolation relay/battery into the tape recorder of your choice, and the TRS-80 recorder remote plug into the isolation relay's jack.

There is no battery current drain when the tape recorder is not running, so it should last as long as battery shelf life. If it does not, do not forget those "freebie" Radio Shack batteries every month. ■

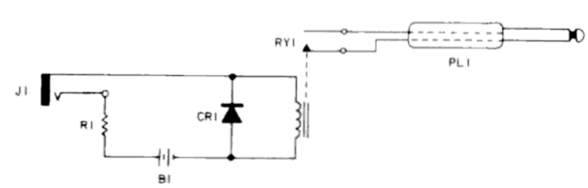
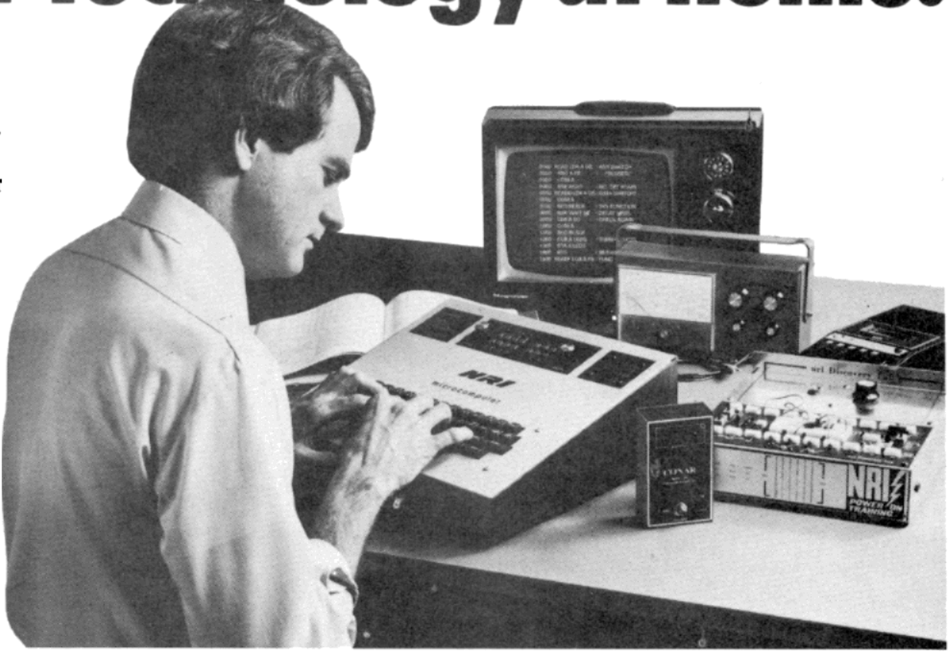


Fig. 1. Circuit diagram.

J-1: 274-333 subminiature phone jack
B-1: 23-464 9 VDC miniature battery
PL1: 274-289 subminiature phone plug
R-1: 390 ohms @ ¼ watt
RY-1: 275-004 6 VDC mini-relay @ 900 ohms
CR-1: any 12 volt PIV diode

Table 1. Parts list. You can probably substitute the RS275-003 12 V dc mini-relay by leaving out R-1. My junk box had a 6 V dc relay, so I used it.

Build your own microcomputer as you learn computer technology at home.



New from NRI! The First Interdisciplinary Home Study Course Ever Offered

As the microprocessor revolutionizes the computer world and microcomputers appear almost everywhere, NRI brings you a new, convenient, and effective way to keep up with this expanding technology. It's NRI's courses in Microcomputers and Microprocessors, created and designed exclusively for learning at home in your spare time.

Designed for the New Breed of Computer Technician

It's no longer enough to be just a programmer or technician. With microcomputers moving into the fabric of our lives as low-cost, easily available tools for business and home, both the programmer and technician must become total professionals. With practical knowledge of hardware, the programmer can design simpler, more effective programs. And with advanced programming skills, the technician can test and debug systems quickly and easily. The NRI course gives you simultaneous training in both skills...makes you one of this rare new breed.

Build Microcomputer, Test Instruments

NRI goes far beyond book learning to give you practical, "hands-on" experience. As you learn, you actually assemble NRI's designed-for-learning microcomputer. It performs like the finest of its kind, and features both assembly and basic language capabilities.

Every assembly step's a learning step. Using the NRI Discovery Lab® plus the NRI transistorized volt-ohm meter and CMOS digital frequency counter you also build, you perform meaningful experiments throughout your course...trace circuitry, interface components, introduce and correct problems, design your own programs, and more.

The Proven Way to Learn at Home

You don't have to worry with travel, classes, or time lost from work when you learn the NRI way. As they have for more than 60 years of teaching technical subjects, NRI brings the material to you. You study in your spare time, at your convenience, using "bite-size" lessons that program material into logical segments for easier assimilation. You perform experiments and build equipment using kits we supply. And your personal NRI instructor is always available for consultation should you have questions or problems. Over a million students have already shown the effectiveness of NRI training.

Choice of Courses

Several courses are available, depending upon your needs and background. NRI's Master Course in Microcomputers and Microprocessors starts with the fundamentals, explores basic electronics and digital theory, the total computer world, and the microcomputer. The Advanced Course, for students already versed in digital electronics, concentrates on software and the world of the microprocessor and microcomputer. In both courses, you build all instruments and your own computer.

Send for Free Catalog... No Salesman Will Call

Get the details on these exciting new courses in NRI's free, 100-page catalog. Shows all kits and equipment, lesson outlines, and full information, including facts on other electronics courses. Mail the coupon today and we'll rush your catalog. No salesman will ever call. Keep up with the latest technology as you learn on your own computer. If coupon has been removed, write to NRI Schools, Computer Department, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

NO SALESMAN WILL CALL
Please check for one free catalog only.

All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Amateur Radio • Basic and Advanced
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating Including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council

179-010

A step-by-step guide to working your fractions out.

Fraction Tutor

H. T. (Tom) Orr
249 Juanita Way
Placentia CA 92670

"Daddy, can you help me with my homework? We're studying fractions."

The voice was that of Sherri, my 9-year-old daughter. After looking at her schoolwork and seeing some of the difficulty she was having, I realized that, just as the computer had helped her learn the multiplication tables, the computer could help her with fractions.

TRS-80 Tutor

Rather than just have the computer ask her how much 1/2 plus 1/3 was and tell her whether she was right or wrong, I felt that the computer program should take her through the fractional addition process step-by-step and check for errors along the way.

The program accomplishes the objective in the following manner:

1. It first provides two random fractions to be added.
2. It then asks the user to determine the lowest common denominator and tells if the response is right or wrong.
3. The program next asks the user to convert the first fraction into the new denominator and checks for accuracy.
4. It does the same thing with the second fraction.
5. It next asks the user to add the two fractions and it checks for accuracy.
6. Should the total be greater than one, it assists the user in converting it to the whole number plus the fraction.
7. If the final answer is not in its lowest terms, it asks the user to convert it and checks for accuracy.
8. Most important, according to Sherri, it keeps a record of the problems worked and the number of errors.

The program uses the random number function to generate two fractions with different denominators. Asking the user to add 1/3 plus 1/3 at the program's beginning would be too simple for the program's purpose.

The program checks to be sure that the random fractions have the numerator smaller than the denominator. For nine-year-olds, I decided that the largest denominator would be 10. This keeps any multiplication required to convert to a lowest common denominator in the range most people can do in their heads.

The FOR-NEXT loop in lines 300 to 330 determines the two fractions' lowest common denominator. The lowest number divisible by both denominators will be the final value of I. This will be compared with the value entered by the program user to determine if the denominator selected is, indeed, the lowest.

The portions of the subroutines starting at line 1000 and 2030 determine if the fractions are at their lowest terms. This prohibits starting or finishing with fractions such as 2/4 or 4/6.

Results

Sherri took to the program like a duck to water. She considers it much more fun than doing similar problems with pencil and paper. Her proficiency with fractions has increased immensely, and you should see her smile when she has worked 20 problems correctly with zero errors.■

The author's nine-year-old daughter, Sherri, finds the use of a computer makes learning fractions fun.

```
                         FRACTIONS

WELCOME TO THE FRACTION PROGRAM.  I WILL GIVE YOU TWO
FRACTIONS TO ADD.  I WILL HELP YOU GO THROUGH THE
STEPS OF ADDING THEM.
PRESS ENTER WHEN READY TO START?


                    3/1Ø      +      1/4
WHAT IS THE LOWEST COMMON DENOMINATOR?  4Ø
WRONG, TRY AGAIN? 2Ø
CORRECT
OK, NOW YOU'VE GOT THE LOWEST COMMON DENOMINATOR.
NEXT, LET'S CALCULATE THE ANSWER.
3/1Ø     =      X/2Ø
X= HOW MANY?6
OK SO FAR
1/4 = Y/2Ø
Y=HOW MANY?5
OK SO FAR
OK, NOW LET'S PUT THEM TOGETHER.
6/2Ø      +      5/2Ø      =     Z/2Ø
HOW MANY IS Z? 11
VERY GOOD, THE ANSWER IS 11/2Ø
YOU DID THAT WITH 1 ERRORS.
THAT WAS 1 PROBLEMS WITH A TOTAL OF 1 ERRORS.
TRY AGAIN (1=YES,Ø=NO)?
```

Sample run.

```
5 REM **** FRACTIONS PROGRAM
7 REM H T(TOM)ORR, W6HT, 249 JUANITA WAY, PLACENTIA,CA 92670
10 CLS
20 O=0:R=0
25 Q=0:P=0
30 PRINT"                    FRACTIONS"
40 PRINT:PRINT"WELCOME TO THE FRACTION PROGRAM. I WILL GIVE YOU TWO"
50 PRINT"FRACTIONS TO ADD.  I WILL HELP YOU TO GO THROUGH THE"
60 PRINT"STEPS OF ADDING THEM."
70 INPUT"PRESS ENTER WHEN READY TO START";A$
190 CLS
197 R=0:L=0
200 REM FIRST FRACTION
210 GOSUB 2000
220 W=A:X=B
230 REM SECOND FRACTION
240 GOSUB 2000
250 Y=A:Z=B
255 IF Z=X THEN 200
260 REM PRINT FRACTIONS
270 PRINT"          ";W;"/";X;"   +   ";Y;"/";Z
280 REM CACLULATE LOWEST COMMON DENOMINATOR
300 FOR I=2 TO 100
320 IF (I/X)=INT(I/X) THEN 350
330 NEXT I
340 END
350 IF (I/Z)=INT(I/Z) THEN 370
360 GOTO330
370 REM I IS LOWEST COMMON DENOMINATOR
380 INPUT"WHAT IS THE LOWEST COMMON DENOMINATOR";E
390 IF E=I THEN PRINT"CORRECT":O=O+1:GOTO420
395 IF (E=I*2)+(E=I*3)+(E=I*4)+(E=I*5) THEN 2120
400 INPUT"WRONG, TRY AGAIN";E:R=R+1:GOTO390
420 PRINT"OK, NOW YOU'VE GOT THE LOWEST COMMON DENOMINATOR."
430 PRINT"NEXT, LET'S CALCULATE THE ANSWER."
440 PRINTW;"/";X;"     =   X/";I
450 C=(I/X)*W
460 INPUT"X=HOW MANY";D
470 IF C=D THEN  PRINT"OK SO FAR":GOTO490
480 PRINT"WRONG, TRY AGAIN":R=R+1:GOTO440
490 PRINT Y;"/";Z;"   =    Y/";I
500 INPUT"Y=HOW MANY";F
510 G=(I/Z)*Y
520 IF F=G THEN PRINT"OK SO FAR":GOTO540
530 PRINT"WRONG, TRY AGAIN":R=R+1:GOTO490
540 PRINT"OK, NOW LET'S PUT THEM TOGETHER."
550 PRINTD;"/";I;"    +    ";F;"/";I;"     =      Z/";I
560 INPUT"HOW MANY IS Z";H
570 IF H=D+F THEN PRINT"VERY GOOD, THE ANSWER IS";H;"/";I:GOTO600
580 PRINT"WRONG, TRY AGAIN.":R=R+1:GOTO550
600 IF H>I THEN 650
610 IF H=I THEN 700
620 U=H:GOTO1000
650 PRINT"THIS IS THE SAME AS 1 AND HOW MANY /";I;
660 INPUTU
670 IF U=H-I THEN PRINT:PRINT"RIGHT ON.":GOTO1000
680 PRINT"WRONG, TRY AGAIN.":R=R+1:GOTO650
700 INPUT"THIS IS EQUIVALENT TO WHAT";V
710 IF V=1 THEN PRINT"RIGHT ON.":GOTO800
800 PRINT "YOU DID THAT WITH";R;" ERRORS."
802 Q=Q+1
805 P=P+R
806 PRINT"THAT WAS";Q;"PROBLEMS WITH A TOTAL OF";P;" ERRORS."
807 INPUT"TRY AGAIN (1=YES, 0=NO)";T
810 IF T=1 THEN 190
815 PRINT:PRINT"**********************************************"
820 PRINT"THANK YOU FOR PLAYING WITH ME. SEE YOU SOON"
830 END
1000 REM CHECK TO SEE IF REMAINDER IS AT LOWEST TERMS
1010 FOR J=2 TO 100
1020 IF(U/J)=INT(U/J) THEN 1080
1030 NEXT J
1035 IF L=1 THEN 1500
1040 GOTO800
1080 IF (I/J)= INT(I/J) THEN 1100
1090 GOTO1030
1100 U=(U/J):I=(I/J)
1105 L=1
1110 GOTO1010
1500 PRINT"THE REMAINDER IS NOT AT LOWEST TERMS."
1510 INPUT"THE NUMERATOR SHOULD BE";M
1520 IF M=U PRINT"THAT'S RIGHT":GOTO1550
1530 P."WRONG, TRY AGAIN":R=R+1:GOTO1510
1550 INPUT"THE DENOMINATOR SHOULD BE";M
1560 IF M=I PRINT"THAT'S RIGHT":GOTO800
1570 PRINT"WRONG, TRY AGAIN":R=R+1:GOTO1550
2000 REM SUBROUTINE TO PICK UP FRACTION
2010 A=RND(9):B=RND(9)+1
2020 IF A>=B THEN 2010
2030 REM TEST FOR FRACTIONS NOT IN LOWEST TERMS
2040 FOR I=2 TO 9
2050 IF (A/I)=INT (A/I) THEN 2080
2060 NEXT I
2070 RETURN
2080 IF B/I=INT(B/I) THEN 2100
2090 GOTO2060
2100 A=A/I:B=B/I
2110 GOTO2040
2120 PRINT"THAT'S A COMMON DENOMINATOR, BUT NOT THE LOWEST."
2130 PRINT"SO I WILL HAVE TO SAY......."
2140 GOTO400
```

To get a printout of this program it was necessary to convert the code to Level II. This listing is a straight conversion of Level I; it has not been modified to actually *run* on a Level II machine. When typing the program for Level I use the following procedure:

1. Use the abbreviated form of each command
2. PRINT@ becomes P.A. or P.AT
3. PRINT# – 1 becomes P.#
4. INPUT# – 1 becomes I.# or IN.#

Program listing.

✔ Reader Service—see page 147

Only 150 bytes needed for this simple renumber program, sans bells & whistles.

Basic BASIC Renumbering

James Orleff
1433 1/2 Charles St.
LaCrosse WI 54601

Most of us, at one time or another, have needed to insert just one more line into a program in which all line numbers have already been used. By using the short program given here, Level II TRS-80 users will be able to do it themselves without waiting for Radio Shack's machine-language version.

The program is easy to use, but because it must be typed in each time it is to be used, it is short and limited only to renumbering the actual line numbers. It requires only about 150 bytes. It will not correct references to lines made within program statements. Still, correcting a few GOTO and GOSUB statements is easier than retyping the whole program.

Operation

To use the routine, simply type it in as shown (see Fig. 1) and use a RUN 10000 command to jump to it. Depending on the length of your program, it will take up to 30 seconds to run and will terminate with an undefined line error as the program renumbers itself and breaks the loop. When you list the program the lines will be separated by ten, except every 25th line, which will be separated from the next by only six.

To understand the program you must first know how Level II BASIC keeps track of its lines. All programs start at location 17129 with the byte at this location being the low byte of the address for the next line. Note that all PEEK commands return decimal values, so this is the decimal value of the low byte of a two-byte address.

The next location is the high byte, followed by the low byte of the actual line number, then the high byte. The next byte is the beginning of the actual text of the statement, which ends at zero. The whole process starts over again at this point with the new line, the address of which was pointed to by the previous one and the contents of that address being the low byte of yet the next one.

In my program, the location of the first line 17129 is initialized in C (for current line). The contents of this location, when added to 256 times the high byte, give the decimal address of the next line. This is stored in N (next line). Next we must alter the current line number in line 10040 by POKEing the initialized value of 10 into the low byte with the variable L. Next, H, which Level II will have set to zero, is loaded into the high byte.

The low counter is increased by 10 in line 10060, where you can increase or decrease the interval between renumbered lines. A test is made in line 70 to see if the low byte is greater than 250, since the largest decimal value that can be expressed in one byte is 255. If it is greater, we set the low byte to zero and increment the high byte counter by one. Line 80 tests to see if the high byte has reached 255 yet and, if so, prints an error message and halts execution. If not, the current line counter is set equal to the next line counter and loops back to the beginning.

Using this information, someone wishing to renumber only a few lines of their program should be able to use the PEEK and PRINT CHR$ functions to locate the bytes containing the line numbers and alter them as he or she wishes. For example, the short program of Fig. 2 will display decimal address, decimal value and character if printable.

By looking for the text of the line you wish to renumber, you can easily locate the line number, which will be the two bytes immediately before the beginning of the text. Simply POKE the desired new line number into these locations in the command mode. Divide the decimal number by 256 and POKE the integer result into the second or high location. The remainder will be what you POKE into the low address. ■

```
10000 L = 10
10010 C = 17129
10020 N = PEEK(C)
10030 N = N + (PEEK (C+1) * 256)
10040 POKE C+2, L
10050 POKE C+3, H
10060 L = L+10
10070 IF L > 250 THEN L = 0 : H = H+1
10080 IF H = 255 THEN PRINT "TOO MANY LINES" : END
10090 C = N
10100 GOTO 10020
```

Fig. 1. Renumbering program.

```
10 CLS
20 N=17129
30 PRINT N, PEEK(N), CHR$(PEEK(N))
40 N=N+1
50 GOTO30
```

Fig. 2. Dump Memory program.

PROFESSIONAL

INCOME TAX PROGRAMS FOR TRS-80™

Accountants, lawyers, tax consultants nationwide, prepared over 100,000 1978 Federal tax returns using our system.

Displays and fills in Form 1040 and related schedules on the screen, then prints out the completed forms automatically.

Change your mind? Make an error? Correct a single entry and you have a brand new form with all re-computations made automatically.

No tax system, running on any computer anywhere, has all the features of our professional system, and yet—

Our base program, which does 1040 and Schedule A costs only

$189.95

And! You can add schedules for only $37.95 each, customizing your system to your requirements.

DEMONSTRATION CASSETTE $3.95
(with sample forms)

Requires 32K System, 2 Disk Drives

We also have available fan-fold Forms 1040, blank stock, and other supplies you will require—all engineered and tested for use with TRS-80.

FREE CATALOG AND BROCHURE TO PROFESSIONALS

CONTRACT SERVICES ASSOCIATES ✓10

706 SOUTH EUCLID **ANAHEIM, CA 92802**

TELEPHONE (714) 635-4055

★ ★ ★ 20 YEARS OF SERVICE ★ ★ ★

Typed NEW by mistake? Don't panic!
Learn how to reclaim your programs, plus . . .

NEW Restored

Ken Fordham
6704 Preston Ct.
Tampa, FL 33615

The TRS-80 Level II machine is a useful and powerful computer. And it's utility is enhanced if you know more about what goes on inside it. As a proud owner of a new Radio Shack wonder, I wanted to find out what made it run and what I could do to expand its capabilities. Among other things, I'll tell you how to RESTORE to anywhere you want and also how to recover a program after you've inadvertently typed NEW.

Try An Experiment

First let's talk about statement format. Since you're probably sitting at the keyboard already (like I am most of the time), we'll do a little experiment. Type NEW, then enter the following statement:

10 STOP

Now from the keyboard do a series of PEEKs at the first eight addresses of the program storage area. As the Level II memory map shows, the first address of the program storage area is 17129. So if we PEEK starting here we find:

17129 = 239
17130 = 66
17131 = 10
17132 = 0
17133 = 148
17134 = 0
17135 = 0
17136 = 0

The first two addresses form an address that points to the beginning of the next statement. Addresses in the TRS-80 are stored in the standard Z-80 format. That is, the first part is the least significant half and the second part is the most significant half. In this case, then, 66 would be the most significant. To convert this to a recognizable address, multiply the most significant part by 256, then add in the least significant half. Following this procedure we have:

$$66 \cdot 256 + 239 = 17135$$

So address 17135 is the address where the next statement will begin (if there is one). More about this later.

The next two addresses form the statement line number. This has the same format as discussed previously for memory addresses. In this example it's easy to see the line number, but for numbers above 255 we can perform the same calculations as before to see the line number.

Let's skip over the next address (17133) for a minute. Address 17134 contains a zero which is used as the end of line

```
4FE1 21ED42  LD HL, dddd      ;set up memory pointer
4FE4 01FFFF  LD BC, dddd      ;set up dummy byte counter
4FE7 AF      XOR A            ;clear A
4FE8 EDB1    CPIR             ;search memory for first end of line
                              ;  indicator (0).
4FEA 22E942  LD (addr), HL    ;restore "next line pointer" in first statement
                              ;
                              ;
                              ;
4FED 23      INC HL           ;increment to point to MSB of "next line pointer"
4FEE 7E      LD A, (HL)       ;
4FEF A7      AND A            ;used only to set CPU flags
4FF0 2807    JR Z, dis        ;is MSB of "next line pointer" = 0
4FF2 46      LD B, (HL)       ;no, following instructions get next pointer
4FF3 2B      DEC HL           ;
4FF4 4E      LD C, (HL)       ;
4FF5 60      LD H, B          ;
4FF6 69      LD L, C          ;
4FF7 18F4    JR, dis          ;go back to 4FED to keep looking
4FF9 23      INC HL           ;increment to LSB of next available line number
                              ;  space
4FFA 22F940  LD (addr), HL    ;restore "next available line" pointer
4FFD C3191A  JP addr          ;go back to BASIC
```

Listing 1. FIXNEW—a program to restore a BASIC program after NEW is typed.

indicator. The last two addresses (17135,17136) are where the next statement would start if there was one. If this is the last statement in the program, Level II sets these addresses to zero to indicate that there are no more statements. If there is another statement, these two addresses will contain the pointer to the next statement just as 17129 and 17130 did for this example.

Now let's back up. All the addresses between the line number and the end of line indicator form the statement text space. In our example there is only one address, but typically there will be many. At this time it would be appropriate to take a look at Table 1.

Level II uses a condensed storage format for all of the statements and functions. In our example address 17133 contains 148. Looking at Table 1 we see that 148 is the code for a STOP statement. If our example had been something like:

```
10 FORA = 1TOB
```

there would have been six spaces between the line number and the end of line indicator. These six addresses would contain 129, 65, 213, 49, 189 and 66.

Looking at Table 1 and an ASCII chart we see that this area does indeed contain a combination of the codes from Table 1 and ASCII codes.

Clearing to Zero

We now have enough information to make ourselves dangerous. Let's look at the NEW command. I don't know everything that happens when this command is used, but I do know two things. Previously I thought this command wiped the program storage area slick. However, this is not the case.

Remember the first two addresses of each BASIC line? Well, when NEW is typed these two locations in the first statement are cleared to zeros. When RUN or LIST is typed and BASIC starts scanning the program storage area, it finds these two zeros and goes back to READY, assuming that no program is there. In most cases these two addresses are 17129 and 17130.

In addition when NEW is typed, BASIC maintains a "next available line" pointer. This is located at addresses 16633 and 16634. It points to the line number area of the next statement if one is to be entered. In the case of our first experiment, this pointer is set to 17137. When NEW is typed it is set back to 17131.

As far as I can tell these four addresses are the only ones that are changed when you type NEW. So if you type NEW and then realize you forgot to CSAVE your evening's programming effort, reset these two pointers to their previous values and you have your program back. This works fine if you haven't attempted any CLOADing before you restore them.

The hardest part of this procedure is finding out what these pointers should be so you can set their proper values. You could perform a series of PEEKs, using the information I've presented, and establish their value. But in a long program, this will take time. Another way is to PEEK at the four locations mentioned above in all of your programs and write them down to be used later. Make a habit of doing this when you're developing new programs.

FIXNEW

By far the easiest method uses the program in listing 1 called FIXNEW. This determines the proper values for the four previously mentioned bytes, sets the four addresses and goes back to the command mode of Level II jumping to address 6681. The program can be put on tape using TBUG. Since it is written to reside anywhere in memory, you can put it at the end of whatever memory you have.

If it becomes necessary to use FIXNEW you can load it with SYSTEM. It is not necessary to invoke the MEMORY SIZE? question. FIXNEW loads over the top of your string space so it is wiped out when you run the program you have just brought back to life.

After FIXNEW is loaded, use the / command to run it. It will restore your BASIC program and return to the command mode. If you have FIXNEW

128	END	171	LSET	214	<
129	FOR	172	RSET	215	SGN
130	RESET	173	SAVE	216	INT
131	SET	174	SYSTEM	217	ABS
132	CLS	175	LPRINT	218	FRE
133	CMD	176	DEF	219	INP
134	RANDOM	177	POKE	220	POS
135	NEXT	178	PRINT	221	SQR
136	DATA	179	CONT	222	RND
137	INPUT	180	LIST	223	LOG
138	DIM	181	LLIST	224	EXP
139	READ	182	DELETE	225	COS
140	LET	183	AUTO	226	SIN
141	GOTO	184	CLEAR	227	TAN
142	RUN	185	CLOAD	228	ATN
143	IF	186	CSAVE	229	PEEK
144	RESTORE	187	NEW	230	CVI
145	GOSUB	188	TAB	231	CVS
146	RETURN	189	TO	232	CVD
147	REM	190	FN	233	EOF
148	STOP	191	USING	234	LOC
149	ELSE	192	VARPTR	235	LOF
150	TRON	193	USR	236	MKI$
151	TROFF	194	ERL	237	MKS$
152	DEFSTR	195	ERR	238	MKD$
153	DEFINT	196	STRING$	239	CINT
154	DEFSNG	197	INSTR	240	CSNG
155	DEFDBL	198	POINT	241	CDBL
156	LINE	199	TIME$	242	FIX
157	EDIT	200	MEM	243	LEN
158	ERROR	201	INKEY$	244	STR$
159	RESUME	202	THEN	245	VAL
160	OUT	203	NOT	246	ASC
161	ON	204	STEP	247	CHR$
162	OPEN	205	+	248	LEFT$
163	FIELD	206	-	249	RIGHT$
164	GET	207	*	250	MID$
165	PUT	208	/	251	
166	CLOSE	209	↑	252	
167	LOAD	210	AND	253	.
168	MERGE	211	OR	254	!
169	NAME	212	>	255	ISA
170	KILL	213	=		

Table 1. Codes for Level II statement storage. Note that some codes cannot be used until you get a disk.

```
10 FOR A= 1 TO 9: READ B: PRINT B;: NEXT A: PRINT

20 FOR C= 1 TO 10: READ D: PRINT D: NEXT C: PRINT

30 RESTORE

40 FOR E= 1 TO 9: READ F: NEXT E

50 GOTO 20

100 DATA 100,300,200,900,500,600,800,400,700

110 DATA 0,1,2,3,4,5,6,7,8,9
```

Listing 2. Illustrates DATA RESTORE problem.

at the beginning of a tape, then the whole procedure shouldn't take more than a minute or two including the time it takes to find the tape.

A short description of the program is in order. The numbers are in hex. The instructions from 4FE1 to 4FE7 set up the proper registers for the CPIR instruction. 4FE4 sets up the B and C registers to a known value. These two registers aren't used, but the dummy value insures that B and C don't decrement to 0 before the CPIR has found the first 0.

The CPIR instruction searches memory from 17133 until it finds a match with the accumulator which in this case is 0 (the end of line indicator). Upon completion of CPIR the HL register pair has been incremented to one address beyond 0. The Hl pair is then put in 17129 and 17130 as the restored "next line pointer."

The instructions from 4FED until the end restore the "next available line" pointer. In the BASIC program text area the most significant half of the "next line pointer" will always be a value in the range of 66 to 255. Therefore, if we skip through memory looking at the upper halves of every "next line pointer," when we come to one that is 0 we know that the previous BASIC line was the last one and that this space is the "next available line." Just add one more to point to the least significant half of the line number space and put it in 16633 and 16634. The final step is to jump back to BASIC. FIX-NEW is short so I didn't spend much time trying to shorten it further.

DATA Manipulations

Have you ever had a program with many DATA statements and the first half of the program used the first half of the DATA, while the last half of the program used only the last half of the DATA? Let's suppose that the last half of the program reads through the last half of the DATA statements several times but the first half of the program only needs to read the first half of the DATA once. A RESTORE at the end of the DATA on the first pass through the program requires a series of dummy reads to get back to the beginning of the second half of the DATA. Remember, we no longer need the first half of the DATA after the first pass through the program.

Dummy READs are wasted time and add significantly to processing time. The program in Listing 2 illustrates the problem. This is just an example, so the number of DATA items is small, but imagine that there could be a hundred or more items in the DATA list!

Line 10 reads the DATA at line 100. Line 20 then reads the DATA at line 110. Hereafter we want only to read the DATA in line 110. So line 30 RESTOREs and line 40 does nine dummy reads so that the next DATA item that we read will be from line 110 again.

Wouldn't it be nice if the RESTORE statement had an argument option allowing it to RESTORE anywhere and not just to the beginning of the first DATA statement? Well we don't have any such option, but you can do the next best thing: use two POKEs to do a homebrew RESTORE. BASIC maintains a pointer at 16639 and 16640 to the next DATA item that will be read. Actually, it points to the comma before the next item to be read, so instead of doing a RESTORE, we can do two POKEs and set the pointer anywhere we want.

Listing 3 shows how this can be done. It's the same as Listing 2 except statement 15 has been added, 40 has been deleted, and 30 has been changed. After line 10 has read the DATA in line 100, the DATA pointer is now at line 110. Line 15 saves the value of the pointer for later use. In line 30 we can now simulate a RESTORE back to line 110, instead of line 100, by POKEing the saved pointer back in. That's all there is to it. Experiment with it awhile and you'll get the hang of it.

Hints and Tips

Normally, if you wanted to change your answer to the memory size question, you would have to turn the keyboard unit off and back on again. Well, if for some reason you don't want to turn it off, there is a way to get around this problem. Get the CPU to execute a jump to address 0. To do this type SYSTEM and then, in response to the *? prompt, type /O then ENTER. This brings you back to the memory size question, but because Level II goes back through it's initialization routine, any program in memory may get wiped out (and FIXNEW won't help).

If for some reason you need to know the cursor position in your program, you can use the POS(0) function to give you a number from 0 to 63 which is the horizontal cursor position, but this gives you no idea which line it is on. Level II has a cursor position pointer at addresses 16416 and 16417. To find the position as it relates to the video memory addresses, use the following statement:

```
A = PEEK(16417)•256 + PEEK(16416)
```

This will return a number between 15360 and 16383, inclusive.

To find a number from 0 to

```
10  FOR A= 1 TO 9: READ B: PRINT B;: NEXT A: PRINT
15  P1= PEEK(16639): P2= PEEK(16640)
20  FOR C= 1 TO 10: READ D: PRINT D;: NEXT C: PRINT
30  POKE 16639, P1: POKE 16640, P2
50  GOTO 20
100 DATA 100,300,200,900,500,600,800,400,700,
110 DATA 0,1,2,3,4,5,6,7,8,9
```

Listing 3. Illustrates solution to RESTORE problem.

```
10  POKE 16526,0: POKE 16527,79
20  X= USR(0)
100 POKE 16526,8
110 Y= USR(0)
```

Listing 4. Multiple USR(x) calls without Disk BASIC. Assume machine language routing starting at 20224 and another of 20232. Line 100 uses only one POKE because it's necessary to change only the LSB's of the address in this example.

1023 that can be used later in a PRINT@ statement, use the following:

A = (PEEK(16417)-60)•256 + PEEK(16416)

The Level II USR(X) command gives you an easy way to branch to a machine language routine. The Level II manual states that there is only one allowable USR call in Level II. Well this is not strictly true — you can have as many calls as you like. All you have to do is POKE the starting address for the routine you want into 16526 and 16527.

Later, when you want to use a different routine, POKE the new starting address and then use USR(X). Refer to listing 4 for an example.

Try developing a short routine at a starting address that looks at the argument you have passed and then jumps to one of the other routines based upon the argument. If you have ten routines then you would pass an argument of any number from 0 to 9 which would represent the routine that you wanted. Use PEEKs and POKEs to pass a computational argument, if needed.

The Blinking Cursor

The control codes listed on page C/1 of the Level II manual can be used to provide a blinking cursor among other things, which draws your attention to the area of the screen where input is needed. This can only be done with the INKEY$ function, as shown in Listing 5.

Basically, you turn on the cursor, delay a little, turn off the cursor, delay a little, and start it all over again. During the delays you do an INKEY$. The example is a program that allows you to enter exactly ten characters and then stops. Lines 10-50 form the loop that calls the subroutine ten times, prints each character you typed, then stops. Lines 1000-1040 are the subroutine that blinks the cursor and RETURNS when a key is pressed. Line 1000 turns on the cursor — 1020 turns it off.

After each of these is a line that forms a delay loop with the INKEY$ function embedded in the loop. When a key is pressed the subroutine is immediately exited. To alter the blink rate, change the maximum loop count in each of the loops. I'm sure there are many ways a blinking cursor could be produced so if you decide to do it your way keep in mind that the INKEY$ function must be executed often in order to produce an output each time a key is pressed.

One thing you may find interesting is that when the cursor is positioned to a spot on the screen where there is already a character, that character will alternate with the cursor — when the cursor is on, the character is off and vice versa. During the time that the cursor is being displayed, the ASCII code for the character is kept in the Video Display Control Block at address 16418 (see Level II manual page D/1).

All the ideas presented were checked out on my 4K Level II machine. Also it was assumed that TRSDOS was not being used. Have fun.■

```
10 FOR I= 1 TO 10

20 GOSUB 1000

30 PRINT A$;

40 NEXT I

50 END


1000 PRINT CHR$(14);

1010 FOR A= 1 TO 50: A$=INKEY$: IF A$<> "" THEN RETURN ELSE NEXT

1020 PRINT CHR$(15);

1030 FOR A= 1 TO 50: A$=INKEY$: IF A$<> "" THEN RETURN ELSE NEXT

1040 GOTO 1000
```

Listing 5. Blinking cursor example.

GIVE YOUR TRS-80 A LITTLE CHRISTMAS PACKAGE. (YOU'VE EARNED IT!)

Graphic Drawing Done On The TRS80 Using "The Electric Artist" Program!

GRAPHIC-TREK "2000" This full graphics, real time game is full of fast, exciting action! Exploding photon torpedoes and phasers fill the screen! You must actually navigate the enterprize to dock with the giant space stations as well as to avoid klingon torpedoes! Has shields, galactic memory readout, damage reports, long range sensors, etc! Has 3 levels for beginning, average, or expert players!

INVASION WORG Time: 2099 Place: Earth's Solar System Mission: As general of earths forces, your job is to stop the Worg invasion and destroy their outposts on mars, venus, saturn, neptune, etc! Earth's Forces: Androids — Space Fighters — Lazer Cannon — Nuetrino Blasters! Worg Forces: Robots — Saucers — Disintegraters — Proton Destroyers! Multi level game lets you advance to more complicated game as you get better!

STAR WARS Manuever your space fighter deep into the nucleus of the Death Star! Drop your bomb then escape via the only escape route. This graphics game is really fun! May the force be with you!

SPACE TARGET Shoot at enemy ships with your lazer gun. If they eject in a lifeboat vehicle, capture them, or if your cruel, destroy them! Full graphics, real time game!

SAUCERS This fast action graphics game, has a time limit! Can you be the commander to win the distinguished cross! Requires split second timing to win! Watch out! The saucers fight back!

Package One — $12.95

CHECKERS 2.1 Finally! A checkers program that will challenge everyone! Expert as well as amateur! Uses 3-ply tree search to find best possible move. Picks randomly between equal moves to assure you of never having identical games. Computer also makes sly remarks about the game!

POKER FACE The computer uses psychology as well as logic to try and beat you at poker. Cards are displayed using TRS-80's full graphics. Computer raises, calls, and sometimes even folds! Great practice for your Saturday night poker match!

PSYCHIC Tell the computer a little about yourself and he'll predict things about you, you won't believe! A real mind bender! Great amusement for party's.

TANGLE MANIA Try and force your opponent into an immobile position. But watch out, their doing the same to you! This graphics game is for 2 people and has been used to end stupid arguments. (And occasionally starts them!)

WORD SCRAMBLE This game is for two or more people. One person inputs a word to the computer while the others look away. The computer scrambles word then keeps track of wrong guesses. Can you make less mistakes then your adversary?

Package Two — $12.95

POETRY This exciting and sometimes funny program lets you choose the subject as well as the mood of the poem you want. You give TRS-80 certain nouns or names, then mood, and he does the rest! TRS-80 will write different poetry about one subject forever if you let him. Has a 1000 word + vocabulary of nouns, verbs, adjectives, adverbs!

ELECTRIC ARTIST Written by a working artist and a computer engineer to make drawing on the video screen easier then picking up a pencil! (Less messy too!) Manual: Draw, erase, move as well as, Auto: Draw, erase, move. Uses graphics bit's not bytes. Graphic face in ad done using this ingenius program!

GALACTIC BATTLE The Swineus enemy have long range phasers but cannot travel at warp speed! You can, but only have short range phasers! Can you blitzkrieg the enemy without getting destroyed! Full graphics — real time!

WORD MANIA Can you guess the computers words using your human intuitive and logical abilities? You'll need to, to beat the computer! He keeps score and gets mad if he loses!

AIR COMMAND Real time, graphics Flight Simulater! Land, take-off, get out of a spin! Be careful not to stall! Watch your fuel gauge! Requires a clear headed pilot.

Package Three — $12.95

LIFE This Z-80 machine language program uses full graphics! Over 100 generations per minute make it truly animated! You make your starting pattern, computer does the rest! Program can be stopped and changes made! Watch it grow!

REAL TIME LANDER This full graphics simulater lets you pick what planet, asteroid or moon you wish to land on! The "Live" keyboard gives super response that gives you the feeling of being in command! Has 3 skill levels that make it fun for everyone.

GREED II Multi-level game is fun and challenging! Beat the computer at this dice game using your knowledge of odds and luck! Computer keeps track of his winnings and yours. Quick fast action. This game is not easy!

THE PHARAOH Rule the ancient city of Alexandra! Buy or sell land. Keep your people from revolting! Stop the rampaging rats and locusts'. Requires a true political personality to become good!

ROBOT HUNTER A group of renegade robots have escaped and are spotted in an old ghosttown on mars! Your job as "Robot Hunter" is to destroy the pirate machines before they kill any more settlers! Exciting! Challenging! Full graphics!

Package Four — $12.95

SUPER HORSERACE! Make your bets just like at the real racetrack! 8 horses race in this spectacular graphic display! Up to 9 people can play! Use's real odds but has that element of **chance** you see in real life! Keeps track of everyone's winnings and losings. This is one of the few computer simulations that can actually get a room of people cheering!

MAZE MOUSE The mouse with a mind! Computer generates random mazes of whatever size you specify then searches for way out! The second time thru he'll always go fastest route! A true display of artificial intelligence! Full graphics, mazes & mouse!

AMOEBA KILLER You command a one man submarine that has been shrunken to the size of bacteria in this exciting graphic adventure! Injected into the presidents bloodstream, your mission is to destroy the deadly amoeba infection rampaging his body!

LOGIC This popular game is based on mastermind but utilizes tactics that make it more exciting and challenging! Has 2 levels of play to make it fun for everyone.

SUBMARINER Shoot torpedoes at the enemy ships to get points. Fast action graphics, arcade type game is exciting and fun for everybody!

Package Five — $12.95

20 HOME FINANCIAL PROGRAMS Did you ever get a loan and wonder if they figured interest or payments correctly? Or did you ever want to see what your payments would be if you borrowed x amount at x% interest over x years! Figures amortization, anuitys, depreciation rates, interest tables, earned interest on savings and much, much more. This program will get used again and again. A must for the conscientious, inflation minded person.

Package Six — $12.95

Home Computer Software For Everyday Applications

SIMUTEK

✓19

Exceptional Software thru Research & Imagination

Send Check, Money Order or Bank Card No. to:

SIMUTEK
P.O. Box 35298
Tucson, AZ 85740

Please Add 2.75 Postage & Handling

24 HOUR HOTLINE
(602) 882-3948

Same Day Shipment on Bank Cards & Money Orders

All Tape Programs Require a Minimum of 16K Level 2

Packages Available on Diskette (32K System) $5.00 Extra

TRS-80 IS A REGISTERED TRADEMARK OF TANDY CORP.

3 or More Packages Get 10% Discount

Use your 80 to analyze the horses; who knows you may even make a killing at the track!

Tout I

Charles J. Wilson
539 Spring House Lane
Camp Hill PA 17011

PSSSST! Hey, buddy! You wanna hot one in the fifth?" Those of you who have visited a racetrack may have heard these or similar words coming from the shadows next to the betting windows. The speaker is the tout, the turf consultant, the seller of "inside" information. The tout is one source, although highly unreliable, of information that you could use in making your betting decisions. I think the TOUT described in this article will prove to be of more value.

I'm the type who visits the track three or four times a year. Even though wagering on the ponies is a recreation rather than a vocation with me, I still like to take an analytical approach to my selection-making. I usually examine past statistics, make some computations and march with confidence to the $2 window, shrugging off the obviously inferior opinions of others as I go.

Because of my interest in horse betting, one of my first programming projects after picking up my TRS-80 was to automate my handicapping computations. Working within the constraints of a Level II, 4K machine, I wrote TOUT I. This handicapping program uses some basic concepts of horse selection.

The Program

Although the program is written in TRS-80 Level II BASIC, I think that an examination of the code in the Program listing will convince you that the program can be modified easily for use on any of the popular microcomputers.

TOUT I uses two criteria to compare the horses in a race. The first is how much money a horse brings home each time it runs a race. This is computed by dividing the amount of money won by the horse by the number of races it has run. This criterion is based on the rationale that a horse with a higher dollar winning per race than another horse either has been running in races with higher purses (and, thus, has been successful against stiffer competition) or has been relatively more successful in races of similar purses.

The second criterion used by TOUT I is a finish factor, which is similar to the grade-point average received by a college student. Each win is given a grade of 3, each second place finish a grade of 2 and each third-place finish a grade of 1. Any finish lower than third place is given a grade of 0. For example, a horse that has run in ten races—winning two, finishing second in three and third in four—has a finish factor of (2•3

Program listing.

```
100 REM *TOUT I*  C. J. WILSON
105 CLEAR 200
110 MM = 11
120 DIM A$(MM),KW(MM),KP(MM),T(MM),R(MM),P(MM)
130 DIM RW(MM),RP(MM),PW(MM),PP(MM),RO(MM),PO(MM)
200 CLS
210 PRINT "TOUT I":PRINT
330 INPUT "NAME OF TRACK";C$
340 INPUT "DATA";D$
350 INPUT "WHICH RACE DO YOU WANT TO HANDICAP FIRST";NR
360 NH = 0
370 INPUT "NAME OF HORSE TO BE EVALUATED";A$(NH)
380 INPUT "NUMBER OF RACES RUN THIS PERIOD";MR
390 IF MR=0 PRINT "NO RACE HISTORY.  REJECT.":GOTO 370
400 INPUT "WINNINGS THIS PERIOD";W
410 INPUT "FINISHES THIS PERIOD";M1,M2,M3
420 KW(NH) = W/MR
430 KP(NH) = (3*M1 + 2*M2 + M3)/MR
440 INPUT "ANY MORE HORSES TO EXAMINE (YES/NO)";B$
450 IF B$="NO" THEN 490
460 IF B$<>"YES" THEN 440
470 NH = NH + 1 :GOTO 370
490 FOR J=0 TO NH: T(J) = KW(J): NEXT J
500 GOSUB 1000
510 FOR J=0 TO NH: RW(J) = R(J): PW(J) = P(J): NEXT J
530 FOR J=0 TO NH: T(J) = KP(J): NEXT J
540 GOSUB 1000
550 FOR J=0 TO NH: RP(J) = R(J): PP(J) = P(J): NEXT J
570 FOR J=0 TO NH: RO(J) = KW(J)*KP(J): T(J) = RO(J): NEXT J
580 GOSUB 1000
590 FOR J=0 TO NH: PO(J) = P(J): NEXT J
600 CLS
610 PRINT "RACE # ";NR;" AT ";C$;" ON ";D$
620 PRINT "RANKING BASED ON $$$$ WINNINGS"
630 PRINT "RANK","NAME",,"$WIN/RACE"
640 FOR J=0 TO NH
650   PRINT J+1;TAB(13)A$(PW(J));
655   PRINT TAB(49) USING "#####.##";KW(PW(J))
660 NEXT J
670 PRINT:PRINT:INPUT "HIT RETURN WHEN REVIEW COMPLETE";Z
690 CLS
700 PRINT "RACE # ";NR;" AT ";C$;" ON ";D$
710 PRINT "RANKING BASED ON FINISH FACTOR"
720 PRINT "RANK","NAME",,"FACTOR"
730 FOR J=0 TO NH
740   PRINT J+1;TAB(13)A$(PP(J));
745   PRINT TAB(48) USING "#.###";KP(PP(J))
750 NEXT J
760 PRINT:PRINT:INPUT "HIT RETURN WHEN REVIEW COMPLETE";Z
780 CLS
790 PRINT "RACE # ";NR;" AT ";C$;" ON ";D$
800 PRINT "RANKING BASED ON COMPOSITE SCORE"
810 PRINT "RANK","NAME",,"COMPOSITE SCORE"
820 FOR J=0 TO NH
830   PRINT J+1;TAB(13)A$(PO(J));
835   PRINT TAB(49) USING "#####.##";RO(PO(J))
840 NEXT J
850 PRINT:PRINT:INPUT "HIT RETURN WHEN REVIEW COMPLETE";Z
870 CLS
890 PRINT "PERSONALLY, I LIKE ";A$(PO(0))
900 FOR J=1 TO NH
910   IF RO(PO(J))<0.9*RO(PO(0)) THEN 950
920   PRINT "AND ";A$(PO(J))
930 NEXT J
950 PRINT:PRINT:INPUT "ANOTHER RACE TO HANDICAP (YES/NO)";B$
960 IF B$="NO" THEN 990
970 IF B$<>"YES" THEN 950
980 INPUT "WHICH RACE NEXT";NR: GOTO 360
990 END
1000 REM *SORTING SUBROUTINE*
1050 FOR J=0 TO NH
1060   MX = -999
1070   FOR J1=0 TO NH
1080     IF T(J1)<MX THEN 1110
1090     MX = T(J1)
1100     JS = J1
1110   NEXT J1
1120   R(JS) = J
1130   P(J) = JS
1140   T(JS) = -9999
1150 NEXT J
1160 RETURN
```

```
TOUT I

NAME OF TRACK? AQUEDUCT
DATE? 26 DECEMBER
WHICH RACE DO YOU WANT TO HANDICAP FIRST? 8
NAME OF HORSE TO BE EVALUATED? THE PRINCE'S PANTS
NUMBER OF RACES RUN THIS PERIOD? 11
WINNINGS THIS PERIOD? 55970
FINISHES THIS PERIOD? 4,3,0
ANY MORE HORSES TO EXAMINE (YES/NO)? YES

        .        .        .
        .        .        .
        .        .        .

NAME OF HORSE TO BE EVALUATED? CREATOR
NUMBER OF RACES RUN THIS PERIOD? 14
WINNINGS THIS PERIOD? 41751
FINISHES THIS PERIOD? 5,1,2
ANY MORE HORSES TO EXAMINE (YES/NO)? NO
```

Fig. 1. Data input sample.

+ 3•2 + 4•1)/10, or 1.6. The larger the finish factor, the closer to the front the horse has been finishing.

These two criteria are not the only ones that can be used in handicapping a race. Another that comes immediately to mind is how fast the horse can make it around the track. The fastest horse wins (usually). However, TOUT I will be concerned with only the two criteria discussed above. There will be no need for you to take your stopwatch to the track in the predawn hours to clock training sessions and workouts.

Data Input

The program begins by requesting the name of the track at which the race is taking place, the date of the race and the number of the race. This information is used for tagging the output. Then, for each horse to be evaluated, four pieces of data must be input:

- The horse's name: A$();
- The number of races in which the horse has competed over some time interval: NR (I use the number of races run in the current year unless this number is less than eight. In that case, I use data from both the current and previous year.);
- The dollar winnings over the time period: W;
- The horse's finishes (first, second, third) over the time period: M1, M2, M3;

All information required by the program is easily obtained from the *Daily Racing Form.* You will be prompted by the program as each data element is needed.

Once all the input information has been entered, the computations begin when you respond "NO" to the question "ANY MORE HORSES TO BE EXAMINED (YES/NO)?" First, the money won per race is calculated. Next, a weighted finish factor is computed using the equation in the third paragraph under the "Program" heading. Two rankings are then made—one by dollars per race and one by finish factor. The dollars per race and finish factor are multiplied together to get a composite factor. A third ranking is made using this composite factor.

Each of the three rankings will be displayed in turn with each ranking remaining on the screen until you indicate that your review is complete. After the three listings have been displayed, TOUT I will present you with its own opinion, and the handicapping of the race is complete. TOUT I picks the horse with the top composite score plus any horse with a composite that is 90 percent or better of the top score. Additional races may be handicapped by responding affirmatively when asked.

The constant MM on line 110 defines the amount of memory reserved for the data arrays—MM being one less than the number of horses that can be evaluated. Those of you with more than 3284 bytes of available memory can increase MM and, thus, permit more than 12 horses to be handicapped in each race. However, if you have trouble eliminating all but the 12 top horses in a race, perhaps you should confine your wagering to your state lottery.

An interesting sidelight of the program development is shown in line 655 (also on lines 745 and 835). I had a requirement to print an output with a certain format at a certain line position. The TRS-80 Level II BASIC manual discusses PRINT TAB(N)X AND PRINT USING X$;X but does not address the use of both in the same statement. I first tried PRINT USING X$;TAB(N)X and got a syntax error. Success was achieved with PRINT TAB(N) USING X$;X. This was another example of the leprechaun-lover's law in action.

You can increase the effectiveness of TOUT I by prescreening the horses. Eliminate any horse that has not raced in the past month or that has not finished in the top four in at least one of its last four races. If the race being handicapped is a claiming race, you should avoid any horse that has been claimed in either of its past two races. Maiden races—races in which none of the entrants (either male or female) has ever won a race—should be approached with caution because of the lack of performance data.

The Toteboard

At this point, you're probably muttering under your breath, "He does a lot of talking, but can he show results?" Let's take a look at a sample run. The input data shown in Fig. 1 and the out-

```
RACE #  8 AT AQUEDUCT ON 26 DECEMBER
RANKING BASED ON $$$$ WINNINGS
RANK              NAME                          $WIN/RACE
 1           THE PRINCE'S PANTS                  5088.18
 2           PRINCE ANDREW                       4738.20
 3           RING OF LIGHT                       3833.00
 4           SILENT JOY                          3006.17
 5           CREATOR                             2982.21
 6           BILLY REDCOAT                       1277.64

HIT RETURN WHEN REVIEW COMPLETE?

     ************************************************

RACE #  8 AT AQUEDUCT ON 26 DECEMBER
RANKING BASED ON FINISH FACTOR
RANK              NAME                          FACTOR
 1           SILENT JOY                          1.750
 2           THE PRINCE'S PANTS                  1.636
 3           CREATOR                             1.357
 4           RING OF LIGHT                       1.300
 5           PRINCE ANDREW                       1.133
 6           BILLY REDCOAT                       0.636

HIT RETURN WHEN REVIEW COMPLETE?

     ************************************************

RACE #  8 AT AQUEDUCT ON 26 DECEMBER
RANKING BASED ON COMPOSITE SCORE
RANK              NAME                          COMPOSITE SCORE
 1           THE PRINCE'S PANTS                  8326.12
 2           PRINCE ANDREW                       5369.96
 3           SILENT JOY                          5260.79
 4           RING OF LIGHT                       4982.90
 5           CREATOR                             4047.29
 6           BILLY REDCOAT                        813.04

HIT RETURN WHEN REVIEW COMPLETE?

     ************************************************

PERSONALLY, I LIKE THE PRINCE'S PANTS

ANOTHER RACE TO HANDICAP (YES/NO)?
```

Fig. 2. Data output sample.

Horse	$Win Race	Finish Factor	Composite Score	Actual Finish
The Prince's Pants	1	2	1	1
Prince Andrew	2	5	2	4
Ring of Light	3	4	4	3
Billy Redcoat	6	6	6	5
Silent Joy	4	1	3	2
Creator	5	3	5	6

Fig. 3. TOUT I rankings and actual results.

put data shown in Fig. 2 are based on information from the *Daily Racing Form* for the eighth race at Aqueduct on December 26, 1978. TOUT I's three rankings are summarized in Fig. 3 along with the actual outcome of the race. As you can see, TOUT I, using the composite score, picked the winner, The Prince's Pants. The second ranked horse finished fourth, but the third ranked horse finished second. Not too bad.

Don't get all excited. The program's not going to do this well for every race. (At least it hasn't in the past.) To compensate for TOUT I's occasional lapses, I've found the best strategy is to bet the two horses with the highest composite scores to win. I don't often triple my money but then, I seldom lose a bundle.

For those of you who are true gamblers and like the exotic bets, I've had some success with the following procedures:

Daily Double—the winners of two successive races (usually the first and second) have to be picked. I take the top two selections for the first race and the top two selections for the second race and couple them. For example, if the first race selections were A and B and the second race selections were C and D, I would bet the combinations: A&C, A&D, B&C and B&D.

Quinella—the first two finishers in a race must be picked with either horse finishing first. I take the top three selections and bet all combinations. For example, if the selections are A, B and C, the three required bets would be: A&B, A&C and B&C.

Exacta—The first two finishers in a race must be picked in the order they will finish. Same strategy as the Quinella except more combinations are required. The bets would be A&B, B&A, A&C, C&A, B&C and C&B; six bets in all.

Triple or Trifecta—the first three finishers in a race must be picked in the order they will finish. Same procedure as the Exacta except top four selections are used and 24 combinations are bet.

A big investment will be required to follow the suggested approach, but sometimes the reward is outstanding. In fact, TOUT I hit the Triple at Penn National on December 26, 1978, when the first, fourth and third selections finished in that order in the fifth race. The payoff was $329.90 for a $24 wager.

So the next time you're planning an outing to the track, go out and buy a *Daily Racing Form*, punch TOUT I into your machine, enter the required input and let the program pick your winners. Of course, in no way do I guarantee that you will become rich beyond your wildest dreams. However, I have found that TOUT I works better than picking horses by the colors of their jockey's silks. Good luck and let me know how you do.■

Find out what Radio Shack left out of your 80 and how to utilize the commands they didn't tell you about.

Hidden Codes & Missing Chips

Patrick and Leah O'Connor
DeVry Inst. of Tech.
3300 N. Campbell
Chicago IL

Radio Shack has put a lot into their TRS-80 microcomputer for a reasonable price. Trying to understand how this computer works at the machine-language and schematic level has occupied our time for the past several months. With the arrival of the T-BUG monitor and Level II BASIC, it became possible to PEEK and POKE around the memory. We have also used a copy of the schematic from Radio Shack and disassembled listing of Level I BASIC from Small Systems Software to piece together a series of facts and conjectures concerning the workings of the TRS-80.

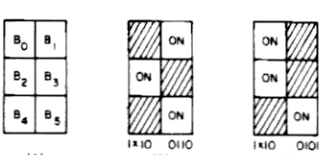


Fig. 1. Bit patterns for graphics characters. X indicates "don't care" condition of bit.

The Level II reference manual (page 8/5) gives the following information about the video display memory: "There are eight bits per byte . . . one is used to identify the byte as graphics or ASCII code . . . [and one] bit is not used. The remaining six bits contain either ASCII, graphics or control codes." You would assume from this that the video memory contains the normal eight bits per byte but only uses seven of them. Not so! Not only is one bit not used, it is not even there.

PEEKing and POKEing

The video memory is a 1K × 7 block located at 3C00 (hex). There is no memory for bit D_6. When data is being written into the video memory, bit D_6 is lost. When data is read out of the video memory, bit D_6 is created by NORing together bits D_5 and D_7. You can test this for yourself by POKEing an ASCII lowercase a into the video memory with a POKE.

If you use POKE 15360,97 and PRINT PEEK(15360) statements, an uppercase A will be displayed on the screen and the number 65, an uppercase ASCII A, will be returned by the print statement. In fact, this is exactly how we discovered what was going on. We tried to POKE the lowercase letters into the video memory, but uppercase letters were always displayed. We also found that POKEing either an FF(hex) or a BF(hex) enters the video memory and turns on all six sections of a graphics character. This would also seem to indicate that the D_6 bit of the video memory was not being used.

By POKEing various numbers into the video memory it was easy to see which bits controlled which portion of each graphics character. These results are shown in Fig. 1. We hate to admit how long we looked at the schematic before we realized that there were only seven 2102 chips in the video memory and that bit D_6 was actually the output of a 74LS02 NOR gate. This artificially produced bit is connected to both the main data bus and the character-generator address lines.

Fig. 2 is a block diagram showing the logic used to shift from the alphabetic to the graphics mode. When bit D_7 is LOW, shift-register 1 is strobed and the character generator can send data to the video section. When bit D_7 goes HIGH, shift-register 2 is strobed and the multiplexer sends graphics information to the video section.

The six bits control each portion of the the graphics character, and the multiplexer and shift-register provide a serial output to the video. Each alphabetic character is scanned on a line of the video raster as five dots wide, with an "undot" between, and the graphics are six dots wide, with no "undots."

Twelve scan lines complete a character, so some serial pulses are sent out at 12 separate times to complete each symbol. In the graphics, all 12 lines are used to form the graphics block, six rectangles (two rectangles wide by three deep); in the alphabetic mode, five of the scan lines are "unlines," leaving a character seven lines deep.

Table 1 shows why NORing together data bits D_7 and D_5 to produce data bit D_6 makes it impossible to address all of the 128 characters stored in the character generator ROM. Whenever bits D_5 and D_7 are both LOW, D_6 must be HIGH (otherwise D_6 will be LOW). Therefore, you will never see 0000 and 0001 in the four most significant bits of the video memory. Instead, you will have 0100 and 0101 in these locations. Likewise, if you try to POKE 0110 or 0111 into the four most significant bits of the video memory, it will be changed into 0010 or 0011.

Fig. 3 shows the contents of the character generator. Only the characters in lines 2, 3, 4 and 5 of Fig. 3 can be addressed. The contents of addresses 0 to 31 and 96 to 127 cannot be determined by software alone. The character gen-

	Normal Binary Sequence				Sequence With $D_6 = D_5$ NOR D_7				
	D_7	D_6	D_5	D_4	D_7	D_6	D_5	D_4	
0	0	0	0	0	0	1	0	0	4
1	0	0	0	1	0	1	0	1	5
2	0	0	1	0	0	0	1	0	2
3	0	0	1	1	0	0	1	1	3
4	0	1	0	0	0	1	0	0	4
5	0	1	0	1	0	1	0	1	5
6	0	1	1	0	0	0	1	0	2
7	0	1	1	1	0	0	1	1	3

Table 1. Modified addresses for video memory.

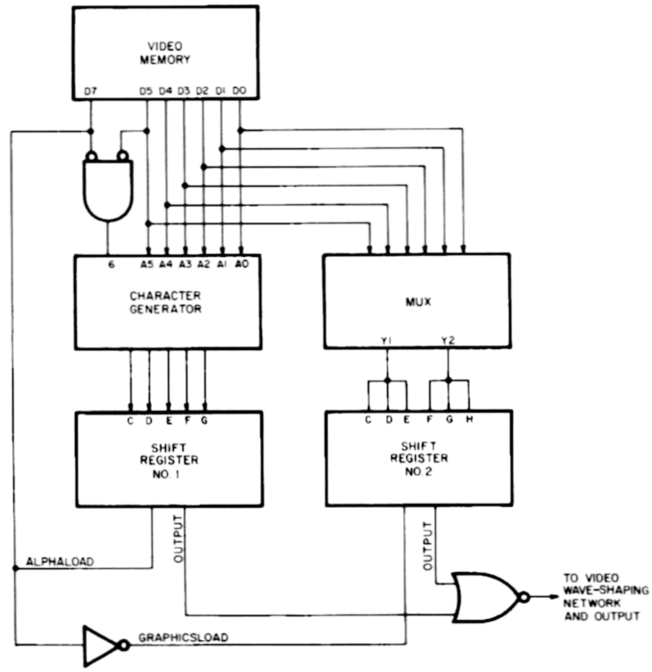


Fig. 2. Block diagram of logic-controlling selection of graphics vs alphabetic video mode.

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	↑	↓	←	→	_
6																
7																

Fig. 3. Contents of character generator ROM.

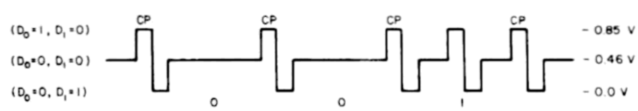


Fig. 4. Self-clocking code sent to tape cassette. CP indicates clock pulse.

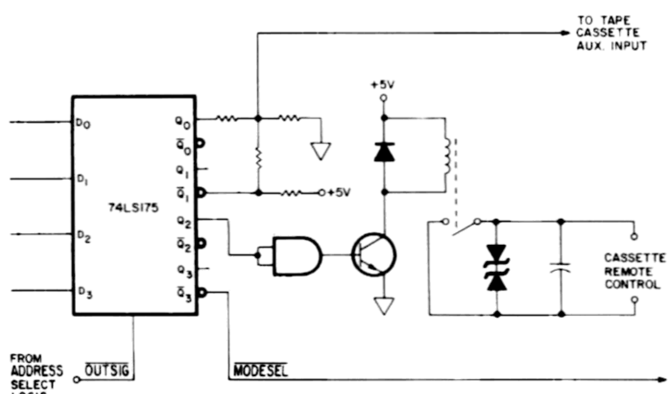


Fig. 5. Schematic for tape cassette output port.

erator (MCM 6670 P) is Motorola's mask-programmed version of their MCM 6674 character generator, which uses a fairly standard ASCII arrangement of characters.

In fact, the middle addresses of the MCM 6674 contain the same characters as the chip manufactured for the TRS-80. Maybe the other addresses are the same also. Since we have been told by several sources at Radio Shack that the TRS-80 can be modified to display lowercase letters on the video monitor if the graphics capabilities are sacrificed, the character generator must at least contain a lowercase alphabet.

Eliminating one 2102 chip from the video memory may have shaved $2 off the price of the video memory, but this is one place we wish they had not cut corners. The video memory cannot be used for anything else, and even with another 16K to use, this seems wasteful. If a full 8-bit byte were present in the video memory, it would be possible to make use of all 128 characters without losing the graphics. One of us (Pat) has had the nerve to open up the TRS-80, and we now have Dan Likins' 2102 kluge in our TRS-80.

Tape Format

One of the first questions we asked when we started looking closely at the TRS-80 concerned the tape format. We got a different answer from almost everyone we asked. It is not Kansas City Standard, but it is similar to it in some ways. It does not use start and stop bits for each byte of data, and it is not even an FSK (frequency shift keying) type of code.

The best answer seems to be that the TRS-80 uses a self-clocking code as shown in Fig. 4. For Level I BASIC, which writes tape at 250 baud, there is a train of pulses called clock pulses, which are separated by 1/250th of a second, or 4 milliseconds. For Level II BASIC, the clock pulses are separated by 2 milliseconds. Any pulse occurring between two clock pulses is read as a one-bit. If no pulse occurs between two clock pulses, it will be interpreted as a zero-bit.

The TRS-80 tape output port is noteworthy because of its simplicity. By utilizing software, for parallel-to-serial conversion and for the baud-rate generator, Radio Shack has reduced cost without sacrificing quality. (We have heard rumors that some big-name boards are put together with a 555 timer as the baud-rate generator!)

The TRS-80 uses a 10.6445 MHz crystal, divided down by six, which gives a very stable baud rate. The hardware for the tape output port consists of address-decode logic, a D-type latch, a passive wave-shaping network and a transistor-driven relay. The schematic is shown in Fig. 5.

Data bit D_2 is used to close the relay that goes to the tape deck's remote control input. To turn on the tape, a high D_2 bit must be written to output port FF (hex). When a byte containing a low D_2 bit is written to port FF (hex), the tape unit will be turned off. When data is read in from tape, it is also necessary to turn the tape on by outputting a byte with a high D_2 bit.

To output a signal to tape, data bits D_0 and D_1 are used. The software subroutine that puts out a clock pulse or a data pulse actually does three separate output operations. It outputs 0000 0110, pauses, outputs 0000 0101, pauses, outputs 0000 0100 and pauses for a third and final time. The complete subroutine requires 3303 clock cycles, or, at a clock frequency of 1.7741 MHz, 1.92 milliseconds. If a train of such pulses is written to tape in rapid succession, as would be done when writing FF (hex), the sound produced on tape will be very close to a C above high-C on the musical scale. This is exactly what a Level II tape sounds like.

Although data bit D_3 is not involved with writing to tape, it does perform an interesting function. When D_3 is HIGH, the $\overline{\text{MODE SELECT}}$ line causes the video monitor to change to the 32-characters-per-line, or "large-type" mode. It appears

✔ Reader Service—see page 147

Note to 80 MICROCOMPUTING SUBSCRIBERS

From time to time 80 Microcomputing makes its subscriber lists available to carefully-screened companies and organizations whose products, services or information may be of interest to you. In every case, we must approve all organizations wishing to mail to our subscribers. In every case they receive a list of names and addresses only—no other information we may have is divulged. Each organization may use the list only once and agrees never to make any personal or telephone solicitations from it.

The overwhelming majority of our subscribers appreciate this controlled use of our mailing lists—it helps them shop conveniently by mail for products and services they need, often at substantial savings. A few people prefer their names not be used. It is to them we address this message.

If you do *not* wish to be part of this service, please complete the form below . . . your name will not be used in this manner for any reason.

(If you asked us in the past to remove your name from our lists, there is no need to repeat the request.) Please allow about six weeks for your request to take effect.

☐ *I wish to have my name removed from the list of subscribers receiving mail other than the regular subscription to 80 Microcomputing.*

Mr.
Mrs.
Send to Ms.____________________
Please Print

Address____________________

City____________________ State__________ Zip__________

Mail this form with your mailing label from the latest issue or fill out the information as it appears on the label to:

80 MICROCOMPUTING
Subscription Services Dept.
P.O. Box 981 ● Farmingdale NY 11737

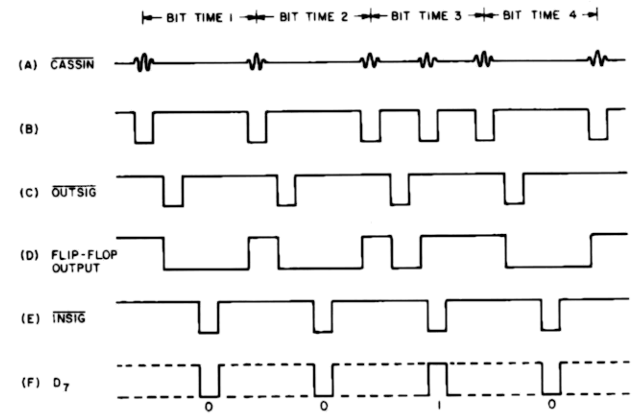


Fig. 6. Timing diagram for cassette input port. (a) OUTSIG resets flip-flop; (b) INSIG latches 0 from flip-flop to data bus; (c) Line B sets flip-flop to start bit-time 2; (d) Start of bit-time 3; (e) OUTSIG resets flip-flop; (f) Before next INSIG, line B sets flip-flop; (g) INSIG latches a 1 from the flip-flop to the data bus.

that a low level on the MODE SELECT line causes the system clock, Ø, to be divided by 2 before it goes to the video section. This means that if you have a T-BUG (so you can write in machine language), you should be able to get the large-size print on the Level I computer by writing a 0001000 to output port FF (hex).

The input port circuitry is a little more complicated than that for the output port because the tape recorder distorts the wave and adds a 60-cycle hum. Fig. 6a shows how the signal from the tape would appear under the most ideal circumstances. The TRS-80 is actually able to recognize a great variety of wave shapes. Fig. 7 shows some of the waves that were taken from prerecorded tapes and displayed on an oscilloscope. All these waveforms can be read equally well by the computer.

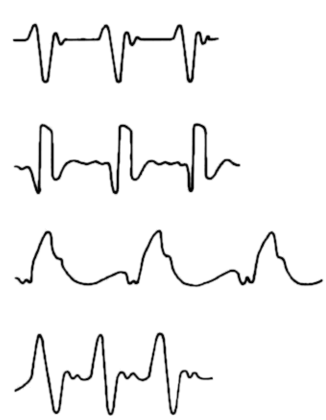


Fig. 7. Wave shapes from prerecorded program tapes. (Not to scale)

The active wave-shaping network shown as a single block in Fig. 8 consists of a high-pass filter to remove the 60-cycle hum, a quad op amp and other passive components. Line B of Fig. 6 shows the cassette input after it comes out of this wave-shaping network. This signal goes to the set-input of an active-low flip-flop. Each negative-going edge of line B changes the flip-flop output to 1. The flip-flop is reset by a signal from the OUTSIG line. The flip-flop output, Q, is connected to the D_7 bit of the data-bus by a Tri-state buffer, which is enabled by INSIG.

The software must turn on the cassette recorder by outputting a byte with the D_2 bit HIGH. The OUTSIG, which turns on the tape recorder, also resets the flip-flop and pulls its Q output LOW. After a short delay time determined by the software, an input instruction causes the INSIG line to go LOW. Whatever is currently on the flip-flop output will be clocked through the buffer and onto the data bus so it can be read by the CPU. During byte-times 1, 2 and 4 of Fig. 6, the flip-flop output remains LOW and a zero is put on D_7 of the data bus. During byte-time 3, there is a pulse present on the CASIN line between the two clock pulses. Therefore, the flip-flop will be set HIGH before

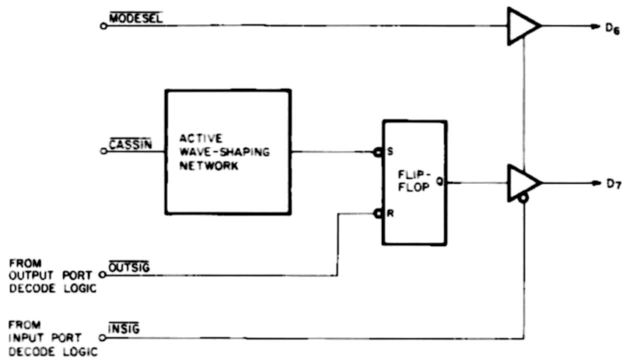


Fig. 8. Block diagram of tape cassette input port.

the $\overline{\text{INSIG}}$ line goes LOW. When the $\overline{\text{INSIG}}$ line goes LOW, 1 will be sent to D_7 of the data bus.

The $\overline{\text{INSIG}}$ line also clocks the $\overline{\text{MODE SELECT}}$ signal onto the D_6 bit of the data bus. This bit does not appear to be used by the tape input software. This design, however, allows other software routines to check the status of the $\overline{\text{MODE SELECT}}$ signal by inputting data from port FF (hex) and checking bit D_6. If D_6 is LOW, the video monitor is in the 32-characters-per-line mode. The input port FF (hex) and output port FF (hex) actually have two purposes each. Besides performing tape I/O operations, bits D_6 and D_3 allow the software to switch between the two sizes of type.

The general format of Level I and Level II BASIC tapes is quite similar. All tapes start with a 5-second leader of 0s, followed by a sync byte. When a CLOAD command is executed, the software reads through the leader until the sync byte is found. The next four bytes on Level I tapes contain the starting and ending addresses where the program will be loaded into memory.

Addresses

Level II BASIC allows for a one-byte filename, which precedes the starting and ending addresses. This feature allows for easy identification of specific programs on multiprogram Level II tapes. The filename feature is optional with the CLOAD command in Level II BASIC but is not optional with CSAVE. If the filename is omitted with the CSAVE command, an error message is printed, but not before the tape leader and sync byte are written to tape.

If you are not aware of this and try to repeat the CSAVE command with a proper filename, you will end up with a program with two leaders and two sync bytes. When this program is read-in, the second leader will be interpreted as the filename, starting address and ending address. It will, therefore, not load properly. This happened to us a few times before we realized what was going on.

The final byte of both Level I and Level II programs is a checksum byte. This is a sum of all the program bytes excluding the leader, filename, start and end addresses. If the checksum computed during a CLOAD is not the same as the last byte of the program, an error message will be printed on the monitor.

Table 2 contains addresses in memory where various subroutines and other useful information can be found. The comments column gives information about how the subroutines are to be used. For example, the comment about the keyboard scan routine includes: "Save DE and IY." This means that the subroutine will alter the contents of these registers, and it is therefore necessary for the programmer to save them on the stack before executing a CALL.

In some cases, the Level I and Level II subroutines for some functions are not comparable and have therefore been listed separately. For example, in Level I BASIC there is

ADDRESS Level I	Level II	COMMENTS
*	40A4, 40A5 L H	Locations where the low and high byte of the starting address are stored.
406C, 406D L H	40F9, 40FA L H	Locations where the low and high byte of the ending address are stored.
4200	42E9	Normal starting address of BASIC program.(address of first instruction)
4068	4020	Locations where the position of the cursor on the screen (it is between 3C00 and 3FFF) is stored.
0B40	002B	Keyboard-scan subroutine. Level I returns bit Z=1 if keyboard clear; Level II returns register A=00 if keyboard clear (no key pressed). Registers DE and IY will be changed; save if needed later.
0010	0033	Subroutine to display byte in register A at current cursor position on screen. (DE and IY must be saved)
0FE9	0212	Subroutine to turn on cassette. Reg. A, in Level II, must contain a '0' before call.
0EF4		Subroutine to turn on cassette, load into memory specified from tape, on return HL=ending address+1, Z=0 if checksum error occurred.
	0296	Subroutine to look for leader & sync byte.
	0235	Subroutine to read one byte from tape. Returns byte read in register A.
	01F8	Subroutine to turn off cassette. (Level II user must do own checksum)
0F4B		Subroutine to save memory on tape. Programmer must load starting address in HL and ending address+1 in DE.
	0287	Subroutine to write leader & sync byte.
	0264	Subroutine to output byte in reg. A to tape.
0000	0000	Entry point into BASIC upon power-up.
01C9	1A19	Reentry point into BASIC. (Monitor will display "READY".) Used to return control to BASIC interpreter after machine language.

Table 2. Subroutines and other information stored in memory.

one subroutine that will turn on the cassette, read the tape, load the tape into the memory at addresses specified by the tape header, compute the checksum and finally turn off the cassette. In Level II BASIC, these functions are done by several separate subroutines. This allows for more flexibility, but it also means that the programmer must do more work.

Most of the addresses given in Table 2 were found in the Radio Shack Editor/Assembler reference manual or the disassembled listing of Level I BASIC from Small Systems Software. We found the memory locations in which the starting and ending addresses of BASIC programs are stored.

```
10  KEY$ = CHR$(205) + CHR$(53) + CHR$(2)
20  FOR N = 0 TO 14334
30  T$ = CHR$(N) + CHR$(N+1) + CHR$(N+2)
40    IF T$ <> KEY$ GOTO 70
50      PRINT N
60      IF INKEY$ = "" GOTO 60
70  NEXT N
```

Program A. BASIC program to search ROM for a three-byte key.

First, we noted that they had to be somewhere between 4000 (hex), the beginning of RAM in Level II memory and 42E9 (hex), the normal beginning of Level II programs. We also knew that when the CLOAD command was executed, it would have to store the starting and ending addresses somewhere in RAM.

If we knew where the CLOAD routine was in ROM, we could dump it out, but we didn't know where it was located. We did, however, know where the read-a-byte subroutine was located. The address for this subroutine, 0235 (hex), was found in the Editor/Assembler reference manual. The CLOAD routine must call the read-a-byte subroutine many times. Therefore, we wrote a program that PEEKed its way through ROM, looking for the three-byte sequence 205, 53, 02 (decimal), which is the machine language for CALL 0235 (hex).

The Program

The BASIC program used is shown in Program A. Line 10 sets up a key equal to the three bytes we were looking for. The FOR loop in line 20 moves through the memory comparing each three consecutive bytes to the key. When a match is found, the address containing the CALL statement is printed on the monitor. Line 60 causes the program to pause so the address can be copied down.

After we found the area in ROM where the read-a-byte subroutine was called from, we dumped it out and disassembled it. There were only two addresses between 4000 (hex) and 42E9 (hex) where data read from the tape was stored. To test and see if these really were where the starting and ending addresses were stored, we wrote a short program and used T-BUG to see where the program ended. Sure enough, the ending address was the same as the two bytes starting at 40F9 (hex), with the low byte first and high-byte second. The two bytes starting at 40A4 (hex) always contained E9, 42 (hex), which is the normal starting address for Level II BASIC programs with the low-order byte given first.

We are sure that had we been a little bit more persistent, we could have gotten the above information by calling Tandy Advanced Systems in Fort Worth TX, but phone calls from Chicago to Texas cost money, and we have found you have about a fifty-fifty chance of getting to talk to someone who can answer your question. This time luck was against us—a hardware man answered the phone, but we had a software question. Three hours of work saved us a $2 phone bill. We don't know if we came out ahead or not, but at least it's a lot more satisfying to know that we did it ourselves. We also now have a technique that is quite useful for exploring the ROM.

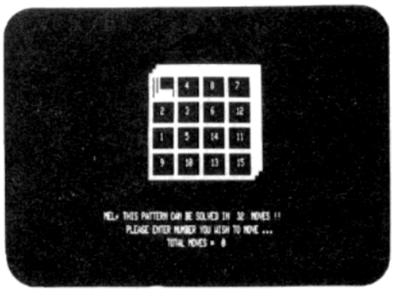
The ways Level I and Level II BASIC programs are stored in memory are quite different. Level I BASIC stores each character you type into a program line as a separate ASCII character. Thus, the word PRINT requires five bytes of storage, while the abbreviation P. requires only two bytes.

Level II BASIC allows for even more economical use of memory. Every Level II operator or instruction is stored in memory as a one-byte code. The code for a PRINT instruction is B2 (hex). Only when the program lines are printed on the monitor are the full words used.

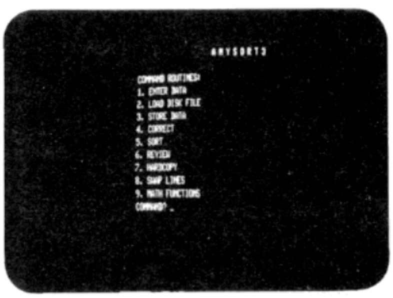
Breaking the code was quite

Software for the TRS-80

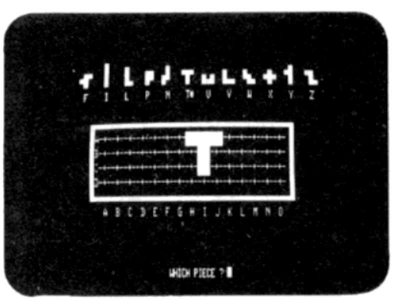
FIFTEEN NUMBERS $6.95



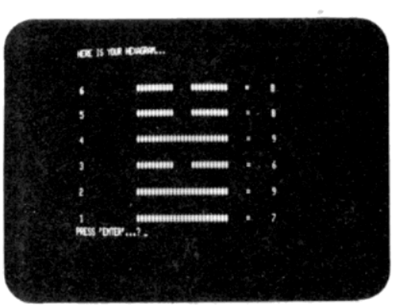
ANYSORT 3 $29.95



PENTOMINOES $9.95



THE I-CHING THING $6.95

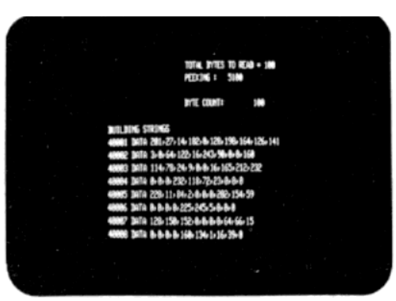


TRS-80 TIELINE

TRS-80 TIELINE is an extended smart terminal program. Functions supported with this package are the ability to send and receive BASIC data and programs. A fully supported set of timeshare ASCII control keys are software selectable. 'ESC' and a 'BREAK' key function do not require any handware modifications. Smart functions make it possible to jump from mode to mode with communication prior to program transmission or reception. Half and full duplex modes as well as line feed transmission or suppression, baud rate, parity, word length, stop bits are software selectable and can be changed while running. A printer can be connected for hard copy of communications as well as LLIST at baud rates that include 134.5 baud for certain serial printers. A special host or source mode allows other computers to use the TRS-80 TIELINE as a timeshare style computer. Programs can be run as well as disk files loaded, saved or transmitted by control from the distant computer. Character echo-back is supported. Host overide of forbidden commands is possible. An additional feature allows testing of the TRS-80/RS232 hardware. The program is available for DOS 2.0 - 2.2 machines with 32K memory, RS232 board and modem. Additionally, the package includes a free copy of 'THE TRS-80 DATA COMMUNICATION HANDBOOK' by Stephen Gibson. The handbook is a compilation of terms and in depth explanations of data communication from the standpoint of the user, the hardware, and the phone company. Various phone line services are detailed. Baud rates, bits and codes are explained at hobbyist level. How a Bell 103 modem works and how to connect it is covered with emphasis on originate and answer frequencies, duplex, half-duplex and RS232 conventions. The EIA standards are given with ASCII control character set information.

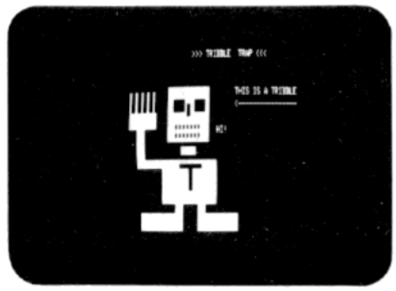
TIELINE $24.95
TRS-80 DATACOMM HANDBOOK $ 5.95

All orders include 3% postage and handling.
TRS-80 is a registered trademark of Radio Shack.
California residents add 6% Sales Tax
VISA & MASTERCHARGE accepted.

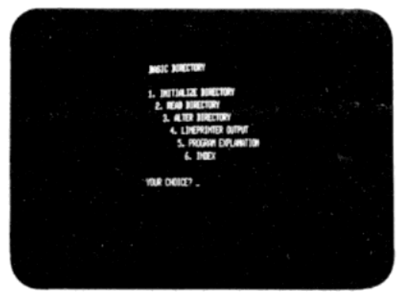


MACHINE TO BASIC $6.95

TIELINE $24.95



TRIBBLE TRAP II $6.95



DIR $14.95

ASTROLOGY II $15.95

PROGRAMMA

Software Products

PROGRAMMA INTERNATIONAL, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579
384-1116
384-1117

Dealer Inquiries Invited

Reader Service—see page 147

128	END	192	VARPTR
129	FOR	193	USR
130	RESET	194	FRL
131	SET	195	ERR
132	CLS	196	STRING$
133	CMD	197	INSTR
134	RANDOM	198	POINT
135	NEXT	199	TIME$
136	DATA	200	MEM
137	INPUT	201	INKEY$
138	DIM	202	THEN
139	READ	203	NOT
140	LET	204	STEP
141	GOTO	205	+
142	RUN	206	–
143	IF	207	*
144	RESTORE	208	/
145	GOSUB	209	↑
146	RETURN	210	AND
147	REM	211	OR
148	STOP	212	>
149	ELSE	213	=
150	TRON	214	<
151	TROFF	215	SGN
152	DEFSTR	216	INT
153	DEFINT	217	ABS
154	DEFSNG	218	FRE
155	DEFDBL	219	INP
156	LINE	220	POS
157	EDIT	221	SQR
158	ERROR	222	RND
159	RESUME	223	LOG
160	OUT	224	EXP
161	ON	225	COS
162	OPEN	226	SIN
163	FIELD	227	TAN
164	GET	228	ATN
165	PUT	229	PEEK
166	CLOSE	230	CVI
167	LOAD	231	CVS
168	MERGE	232	CVD
169	NAME	233	EOF
170	KILL	234	LOC
171	LSET	235	LOF
172	RSET	236	MKI$
173	SAVE	237	MKS
174	SYSTEM	238	MKD
175	LPRINT	239	CINT
176	DEF	240	CSNG
177	POKE	241	CDBL
178	PRINT	242	FIX
179	CONT	243	LEN
180	LIST	244	STR$
181	LLIST	245	VAL
182	DELETE	246	ASC
183	AUTO	247	CHR$
184	CLEAR	248	LEFT$
185	CLOAD	249	RIGHT$
186	CSAVE	250	MID$
187	NEW	251	
188	TAB(	252	
189	TO	253	.
190	FN	254	!
191	USING	255	ISA

Table 3. Level II codes for reserved words and operators. These are the values (in decimal) actually stored in memory for each of the words at the right of each number. Some of these words have no function in Level II BASIC; they are reserved for use in Level II Disk BASIC.

Address	Contents Hex	Comments
42E9	F3	Line Pointer
42EA	42	
42EB	0A	Line Number
42EC	00	
42ED	B2	PRINT
42EE	22	"
42EF	58	X
42F0	59	Y
42F1	22	"
42F2	00	
42F3	F9	Line Pointer
42F4	42	
42F5	14	Line Number
42F6	00	
42F7	80	END
42F8	00	
42F9	00	Line Pointer
42FA	00	

10PRINTXY
20END

tedious. We first wrote a simple program line such as 10 PRINT X and looked at how it was stored in memory. After deciding which address contained the code for PRINT, we POKEd other values into that address. When we changed the code from 178 (decimal) [Equal to B2 (hex)] to 128 (decimal) and listed line 10, it read 10 END X. By repeating this procedure, we were able to construct Table 3. This table is also found in ROM starting at address 1650 (hex).

If you examine it with a PEEK statement, you will see that the first letter of each word is missing. Since the highest-order bit of an ASCII character is always zero, this bit is used to indicate the beginning of a new word in the table. For example, the ASCII code for the word PRINT is 50, 52, 49, 4E, 54 (hex). Changing the high-order bit from a 0 to a 1 is the same as adding 80 (hex) to the byte. Therefore, it is actually found in the table as D0, 52, 49, 4E, 54 (hex). The reserved words found in the ROM are in the same order as the words in Table 3. The software apparently counts down through the table to find the ASCII characters that correspond to each code.

In Level I BASIC the line-number is stored as a 2-byte unsigned number with the low-order byte first. The line-number 10 (decimal) would appear in memory as 0A00 (hex), and the line-number 256 (decimal) would appear as 0001. The line-number is, naturally enough, the first two bytes of each program line as stored in the RAM. The last byte of each line is the ASCII carriage-return character 0D (hex).

Level II BASIC uses a more complicated method of delineating program lines. The two line-number bytes are always preceded by another two bytes, which are called the line-pointer in Level II BASIC. These two bytes contain the address (low-order byte first) of the first byte of the next line. In other words, each line-pointer points to the line-pointer of the following line. Table 4 shows the contents of memory for a program consisting of a PRINT statement and an END statement. The program contains only two lines, but there are three line-pointers. The last pointer is always zero. Notice, also, that Level II BASIC uses a zero for the last character of each program line instead of a carriage return.

Conclusion

This concludes the information we have been able to piece together so far. It should fill in some of the gaps left by the Radio Shack manual. Our next project is going to be a Level-II-to-Level-I conversion program (by the time it's finished, everybody will probably have Level II BASIC anyway) and a program to renumber BASIC program lines so that more than one program can be read into memory at the same time (yes, we know someone else has one for sale, but we'd rather do it ourselves).

We hope that what we have presented will give readers ideas for other programs.■

ATTENTION TRS-80'S

Why sit in the corner in the dark and turned off while your master is sitting by the light, turned on to 80 Microcomputing?
You need a magazine of your own for Education-Enlightenment-Enjoyment and for the personal satisfaction (you're a personal computer, aren't you?) of your very own possession . . . A Subscription to **CLOAD MAGAZINE!**
Turkey your master into sending a check (U.S.A.: $36.00, Overseas: $38.00 Surface Delivery — $48.00 Airmail) to the jive cats at **CLOAD MAGAZINE**. You will get 12 C-30 cassettes, one a month, each one filled with all kinds of juicy software — Games, Tutorials, Practical Programs and Impractical Trivia. All programs rated G for computers under 18 years old.

Do it! Subscribe Now!

CLOAD
MAGAZINE

Box 1267
Goleta, CA 93017
(805) 964-2761
Master Charge/VISA Welcome

"TRS-80 is a registered trademark of TANDY CORP." ✓32

© 1978 **CLOAD MAGAZINE**

T-BUG™ USER:

Super TLEGS: Onboard relocator for T-BUG, moves T-BUG to any desired RAM location. Now examine, modify, # P punch backup copies of formerly coincident program material, experiment with parallel kustom T-BUGs. Also will relocate TSTEP.
16K Level II Super TLEGS No. LL-0 — 9.95

TSTEP: Single-stepper for T-BUG. Displays **all** CPU aspects related to instruction set as you SPACE through ROM or RAM. Indispensible for debugging, analyzing alien program material or learning Z-80.

a) CPU registers in before/after # R-like format, user accessible, independent of T-BUG registers.
b) Testable flag status in before/after format.
c) Top stack elements in before/after format.
d) 8 key Implicit Keypad. Backspace, CLEAR, more.

Subroutines can be single-stepped or run directly, control remaining with TSTEP. Also, Super TLEGS will relocate TSTEP, making monitor and single-stepper into an independently relocatable unit. Confirm any code by **seeing** what you are imagining.
16K Level II TSTEP No. LL-1 — 11.95

EMU 02: Software emulation of the 6502 microprocessor. T-BUG displays byte, EMU takes it from there. Now you can write, debug, execute 6502 programs **on your TRS-80!**

a) Disassembler posts standard 6502 Assembly mnemonic next to T-BUG displayed byte, in scrolling field.
b) Single-stepper displays 6502 Processor Model in before/after form, expanded flag and stack elements, all updated after each instruction is SPACE-ed.
c) 4-speed TRACE mode animates 6502 Models, activates keyboard scan port accessible to 6502 instructions.
d) **Fast** interpretive RUN mode for realistic execution.
e) 13 key Implicit Keypad, Backspace, Relspace, more.

How to have a 6502 without having a 6502! Compare and contrast, work in a powerful programming language that is distinct from BASIC, Z-80. Read Apple, PET code.
16K Level II EMU 02 No. BL-1 — $24.95

.75 mailing each program. CA add 6%

ALLEN GELDER
Box 11721 Main Post Office
San Francisco, CA 94101

T-BUG, TRS-80 tm Radio Shack/Tandy Corp. ✓79

With these disks, I can turn your TRS-80 into a serious computer.

I'm Irwin Taranto, and I've put the first computer into more than 300 different businesses.

It's taught me that the TRS-80 is an elegant piece of hardware despite its low price. Given the right programs, it can jump through hoops.

Put simply, I have the right programs. Four of them are the genuine Osborne & Associates systems, originally designed for the $30,000 Wang computer. I've made a few minor modifications, and they now work on a $4000 TRS-80. The other two programs I added myself.

These programs are fully-documented, and you can buy the books locally or from me. I made them work on the TRS-80, and if you buy them from me, I'll make them work for you.

If you're not sure about that, call the number below and get the names of some of the people who've bought all over the world. Then ask them.

These programs cost $99.95 each. (The Cash Journal option on the General Ledger adds another $50.) That gets you the disk, all the instructions you need and my phone number. If you call, we answer all your questions. If your question's tough enough, I'll talk to you personally.

Because I plan to turn that TRS-80 of yours into a serious computer.

THE OSBORNE PROGRAMS

Accounts Payable: invoice-linked, it reports, does checks and links to general ledger.
Accounts Receivable: invoice-linked, it tracks invoices and aging, prints statement and links to general ledger.
General Ledger: handles 1750 transactions each on 200 different accounts. Cash journal option available.
Payroll: figures the pay, does the checks and all the bookkeeping.

AND MY OWN PROGRAMS

Inventory Control: gives immediate readout on any inquiry. It has many existing versions or can be individually tailored.
NEW! Invoicing: linked to accounts receivable. Prints invoices and feeds data into receivables.

✓45

Taranto
& ASSOCIATES

P.O. Box 6073, 4136 Redwood Highway, San Rafael CA 94903 · (415) 472-1415. Add $3 per order for handling, 6% sales tax in California only. If you don't already have the books, add $15 each (invoicing book, $10). Mastercharge, Visa OK.

More data to sort than will fit in memory? Use disk random access to solve the problem.

Sort 80K in 6K!

D. E. Fitchhorn
3504 Piermont Dr. N.E.
Albuquerque NM 87111

When your mailing list, inventory or other data file grows beyond reading the complete file into your computer memory, how do you sort the data? If you have a TRS-80 with Level II Disk BASIC, you are blessed with the capability of creating and editing large files, but that disk operating system leaves you only about 6K bytes of memory to hold programs and working data.

Now is the time to start making use of the random access disk file capability of Level II Disk BASIC. While random access disk files will take more planning in the beginning and usually waste more disk storage space than sequential disk files, they offer many advantages to the user.

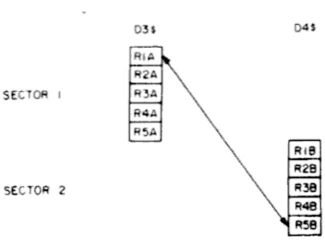


Fig. 1. Swapping records in different sectors.

Advantages

One attractive advantage of the random access disk file is that it can be sorted as if the complete file were committed to memory. A second advantage is that the memory used is constant, regardless of the size of the file. It is true that sorting disk records is slower than sorting in working memory, but can you hang 80K of memory on your system just to run a sort program? Even if you could address the memory, the cost would drive you right back to a handwritten card-index system.

Another advantage is that you don't have to read the file with exactly the same FIELD statement that was used to write it. You can read the sector as if it contained only one record and move through the string using MID$ functions to get to the desired information. I made use of this capability in order to make a general-use sort program.

A quick review of the structure of a random access disk file shows us that the file is composed of a number of major records. Each major record

```
ZZZZZZZZZZ  YYYYYYYYYY  XXXXXXXXXX  WWWWWWWWWW  VVVVVVVVVV
UUUUUUUUUU  TTTTTTTTTT  SSSSSSSSSS  RRRRRRRRRR  QQQQQQQQQQ
PPPPPPPPPP  OOOOOOOOOO  NNNNNNNNNN  MMMMMMMMMM  LLLLLLLLLL
KKKKKKKKKK  JJJJJJJJJJ  IIIIIIIIII  HHHHHHHHHH  GGGGGGGGGG
FFFFFFFFFF  EEEEEEEEEE  DDDDDDDDDD  CCCCCCCCCC  BBBBBBBBBB
AAAAAAAAAA  JJJJJJJJJJ  IIIIIIIIII  HHHHHHHHHH  GGGGGGGGGG
```

Example 1. Unsorted data.

```
AAAAAAAAAA  BBBBBBBBBB  CCCCCCCCCC  DDDDDDDDDD  EEEEEEEEEE
FFFFFFFFFF  GGGGGGGGGG  HHHHHHHHHH  IIIIIIIIII  JJJJJJJJJJ
KKKKKKKKKK  LLLLLLLLLL  MMMMMMMMMM  NNNNNNNNNN  OOOOOOOOOO
PPPPPPPPPP  QQQQQQQQQQ  RRRRRRRRRR  SSSSSSSSSS  TTTTTTTTTT
UUUUUUUUUU  VVVVVVVVVV  WWWWWWWWWW  XXXXXXXXXX  YYYYYYYYYY
ZZZZZZZZZZ  JJJJJJJJJJ  IIIIIIIIII  HHHHHHHHHH  GGGGGGGGGG
```

Example 2. Data sorted on the first 26 records.

```
AAAAAAAAAA  BBBBBBBBBB  CCCCCCCCCC  DDDDDDDDDD  EEEEEEEEEE
FFFFFFFFFF  GGGGGGGGGG  GGGGGGGGGG  HHHHHHHHHH  HHHHHHHHHH
IIIIIIIIII  IIIIIIIIII  JJJJJJJJJJ  JJJJJJJJJJ  KKKKKKKKKK
LLLLLLLLLL  MMMMMMMMMM  NNNNNNNNNN  OOOOOOOOOO  PPPPPPPPPP
QQQQQQQQQQ  RRRRRRRRRR  SSSSSSSSSS  TTTTTTTTTT  UUUUUUUUUU
VVVVVVVVVV  WWWWWWWWWW  XXXXXXXXXX  YYYYYYYYYY  ZZZZZZZZZZ
```

Example 3. Data sorted on the first 30 records.

takes one sector of the disk (256 bytes on the TRS-80). Thus, each time a major record is read from or written to the disk, a complete sector is read or written. For example, PUT 1,1 will write the first sector of the file, and PUT 1,25 will write the 25th sector of the file. GET 1,1 will read the first sector of the file, and GET 1,25 will read the 25th sector of the file.

The major records can be divided into subrecords when the number of bytes of data associated with a file element is a submultiple of the number of bytes in the sector. That is, if your data element takes only 25 bytes for a record, then you can fit ten subrecords into one major record (sector) on the disk.

The system will take care of reading and writing the major record, but you will have to make provision in your program to read and write the subrecords. For example, if you wish to work with the 25th subrecord of the file, you have to tell the system through the GET/PUT statements and the FIELD statements that you are talking about the third sector and the fifth subrecord of that sector.

Take heart, it is really simple once you get a feel for the structure of the file. Another important consideration is that each data element in each subrecord must contain the same number of bytes, and therefore is padded with blanks if the data does not fill the space allowed by the variables in the FIELD statement of the program.

How to Sort a Random Access Disk File

The program listing was written so that I could use it to sort any of my random access files. Since the data as read from the disk is stored as if it were one long string, we can sort as if all data were alpha and only have to consider conversion of numerics when the data to be sorted on is numeric.

The program allows defining the many elements associated with the sort, the size of the sector, the number of records with the sort: the size of the sector; the number of records; the size of a record; the number of subrecords in a major record; the number of characters to be used; and an offset if the sort data is not the first data in a subrecord.

The actual sorting used is Shell-Metzner (this can be replaced by any sort procedure without too much difficulty). Shell-Metzner does save some time even for this program.

The main problem in sorting disk files comes in keeping track of which sector and which subrecord the data comes from and which sector and subrecord the data should be written to. Both the disk reading/writing and the swapping routines must understand when they are working with data from the same sector and when they are not.

Since the data may come from different sectors, two strings (D3$ for one sector and D4$ for the second sector in the sort program) must be used. If the sectors containing the records to be swapped are different, then a swap is made of the record in one string (D3$) to the other string (D4$) (see Fig. 1). However, if the sectors are the same, then both records in both strings must be swapped (Fig. 2).

The accompanying examples show a test file in its unsorted condition (Example 1), then sorted on the first 26

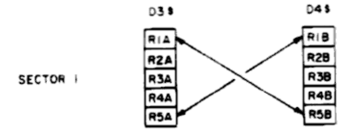


Fig. 2. Swapping records in the same sector.

records (Example 2) and finally sorted on all 30 records (Example 3).

If your files have few individual records per sector, it is wise to create an index file—in the form of a random access file—that is composed of the data to be sorted on and the original record number that the data comes from. This index file could then be sorted, and the rearranged order of the record numbers could be used to rewrite a new file in the order desired. Since the index file will have more records per sector, it will run faster.

While this program is written as a stand-alone general-use program, it can be easily trimmed to be attached to another program as a subroutine by replacing the question with defined variables and renumbering to fit into your program.■

Program listing

```
10 REM ***********************************
11 REM *** RANDOM ACCESS DISK FILE     ***
12 REM *** SORTING PROGRAM             ***
13 REM *** D. E. FITCHHORN             ***
14 REM *** ALBUQUERQUE, NEW MEXICO     ***
15 REM *** VERSION 1, JAN  1979        ***
16 REM ***********************************
17 REM
20 CLOSE 1: CLEAR 1000
30 REM ***********************************
31 REM *** ALLOW THE OPERATOR TO       ***
32 REM *** DEFINE THE SORTING          ***
33 REM *** REQUIREMENTS                ***
34 REM ***********************************
35 REM
40 INPUT"NUMBER OF RECORDS TO BE SORTED ";N5
50 IF N5 <= 0 GOTO 40
60 INPUT"NUMBER OF RECORDS IN A SECTOR ";N
70 IF N <=0 THEN 60
80 INPUT"NUMBER OF BYTES IN A RECORD ";N1
90 IF N1 <= 0 THEN 80
100 INPUT"NUMBER OF BYTES IN SORT FIELD ";N2
110 IF N2 <=0 THEN 100
115 INPUT"ALPHA OR NUMERIC SORT (A OR N) ";X$
116 IF X$ <> "A" AND X$ <> "N" GOTO 115
120 INPUT"OFFSET TO DATA IN RECORD ";O
122 IF O < 0 GOTO 120
130 REM
131 REM **********************************
132 REM *** OPEN FILE AND GET FIRST    ***
133 REM *** SECTOR                     ***
134 REM **********************************
135 REM
140 OPEN "R",1,"TSTFL":FIELD 1,255 AS D1$
150 GET 1,1:D3$=D1$:D4$=D1$:P8=1:P9=1
160 REM
161 REM **********************************
162 REM *** START OF SORT PROCEDURE    ***
163 REM **********************************
164 REM
170 M1 = N5
180 M1 = INT(M1/2):IF M1 = 0 THEN 530
190 K1 = 1:K = N5 - M1
200 P1 = K1
210 P2 = P1 + M1
220 P3 = INT((P1 - 1)/N)+1:P5 = P1 - N * INT((P1 -1)/N) -1
230 P4 = INT((P2 - 1)/N)+1:P6 = P2 - N * INT((P2 - 1)/N) -1
240 IF P8 = P3 GOTO 260
250 LSET D1$ = D3$:PUT 1,P8:GET 1,P3:D3$ = D1$
260 IF P9 = P4 GOTO 280
270 LSET D1$ = D4$:PUT 1,P9:GET 1,P4:D4$ = D1$
280 N6 = P5 * N1:N7 = P6 * N1
285 IF X$ = "A" GOTO 290
286 F1$ = MID$(D3$,1+O+N6,N2):F2$=MID$(D4$,1+O+N7,N2)
287 F1=CVI(F1$):F2=CVI(F2$)
288 IF F1 < F2 GOTO 500
289 GOTO 310
290 IF MID$(D3$,1 + O + N6,N2)<MID$(D4$,1+O+N7,N2) GOTO 500
300 REM
301 REM **********************************
302 REM *** SWAPPING ROUTINE           ***
303 REM **********************************
304 REM
310 T1$ = MID$(D3$,1+N6,N1):T2$ = MID$(D4$,1+N7,N1)
320 MID$(D3$,1+N7,N1)=T2$
330 MID$(D4$,1+N6,N1)=T1$
370 IF P3 <> P4 THEN 410
380 MID$(D4$,1+N7,N1)=T2$
390 MID$(D3$,1+N6,N1)=T1$
430 IF P3 = P4 THEN D3$ = D4$
440 LSET D1$ = D3$:PUT 1,P3:GET 1,P3:D3$ = D1$
450 IF P3 = P4 THEN D4$ = D3$:GOTO 470
460 LSET D1$ = D4$:PUT 1,P4:GET 1,P4:D4$ = D1$
470 P1 = P1 - M1
480 IF P1 < 1 GOTO 500
490 GOTO 210
500 K1 = K1 + 1
510 IF K1 > K GOTO 180
520 GOTO 200
530 CLOSE 1
540 END
```

Use your machine as an intelligent terminal for a larger system.

Smart Terminal

Jimmy D. Shirley
2105 Rosedale
Las Cruces NM 88001

Have you been dreaming of turning your TRS-80 into an intelligent terminal so you can talk to the big computers—right from the comfort of your own home? Well, it's not difficult to do if you have the right equipment, software to go with it and a little help. Let me give you the benefit of my experience in turning my TRS-80 into an intelligent terminal.

Introduction

When I first learned that Radio Shack would be coming out with an RS-232C serial interface board, I rushed down to the store and ordered one. With this addition and my Pennywhistle 103 modem, I could use my TRS-80 as an intelligent terminal. Although it took several weeks for the board to arrive (I ordered before the availability date), it was well worth the wait.

The board is well built and is easily installed in the expansion interface. It fits into the large space under the access plate on the top of the expansion interface on the left side. However, the arrangement for connecting the board to the socket in the interface did give me some trouble. The board does not simply plug in. Instead, it mounts on top of the 42 contact connector in the interface and is aligned and secured by two small machine screws and a plastic washer. As luck would have it, my board arrived without the washer. Even so, I managed to get the board installed and the cable hooked up in a few minutes.

Using my Pennywhistle 103 modem, I then had to load the Term program supplied by Radio Shack and find the proper settings of the modem switches and the DIP switches on the interface board. Initially, I set the modem switches to full duplex, low band and the serial interface board switches to 300 baud, 1 stop bit and seven data bits plus a parity bit (odd).

At these settings, absolutely nothing happened when I dialed the time-sharing system at New Mexico State University. This was a real surprise to me because this is the way I wired my Heathkit H9 terminal, which has no difficulty communicating with the computer at NMSU. But, it is not clear from the instruction manual or any markings on the board which way to flip the DIP switches for on and which way for off.

After consulting the schematic diagram in the manual and checking with an ohmmeter, I realized that I had set all the switches wrong. Correcting this error cleared up most of my problems; however, parity was apparently still wrong because vertical bars were displayed preceding each character received. A vertical bar is displayed if a parity error, framing error or overrun error is detected.

Incorrect parity seemed to me to be the most likely cause of the error indication. The H9 terminal does not check parity on data it receives, and, therefore, my original parity setting on it could have been wrong. Sure enough, switching to even parity got rid of the vertical bars.

Now, only one problem remained. Nothing I typed in was displayed. Radio Shack suggests connecting pin 2 to pin 3 on the DB-25 connector. This causes transmitted data to be echoed back to the UART receiver.

Later, on page 21 of the manual, they suggest setting the modem to half duplex. This is certainly a more convenient solution and does the same thing, although it is confusing to set the modem to half duplex just to get the keyboard output displayed. It occurred to me that a minor software change might be called for.

The BREAK Key

After using the Term program for a while, I began to come up with some ideas for other improvements. First, I found that I needed a way to interrupt the host computer. The need for this capability becomes clear the first time you receive a long-winded response and can't figure out how to stop it. Without a way to interrupt, you must wait for the transmission to finish before you can send. Of course, you can always uncradle the telephone from the modem and hang up, but that is rude, inconvenient and inelegant.

Many terminals have a BREAK key for this purpose. The BREAK key causes a continuous logic 0 signal to be sent as long as the key is depressed. This tells the computer to interrupt its transmission and allow the terminal to transmit. There is no provision in the Term program supplied by Radio Shack to send a break signal. However, the serial interface board does have the capability of sending a break signal under software control. Specifically, clearing bit D2 in the UART control register causes a break signal to be output. Thus, to break under software control, all you need is to add the appropriate software to the Term program (see the assembly-language listing).

There is a hardware solution also. My approach was to implement the hardware solution first because it is so simple and also because it is independent of the software. With this approach I have a BREAK key even when I use the unmodified Term program. I installed a BREAK key on the modem itself by using a normally open momentary contact switch to connect +18 volts from the modem power supply to pin 2 of the DB-25 connector in the modem. The RS-232C standard defines a positive voltage greater than 3.0 volts and less than 25 volts as a logic zero.

Specifically, in both the Radio Shack interface and the Pennywhistle modem, logic zero is represented by +12 volts with respect to signal

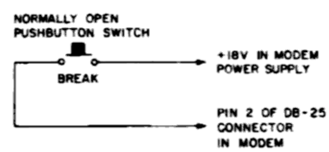
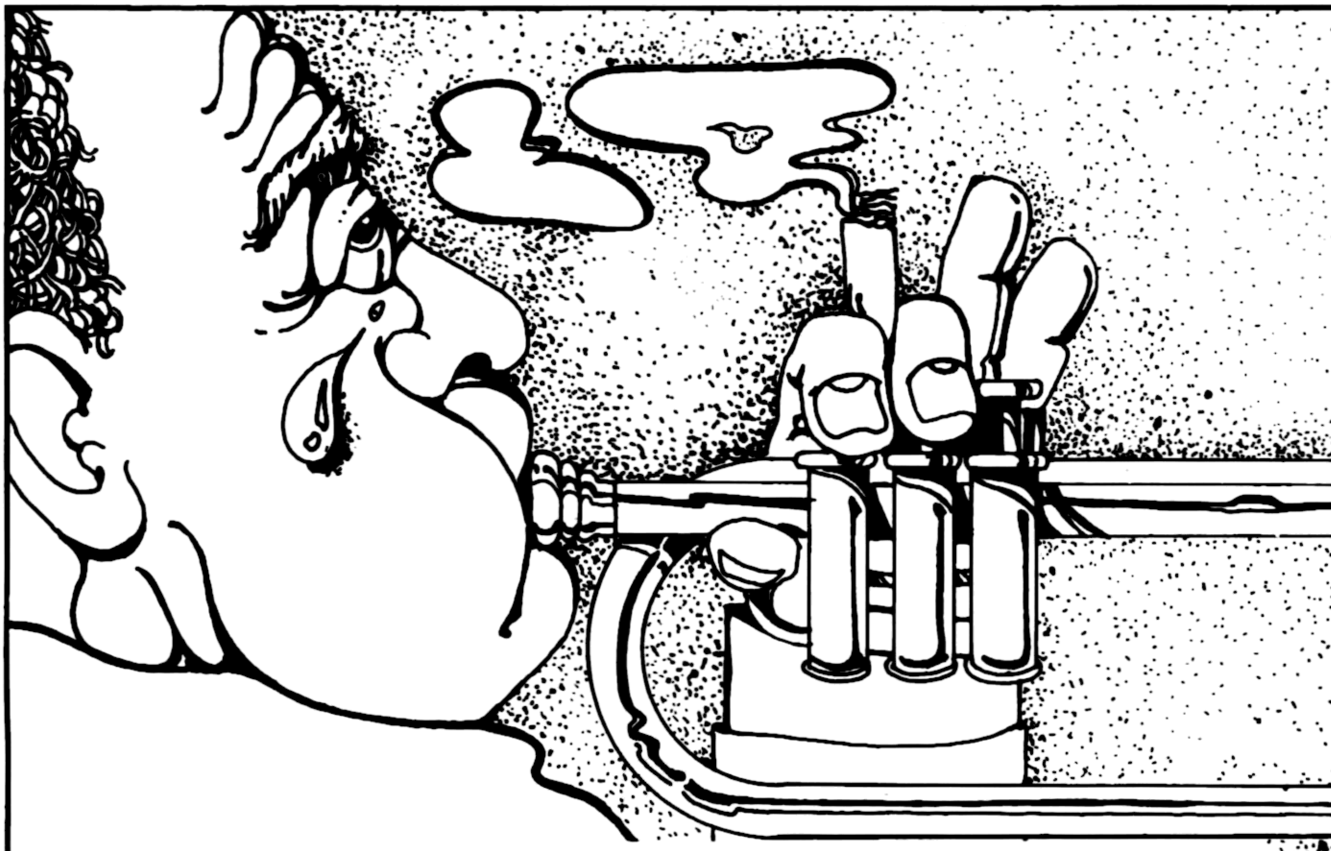


Fig. 1. Adding a BREAK key onto the modem.



ground. However, switching 12 volts onto pin 2 does not work. If pin 2 is initially at −12 volts, then switching +12 volts onto pin 2 will not give a valid RS-232C logic level. However, switching +18 volts onto pin 2 does give a valid RS-232C logic level, even if pin 2 is initially at +12 volts. This voltage can be applied directly to pin 2 without damage to any components because the RS-232C standard also specifies that logic signals must be short-circuit protected, not only to ground but also to each other. Fig. 1 shows the modem modification schematically.

A New Terminal Program

If the thought of drilling a hole in your modem cabinet gives you an empty feeling, there is still the software solution. The assembly-language listing shows my version of the Term program, which I have called Term2. Basically, it is the Term program with a different mainline. As you can see, many of the subroutines from Term are still intact. Term2 differs from Term in several respects, however.

In Term2 no text is transmitted until the ENTER key is pressed. This feature allows you to conveniently correct typing errors or to change your mind in the middle of typing a command. The up-arrow key is used to clear the screen and reinitialize the UART. This feature is useful if you can't stand a cluttered CRT screen or if you forget to turn on the expansion interface before starting the program. Shift @ is used to clear the current line. This feature is convenient if you have typed in a long command and then decide to scratch it and start over.

BREAK is the software break key. One minor difference from the hardware break key is that the break signal is sent for a fixed period of time instead of being sent as long as the key is depressed. The left-arrow key still functions as the backspace key. Note that this function is provided by the subroutine KBD in ROM, not by Term2. Finally, shift B is used to return control to BASIC. This is much more convenient than pressing BREAK and RESET at the same time, as you must do to exit from Term.

There are also some differences in the receive function of Term2. One difference is that Term2 ignores DEL characters (hex 7F). To the TRS-80 this character is a lowercase underscore. The computer at NMSU transmits DEls when it is being waited on but is not ready to reply. Using Term, I found that when the computer is heavily loaded it is possible to receive an entire page of underscore characters before any response is obtained.

Another difference in Term2 is that the receiver does not respond while characters are being input from the keyboard. I wrote the program this way because I could see no benefits in having extraneous characters appear on the screen right

Assembly language listing of Term 2; it can be entered using a monitor such as TBUG or the R.S. Editor/Assembler.

```
                00110 ;  THIS ROUTINE IS USED WITH THE RADIO SHACK RS-232-C
                00120 ;  SERIAL INTERFACE AND MAY BE USED INSTEAD OF THE TERM
                00130 ;  PROGRAM FURNISHED BY RADIO SHACK.  NOTE THAT THIS
                00140 ;  PROGRAM ALLOWS THE USER TO CORRECT TYPING ERRORS.
                00150 ;  NO INFORMATION IS TRANSMITTED UNTIL ENTER IS PRESSED.
                00160 ;  NOTE ALSO THAT PRESSING THE UP-ARROW KEY CLEARS THE
                00170 ;  ENTIRE SCREEN. SHIFT @ CLEARS THE CURRENT LINE,
                00180 ;  SHIFT B RETURNS TO BASIC, AND BREAK REQUESTS TO
                00190 ;  BREAK TRANSMISSION FROM THE HOST COMPUTER.
                00200 ;
                00210 ;
0033            00220 DSP     EQU     33H            ;DISPLAY BYTE ROUTINE
002B            00230 KBD     EQU     2BH            ;KEYBOARD INPUT ROUTINE
0046            00240 CIO     EQU     46H            ;COMMON IO ROUTINE
4020            00250 CSRPOS  EQU     4020H          ;CURSOR POSITION
1A19            00260 BASIC   EQU     1A19H          ;BASIC ENTRY PT
                00270 ;
7E00            00280         ORG     7E00H          ;PROGRAM ORIGIN (32256)
                00290 ;
7E00 CDA77E     00300 TERM    CALL    HOMCLR         ;HOME CURSOR/CLEAR SCREEN
7E03 CD107F     00310         CALL    MRUART         ;INITIALIZE UART
7E06 CDED7E     00320 GETCP   CALL    GETCRS         ;GET CURSOR POSITION
7E09 CD2B00     00330 RDKBD   CALL    KBD            ;CHECK KEYBOARD
7E0C B7         00340         OR      A              ;ANY INPUT?
7E0D 28FA       00350         JR      Z,RDKBD        ;GO IF NO KEYBOARD INPUT
7E0F FE5B       00360 TESTIN  CP      91             ;UP ARROW?
7E11 28ED       00370         JR      Z,TERM         ;CLEAR SCREEN IF YES
7E13 FE62       00380         CP      98             ;SHIFT B?
7E15 2006       00390         JR      NZ,$+8         ;SKIP 2 INSTR IF NOT
7E17 CDA77E     00400         CALL    HOMCLR         ;CLEAR SCREEN
7E1A C3191A     00410         JP      BASIC          ;RETURN TO BASIC
7E1D FE60       00420         CP      96             ;SHIFT @?
7E1F 2005       00430         JR      NZ,$+7         ;SKIP 2 INSTR IF NOT
7E21 CDB77E     00440         CALL    CLRLIN         ;CLEAR CURRENT LINE
7E24 18E0       00450         JR      GETCP          ;GET NEW POSITION
7E26 FE01       00460         CP      01             ;BREAK?
7E28 2005       00470         JR      NZ,$+7         ;SKIP 2 INSTR IF NOT
7E2A CDC27E     00480         CALL    BREAK          ;XMIT BREAK SIGNAL
7E2D 1848       00490         JR      RCV            ;GO INTO RECEIVE MODE
7E2F FE0D       00500         CP      0DH            ;IS CHAR A C.R.?
7E31 2805       00510         JR      Z,ETX          ;GO IF C.R.
7E33 CD3300     00520         CALL    DSP            ;ELSE DISPLAY CHAR
7E36 18D1       00530         JR      RDKBD          ;GO BACK FOR MORE
7E38 2A2040     00540 ETX     LD      HL,(CSRPOS)    ;LOAD CSR POS
7E3B 22A37E     00550         LD      (ENDAD),HL     ;STORE END ADDRESS
7E3E AF         00560         XOR     A              ;CLEAR CARRY FLAG
7E3F ED5BA17E   00570         LD      DE,(START)     ;CURSOR START ADDR
7E43 ED52       00580         SBC     HL,DE          ;# CHAR TO SEND
7E45 7C         00590         LD      A,H            ;LOAD HI LEN BYTE
7E46 E603       00600         AND     03H            ;LEN NOT > 3FFH
7E48 67         00610         LD      H,A            ;RESTORE H
7E49 22A57E     00620         LD      (CTR),HL       ;STORE BYTE COUNT
                00630 ;
                00640 ;   (BELOW) TRANSMIT THE TEXT JUST ENTERED
7E4C ED5BA57E   00650         LD      DE,(CTR)       ;LOAD BYTE COUNT
7E50 7A         00660 TSTCT   LD      A,D            ;LOAD HI BYTE
7E51 B3         00670         OR      E              ;TEST FOR ZERO
7E52 2819       00680         JR      Z,SENDCR       ;GO IF COUUNT = 0
7E54 2AA17E     00690 SEND    LD      HL,(START)     ;GET CHAR ADDRESS
7E57 7E         00700         LD      A,(HL)         ;PICK UP A CHARACTER
7E58 23         00710         INC     HL             ;POINT TO NEXT CHAR
7E59 22A17E     00720         LD      (START),HL     ;STORE NEW PTR
7E5C CDFE7E     00730         CALL    SOUT           ;XMIT CHAR
7E5F CD057E     00740         CALL    WAIT           ;WAIT A WHILE
7E62 ED5BA57E   00750         LD      DE,(CTR)       ;LOAD BYTE COUNT
7E66 1B         00760         DEC     DE             ;DECREMENT COUNT
7E67 ED53A57E   00770         LD      (CTR),DE       ;RESTORE COUNT
7E6B 18E3       00780         JR      TSTCT          ;TEST COUNT
7E6D 3E0D       00790 SENDCR  LD      A,0DH          ;LOAD A C.R.
7E6F CDFE7E     00800         CALL    SOUT           ;AND SEND IT
7E72 3E0D       00810         LD      A,0DH          ;DO C.R. ON CRT
7E74 CD3300     00820         CALL    DSP
                00830 ;
7E77 CD077F     00840 RCV     CALL    SIN            ;ANY UART INPUT?
7E7A B7         00850         OR      A              ;TEST AND SEE
7E7B 200B       00860         JR      NZ,DSPC        ;GO IF UART INPUT
7E7D CD2B00     00870         CALL    KBD            ;TEST FOR KEYBOARD INPUT
7E80 B7         00880         OR      A              ;ANYTHING?
7E81 28F4       00890         JR      Z,RCV          ;GO IF NOT
7E83 CDED7E     00900         CALL    GETCRS         ;ELSE GET CURSOR POSITION
7E86 1887       00910         JR      TESTIN         ;BACK TO INPUT ROUTINE
7E88 FE7F       00920 DSPC    CP      127            ;TEST FOR DEL
7E8A 28EB       00930         JR      Z,RCV          ;GO IF DEL RECEIVED
7E8C FE0A       00940         CP      0AH            ;TEST FOR L.F.
7E8E 28E7       00950         JR      Z,RCV          ;IGNORE L.F.
7E90 E67F       00960         AND     7FH            ;STRIP PARITY BIT OFF
7E92 FE60       00970         CP      60H            ;TEST FOR LOWER CASE
7E94 FA997E     00980         JP      M,$+5          ;GO IF NOT
7E97 E65F       00990         AND     5FH            ;CONV LOWER TO UPPER CASE
7E99 CDDE7E     01000         CALL    TSTERR         ;RECEIVE ERROR?
7E9C CD3300     01010         CALL    DSP            ;DISPLAY RCD CHAR
7E9F 18D6       01020         JR      RCV            ;BACK TO RECEIVE ROUTINE
                01030 ;
7EA1 0000       01040 START   DEFW    0              ;CURSOR START ADDR
7EA3 0000       01050 ENDAD   DEFW    0              ;CURSOR END ADDR
7EA5 0000       01060 CTR     DEFW    0              ;NR BYTES TO XMIT
                01070 ;
7EA7 3E1C       01080 HOMCLR  LD      A,1CH          ;HOME CURSOR
7EA9 CD3300     01090         CALL    DSP
7EAC 3E1F       01100         LD      A,1FH          ;CLEAR SCREEN
7EAE CD3300     01110         CALL    DSP
7EB1 3E0E       01120         LD      A,0EH          ;TURN CURSOR ON
7EB3 CD3300     01130         CALL    DSP
7EB6 C9         01140         RET
                01150 ;
7EB7 3E1D       01160 CLRLIN  LD      A,29           ;CURSOR TO START OF LINE
7EB9 CD3300     01170         CALL    DSP
7EBC 3E1E       01180         LD      A,30           ;ERASE TO EOL
7EBE CD3300     01190         CALL    DSP
7EC1 C9         01200         RET
                01210 ;
```

```
7EC2 3A207F   01220 BREAK  LD    A,(CTRLIM)     ;LOAD CTRL REG IMAGE
7EC5 E6FB     01230        AND   0FBH           ;CLEAR BREAK BIT
7EC7 D3EA     01240        OUT   (CTRL),A       ;STORE IN CTRL REG
7EC9 21FFFF   01250        LD    HL,0FFFFH      ;LOAD LONG COUNT
7ECC CD087E   01260        CALL  WL             ;WAIT A WHILE
7ECF 3A207F   01270        LD    A,(CTRLIM)     ;RELOAD CTRL REG IMAGE
7ED2 D3EA     01280        OUT   (CTRL),A       ;SET BREAK BIT
7ED4 C9       01290        RET                  ;RETURN
              01300 ;
7ED5 21FF10   01310 WAIT   LD    HL,10FFH       ;LOAD WAIT COUNT
7ED8 2B       01320 WL     DEC   HL             ;DECREMENT COUNT
7ED9 7C       01330        LD    A,H            ;TEST FOR ZERO
7EDA B5       01340        OR    L
7EDB 20FB     01350        JR    NZ,WL          ;GO IF NOT ZERO
7EDD C9       01360        RET                  ;ELSE RETURN
              01370 ;
7EDE F5       01380 TSTERR PUSH  AF             ;TEST FOR RECEIVE ERR
7EDF 3A1E7F   01390        LD    A,(STATUS)     ;GET UART STATUS
7EE2 E638     01400        AND   38H            ;FROM UCB
7EE4 2805     01410        JR    Z,NOFLT        ;OV, FE, PE?
7EE6 3EAA     01420        LD    A,0AAH         ;ERROR DETECTED
7EE8 CD3300   01430        CALL  DSP            ;DISPLAY BAR
7EEB F1       01440 NOFLT  POP   AF
7EEC C9       01450        RET
              01460 ;
7EED F5       01470 GETCRS PUSH  AF             ;SAVE ACCUMULATOR
7EEE 3E0E     01480        LD    A,0EH          ;CURSOR CHARACTER
7EF0 CD3300   01490        CALL  DSP            ;TURN CURSOR ON
7EF3 F1       01500        POP   AF             ;RECOVER ACCUMULATOR
7EF4 2A2040   01510        LD    HL,(CSRPOS)    ;GET CURSOR ADDRESS
7EF7 22A17E   01520        LD    (START),HL     ;CSR START ADDRESS
7EFA 22A37E   01530        LD    (ENDAD),HL     ;CSR END ADDRESS
7EFD C9       01540        RET
              01550 ;
7EFE 11197F   01560 SOUT   LD    DE,SUCB        ;OUTPUT BYTE TO RS-232-C
7F01 C5       01570        PUSH  BC
7F02 0620     01580        LD    B,20H
7F04 C34600   01590        JP    CIO
              01600 ;
7F07 11197F   01610 SIN    LD    DE,SUCB        ;INPUT BYTE IF ANY
7F0A C5       01620        PUSH  BC
7F0B 0640     01630        LD    B,40H
7F0D C34600   01640        JP    CIO
              01650 ;
7F10 11197F   01660 MRUART LD    DE,SUCB        ;RESET UART
7F13 C5       01670        PUSH  BC
7F14 0680     01680        LD    B,80H
7F16 C34600   01690        JP    CIO
              01700 ;
              01710 ;   RS-232-C UNIT CONTROL BLOCK
              01720 ;
7F19 E0       01730 SUCB   DEFB  0E0H           ;FUNCTIONS MASK
7F1A 217F     01740        DEFW  RS232          ;DRIVER ADDRESS
7F1C 00       01750        DEFB  0              ;TERM SWITCH CONFIG
7F1D 00       01760        DEFB  0              ;BAUD RATE CODE
7F1E 00       01770 STATUS DEFB  0              ;UART STATUS
7F1F 00       01780        DEFB  0              ;MODEM STATUS
7F20 00       01790 CTRLIM DEFB  0              ;CTRL REG IMAGE
              01800 ,    RS-232-C DRIVER
              01810 ;
              01820 ;    ENTRY:  IX => UCB+0
              01830 ;            C  =  PARAMETER
              01840 ;            B  =  FCT CODE
              01850 ;
              01860 ;    EXIT:   A  =  STATUS OR DATA
7F21 78       01870 RS232  LD    A,B
7F22 17       01880        RLA                  ;EXAMINE FCT REQUEST
7F23 3821     01890        JR    C,IUART        ;INITIALIZE UART?
7F25 17       01900        RLA
7F26 3805     01910        JR    C,RSRD         ;READ DATA FROM UART?
7F28 17       01920        RLA
7F29 380E     01930        JR    C,RSWR         ;WRITE TO UART?
7F2B AF       01940 RSX    XOR   A
7F2C C9       01950        RET
              01960 ;
7F2D DBEA     01970 RSRD   IN    A,(CTRL)       ;GET UART STATUS REG
7F2F DD7705   01980        LD    (IX+5),A       ;IMAGE TO UCB
7F32 CB7F     01990        BIT   7,A            ;IS RCVD BYTE AVAILABLE
7F34 28F5     02000        JR    Z,RSX          ;IF NOT
7F36 DBEB     02010        IN    A,(DATA)       ;ELSE GET DATA BYTE
7F38 C9       02020        RET
              02030 ;
7F39 DBEA     02040 RSWR   IN    A,(CTRL)       ;GET STATUS REG
7F3B DD7705   02050        LD    (IX+5),A
7F3E CB77     02060        BIT   6,A            ;XMIT HOLD REG EMPTY?
7F40 28F7     02070        JR    Z,RSWR         ;WAIT IF NOT
7F42 79       02080        LD    A,C
7F43 D3EB     02090        OUT   (DATA),A       ;OUTPUT BYTE
7F45 C9       02100        RET
              02110 ;
00E8          02120 MR     EQU   0E8H
00E8          02130 MODEM  EQU   0E8H
00E9          02140 CONFIG EQU   0E9H
00EA          02150 CTRL   EQU   0EAH
00EB          02160 DATA   EQU   0EBH
              02170 ;
              02180 ;    INITIALIZE RS-232-C HARDWARE USING CONFIG SWITCHES
              02190 ;
7F46 D3E8     02200 IUART  OUT   (MR),A         ;RESET UART WITH OUT DATA
7F48 DBE9     02210        IN    A,(CONFIG)     ;GET CONFIG SWITCHES
7F4A DD7703   02220        LD    (IX+3),A       ;SAVE IMAGE
7F4D E6F8     02230        AND   0F8H           ;MASK OFF BAUD RATE INFO
7F4F F605     02240        OR    05H            ;SET BRK, RESET DTR, RTS
7F51 DD7707   02250        LD    (IX+7),A       ;SAVE IMAGE OF CTRL REG
7F54 D3EA     02260        OUT   (CTRL),A       ;AND PUT IN CTRL REG
              02270 ;
7F56 DBE9     02280 IBRG   IN    A,(CONFIG)     ;GET BAUD RATE SWITCHES
7F58 E607     02290        AND   07H            ;BAUD RATE BITS ONLY
7F5A 21697F   02300        LD    HL,BAUDTB
7F5D 0600     02310        LD    B,0
7F5F 4F       02320        LD    C,A
7F60 09       02330        ADD   HL,BC          ;HL => BAUD RATE CODE
7F61 7E       02340        LD    A,(HL)         ;GET BAUD RATE CODE
7F62 DD7704   02350        LD    (IX+4),A       ;SAVE IMAGE IN UCB
7F65 D3E9     02360        OUT   (CONFIG),A     ;LOAD BAUD RATE GEN
7F67 AF       02370        XOR   A
7F68 C9       02380        RET
              02390 ;
              02400 ;    BAUD RATE CODE TABLE
              02410 ;
7F69 22       02420 BAUDTB DEFB  22H            ;110 BAUD
7F6A 44       02430        DEFB  44H            ;150 BAUD
7F6B 55       02440        DEFB  55H            ;300
7F6C 66       02450        DEFB  66H            ;600
7F6D 77       02460        DEFB  77H            ;1200
7F6E AA       02470        DEFB  0AAH           ;2400
7F6F CC       02480        DEFB  0CCH           ;4800
7F70 EE       02490        DEFB  0EEH           ;9600
              02500 ;
              02510 ;    TABLE OF SPECIAL CODES
              02520 ;
7F71 03       02530 SPECTB DEFB  03H            ; EOT
7F72 1B       02540        DEFB  1BH            ; ESC
7F73 7C       02550        DEFB  7CH            ; VERT BAR
7F74 7F       02560        DEFB  7FH            ;DEL
              02570 ;
7E00          02580        END   TERM           ;WRITE ENTRY PT ADDR
00000 TOTAL ERRORS
SPECTB  7F71
BAUDTB  7F69
IBRG    7F56
CONFIG  00E9
MODEM   00E8
MR      00E8
DATA    00EB
RSX     7F2B
RSWR    7F39
RSRD    7F2D
IUART   7F46
RS232   7F21
SUCB    7F19
NOFLT   7EEB
STATUS  7F1E
WL      7ED8
CTRL    00EA
CTRLIM  7F20
TSTERR  7EDE
DSPC    7E88
SIN     7F07
WAIT    7ED5
SOUT    7EFE
SEND    7E54
SENDCR  7E6D
TSTCT   7E58
CTR     7EA5
START   7EA1
ENDAD   7EA3
ETX     7E38
RCV     7E77
BREAK   7EC2
CURLIN  7EB7
TESTIN  7E0F
RDKBD   7E09
GETCRS  7EED
GETCP   7E06
MRUART  7F10
HOMCLR  7EA7
TERM    7E00
BASIC   1A19
CSRPOS  4020
CIO     0046
KBD     002B
DSP     0033
```

in the middle of typing in a command. Also, I prefer to type my sign-on message before communication is established and then just press ENTER when the system is ready to communicate. If the receive function is active while communication is being established, noise in the room or on the phone will invariably cause a garbage character or two to be added to your sign-on command.

If you look at the portion of the code in Term2 where data from the CRT screen is transmitted character by character, you will notice that there is a call to WAIT after each byte is sent. When I am communicating with the NMSU time-sharing system, this delay is necessary on the initial transmission. After the first transmission no delay between characters is needed; however, it simplifies the logic to leave the delay in. I experimented and found that taking the delay out does not noticeably improve operation, so I left the delay in.

The listing of Term2 and an object code tape were produced by Radio Shack's Editor/Assembler program, which I highly recommend. You can use T-BUG to produce an object tape if you do not have the Editor/Assembler program. However, it is tedious to input an object program with T-BUG. If you wish to experiment with the delay time between characters, T-BUG is convenient to use.

Now What?

Now that you can use your TRS-80 as an intelligent terminal, what do you do now? To answer that question, you must first know what is available on the particular system(s) on which you will be time-sharing. At NMSU, for example, there are a number of useful packages available. In particular, I plan to use those functions that allow me to create and edit programs in FORTRAN, COBOL and other high-level languages. Other functions are available to submit jobs to the system, check job status, review the output from a run and request output to be printed or canceled. All this is available from an ASCII terminal. If your terminal has the APL character set, even more power and fun are at your disposal.

For those who are interested in using the APL, transmitting the APL character set instead of ASCII should involve a fairly direct modification to Term2. The problem of displaying APL characters is a little more difficult. Perhaps each APL operator could be represented by a two-character abbreviation enclosed in brackets. I leave this as an interesting problem to be solved.■

Debounce your fingers and listen to the results.

Listen To Your Keyboard

Allan J. Domuret
7825 Willowcrest Way
Fair Oaks CA 95628

Radio Shack is marketing a software debounce program to cure unintended multiple-character generation from the keyboard. For those of you who are unfamiliar with the problem, keyboard bounce is caused by the mechanical opening and closing of keyboard switches, which results in multiple-character outputs to the computer. The bounce problem can be severe if the keyboard contacts become dirty or if you have nervous fingers. Debounce can be overcome with either software or hardware, but Radio Shack neglected both, with one exception that will be addressed in the following paragraphs. Radio Shack's software fix is already on the market. If you haven't already purchased it, here is my version, free. Just load it in with the Radio Shack Editor/Assembler.

In fact, I believe my debounce program is superior to Radio Shack's because mine includes generation of keyboard audio feedback so that you can hear every keystroke, accidental multiple keystrokes and missed keystrokes, with only some minor, and optional, modifications to your cassette recorder. The audio feedback supplements the debounce software by contributing to the reduction of typing errors.

As an added bonus, some cassette recorder modifications, which will allow for DEBNC audio feedback and also improve the performance of your recorder, are included.

The DEBNC program sends keyboard audio signals to the cassette recorder without activating the cassette operating relay with every keystroke. This design prevents beating the relay to death while typing and it also keeps DEBNC from interfering with CLOAD and CSAVE functions. However, you have to manually turn on your recorder in order to hear the audio feedback. This provides a built-in safety feature, which should prevent accidental erasure of tapes left in the recorder.

Keyboard Bounce: Its Causes and Cures

As was mentioned above, keyboard bounce is caused by the mechanical opening and closing of keyboard switch contacts. Fig. 1 explains what actually happens every time a key is pressed. In the TRS-80, all eight data lines are held at logical zero while ROM software scans the keyboard for a keystroke. When you press a key, a logical one is output to the appropriate data line, which is then detected and decoded by ROM software. (The details of how ROM scans and decodes the keyboard are beyond the scope of this article, but for those who are interested I recommend an excellent book by Titus, Rony, Larsen and Titus called *8080/8085 Software Design*, published by Howard W. Sams & Co., 1978. As a relative newcomer to the field of microcomputers and machine-language programming, I found this book extremely informative and easy to read, even though it is oriented to the 8080/8085 CPUs. Keyboard scanning and debounce routines are covered in chapter 7.)

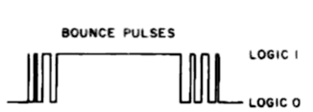


Fig. 1. Leading and trailing pulses when a key is pressed.

Fig. 1 shows the generation of a series of random pulses when a keyboard switch is initially closed. As the key is held down for a few milliseconds, the pulsations even out as the switch contacts settle down against each other.

When the key is released, another series of random pulses is generated as the switch contacts separate. As a result of this switch bounce, a collection of logical ones is sent to the data lines, and, depending on the severity of the bounce, ROM sometimes interprets these bounce pulses as multiple keystrokes rather than only one keystroke, hence, the multiple-character problem. Ideally, if we could send a single pulse to ROM as shown in Fig. 2, a single keystroke would be properly decoded by ROM and the multiple-character generation problem would be eliminated.

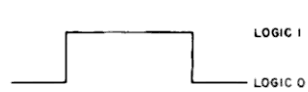


Fig. 2. Continuous output.

An inspection of the TRS-80 keyboard switches will help clarify the cause of keyboard bounce. Gently pry up the space bar at its center (the space bar is easier to get at than the other keys) with a plastic lever such as a thin comb. Don't use a metal prier, such as a screwdriver, or you will nick the plastic. Now, watch the exposed metal-switch contacts while you press down the square key holder. It should be fairly obvious from observing the action of these contacts that there is some inherent spring or bounce in them. Since the space bar is loose, leave it off because later we'll see how to clean all the keyboard contacts.

In order to eliminate the bounce problem, it is necessary to smooth out the leading and trailing pulses as illustrated in Fig. 1 to obtain a reasonably continuous output as shown in Fig. 2. Hardware can be employed to filter these pulses into a smooth output pulse, such as an alternating current rectifier circuit. But the focus here is on software, so we won't be getting into hardware design.

The leading and trailing pulses rarely last longer than a few milliseconds, so if ROM can be convinced to ignore the first and last few milliseconds of keyboard output it could direct its processing efforts on the center or flat part of the keyboard output pulse. The solution, then, is to tell ROM to ignore the first and last ten milliseconds or so of keyboard output, thereby solving the bounce problem. This is what DEBNC does.

TRS-80 Debounce Software

The program is a computer-typed output from the TRS-80 Editor/Assembler (typed by an IBM Selectric rather than a dot-matrix printer, hence, it is more readable). The first column is memory location for a 16K system. Note that the program resides in upper memory. For 32K or 48K systems, the ORG (ORiGinate) instruction on the top line should be adjusted to BFBCH (BFBC hex, which corresponds to 49084 decimal) or FFBCH (FFBC hex, which corresponds to 65468 decimal). For the accompanying 16K program, the 7FBCH originate address corresponds to 32700 decimal. These decimal originate addresses correspond to MEMORY SIZE? as requested by ROM when the computer is powered up; keep them handy.

Column two is hexadecimal machine language, which is automatically generated by the TRS-80 Editor/Assembler. Column three represents line numbers to ease programming and editing. Column four is the label field used to simplify addressing and branching in the program. Columns five and six are the familiar Z-80 mnemonics. Note that the labels in column four correspond to addresses referenced in column six. For instance, subroutine DELAY in column four corresponds to memory location 7FE0 hex and is referenced by the instruction CALL DELAY at memory location 7FCD hex.

Programming with the TRS-80 Editor/Assembler only requires typing in the information in columns four through six. The assembler automatically generates the line numbers and computes the information in columns three and four during assembly.

Debounce Relay

DEBNC keeps an eye on the keyboard, which, in its quiescent (idle) state, outputs a continuous stream of logical zeros on all data lines. If the instructions at lines 140 through 160 detect only zeros from the keyboard, scanning the keyboard continues until something more interesting is detected. When a key is pressed, a logical one is put onto one of the data lines and lines 140 through 160 immediately recognize this different-from-zero output.

Before the CPU is allowed to process this nonzero keyboard output, the debounce software introduces a short time delay of a few milliseconds to allow the keyboard switch bounce to settle down. It is during this short delay period that the program generates an audio tone and sends it to the cassette output port. After all, why not let the computer do something useful while it is killing time?

If you study the program closely, you will note that no time delay is provided to compensate for keyboard bounce upon key release. The reason is that ROM already contains a short delay to do this. (Those of you who have a monitor such as the Small System Software RSM-1S can see the ROM CALL for this key-release time delay at memory location 044F hex. The actual delay is a subroutine at memory location 0060 hex.)

I don't know why Radio Shack designers went only half-way by providing for debounce upon key release and not upon initial key press. At any rate, this is the exception I mentioned above in the introductory paragraphs.

As for the audio output, the DEBNC DELAY subroutine simply calls up the save-memory-to-cassette software in ROM and outputs a series of pulses to the cassette port. The pulses consist of alternating sync pulses used in all cassette recordings, interspersed with logic ones (FF hex in lines 390 and 410). These pulses are sent out to the cassette as if the computer intended to record

```
7FBC               00100          ORG     7FBCH
7FBC 213640        00110 DEBNC    LD      HL,4036H
7FBF 010138        00120          LD      BC,3801H
7FC2 1600          00130          LD      D,00H
7FC4 0A            00140 CKKEY    LD      A,(BC)
7FC5 A7            00150          AND     A
7FC6 2809          00160          JR      Z,ZERO
7FC8 5F            00170          LD      E,A
7FC9 7E            00180          LD      A,(HL)
7FCA BB            00190          CP      E
7FCB 2803          00200          JR      Z,INAGN
7FCD CDE07F        00210          CALL    DELAY
7FD0 0A            00220 INAGN    LD      A,(BC)
7FD1 5F            00230 ZERO     LD      E,A
7FD2 AE            00240          XOR     (HL)
7FD3 73            00250          LD      (HL),E
7FD4 A3            00260 INCSCN   AND     E
7FD5 C2FA03        00270          JP      NZ,03FAH
7FD8 14            00280          INC     D
7FD9 2C            00290          INC     L
7FDA CB01          00300          RLC     C
7FDC F2C47F        00310          JP      P,CKKEY
7FDF C9            00320          RET
7FE0 3E00          00330 DELAY    LD      A,0
7FE2 32E437        00340          LD      (37E4H),A
7FE5 E5            00350          PUSH    HL
7FE6 2100FF        00360          LD      HL,0FF00H
7FE9 CD2102        00370          CALL    0221H
7FEC E1            00380          POP     HL
7FED 3EFF          00390          LD      A,0FFH
7FEF CD6402        00400          CALL    0264H
7FF2 3EFF          00410          LD      A,0FFH
7FF4 CD6402        00420          CALL    0264H
7FF7 CDF801        00430          CALL    01F8H
7FFA C9            00440          RET
4016               00450          ORG     4016H
4016 BC7F          00460          DEFW    DEBNC
0000               00470          END
00000 TOTAL ERRORS

INCSCN   7FD4
DELAY    7FE0
INAGN    7FD0
ZERO     7FD1
CKKEY    7FC4
DEBNC    7FBC
```

DEBNC program symbolic list.

them.

One concern in developing the program, however, was to keep the recorder in a normally off condition to prevent accidental tape erasures, while still preventing the computer from turning on the cassette-controlling relay every time it output a tone in response to each keyboard keystroke. This is accomplished by modifying the ROM CSAVE subroutine in DEBNC lines 340 through 380.

The cassette relay-turn-on override takes place in line 360: To turn on the motor for recording, ROM software would normally "LD HL,FF04H," but instead we simply "LD HL,-FF00H" to prevent the cassette from being turned on while still allowing the audio output go to the cassette port.

Without this feature, the ROM software, if it had its own way, would turn on the cassette every time a keystroke was output to the cassette port, and by now most TRS-80 owners are aware that such abuse of the cassette-control relay would send the relay to an early grave. Now, how do we get the cassette recorder to cooperate and give us the audio output from software? There are several options.

The easiest way to get the audio out of the cassette is to connect a small 3.2 Ohm speaker to a miniature phone jack and plug it into the EAR output jack on the side of the recorder (see the cassette recorder modifications section for an alternative, and preferred, method). Next, it is necessary to get manual control of the cassette recorder by either pulling the Remote plug from the side of the recorder or installing an override switch of the type described by Frank B. Rowlett, Jr., in *Microcomputing*, January 1979, p. 54 (for an alternative method, see the recorder mod section).

With the recorder now enabled, raise the tape cover by pressing the EJECT lever on the recorder. Then in the upper-left corner of the tape cavity you will find an "erase protect" lever that protrudes when you attempt to depress the RECORD lever. Hold this erase-protect lever in while simultaneously depressing the RECORD and PLAY levers as you would in preparing a recording. Manually holding in this erase-protect lever enables the red RECORD lever to be depressed. This activates the cassette amplifier and allows the audio from the computer to enter the amplifier via the cassette Aux Input.

By now you probably have noticed one glitch. This procedure keeps the cassette motor running continuously while DEBNC is used in this mode. If you spend hours typing a BASIC program into the computer using DEBNC with its audio feature, your cassette motor will run for these same hours. You have several options:

1. Let the motor run. It has a long life and you really won't hurt it.

2. Install a motor turn-off switch to deactivate the motor without defeating the cassette amplifier. This, too, is covered in the recorder mod section.

3. Ignore the audio output. The debounce program will still use the audio output subroutine to generate the necessary debounce time delay, but you just won't hear it and it won't hurt anything.

4. Feed the audio tone to a separate amplifier.

Notice the built-in safety feature of this design. There is no way to activate the cassette recorder with DEBNC and accidentally erase a valuable cassette tape. Of course, it is possible to leave a tape in the recorder to enable activation of the Record/Play levers, but the danger of doing this, I believe, is low. By now, most computerists have developed good tape-handling practices so as to avoid such accidents.

Perhaps it would be worthwhile to mention the purpose of lines 170 through 200 in DEBNC. Without these program steps, the TRS-80 keyboard would output what would sound like a continuous audio output for as long as a key remains depressed. The reason is that as a key is held down, keyboard scanning continues and an audio tone would be output on every scan cycle for as long as the key is held down. Steps 170 through 200 determine if the keyboard output is the same as it was in the last scan cycle. If so, it skips the tone-generating delay. If a keyboard output that is different than the last scan output is detected, then the delay is permitted. This technique still preserves the debounce feature.

Lines 450 and 460 in DEBNC are used to gain control of the keyboard scan routine. In normal operation, the keyboard memory scan routine vector is stored in memory locations 4016H and 4017H. When ROM wants to scan the keyboard, it calls the contents of memory locations 4016H and 4017H and finds the ROM scan routine at memory location 03E3H. To gain control of the keyboard scan routine, it is necessary to change the contents of 4016H and 4017H so that the jump will be to DEBNC at memory 7FBCH instead of to 03E3H. This is what lines 450 and 460 in DEBNC do.

If for some reason it becomes necessary to RESET the computer while DEBNC is

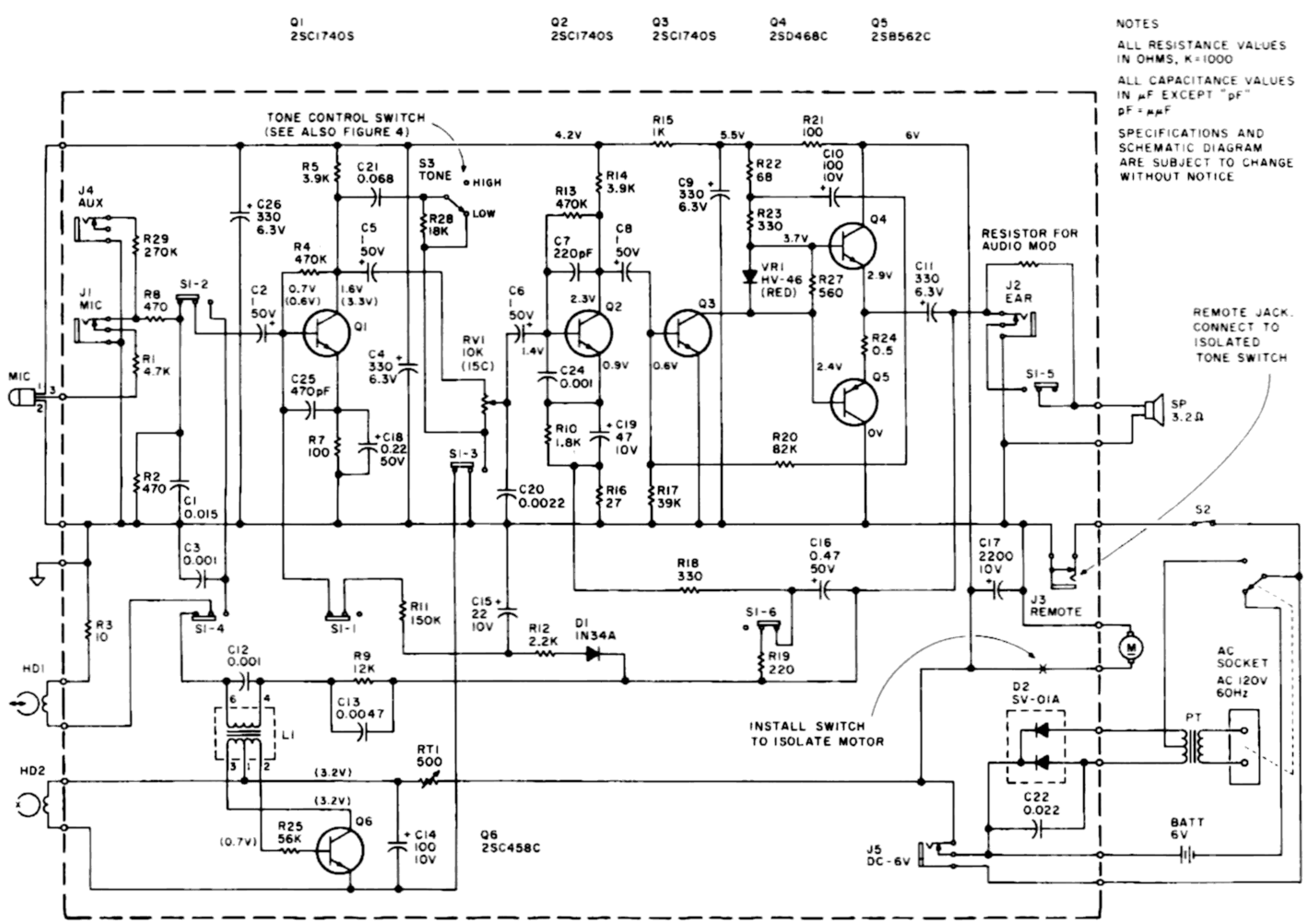


Fig. 3. CTR-41 cassette recorder schematic.

working, ROM will regain control of the keyboard scan and DEBNC will be defeated. If this happens it will be necessary to reload DEBNC. This should be no problem because DEBNC only takes a few seconds to load.

It might occur to you as it did to me to POKE the DEBNC start address into 4016H and 4017H. It won't work. Any attempt to change the keyboard scan vector located in 4016H and 4017H while ROM is busy scanning will crash the system. This will require turning the computer off and back on to reset everything.

Loading and Operating Debounce

Upon RESET or initial application of power, enter the appropriate MEMORY SIZE when so requested by the computer. Use the addresses as provided above in the TRS-80 Debounce Software section. As an example, for a 16K system, DEBNC should originate at memory 7FBCH, which corresponds to MEMORY SIZE 32700 decimal (enter the decimal figure into the computer, not the hex number).

Next, the usual System code is entered, followed by the file name, DEBNC, to start loading. After the tape loads, hit the BREAK key and DEBNC is ready. Note that this is a departure from typical system tape-loading procedures. No slash (/) key or ENTER key should be pressed because the modified keyboard vector which has been loaded into 4016H and 4017H automatically addresses DEBNC.

Finally, set up the cassette recorder as described in preceding paragraphs if audio feedback is desired. Personally, I find the audio feedback indispensable because it eliminates many typing errors.

Cleaning Keyboard Contacts

While you are sitting there with your space bar still hanging out, use your plastic comb or whatever and pop off all the other key caps to expose the key contacts. Now spray all the key contacts with tuner cleaner, rubbing alcohol or something similar.

Three cautions should be observed in this cleaning process. First, don't use a cleaner that could mar or otherwise damage your plastic keyboard. Perform a chemical reaction test using the cleaner on the bottom of your keyboard where possible melting or damage won't show. Second, don't use cotton swabs to dab liquid cleaner on the contacts. The cotton may leave small threads on the contacts which could interfere with normal operation of the contacts. And third, don't put any unguents on the contacts, such as Vaseline, which is an insulator, not a conductor, and will only serve to latch onto dust, cigarette smoke particles and so on to the extent that the contacts will become inoperative, either wholly or partly.

Of course, if your TRS-80 is new, this cleaning procedure should not be necessary, but if your keyboard has been setting on the table uncovered for months, the cleaning will not hurt. As a final protection, keep your keyboard covered when not in use.

The debounce software should solve most of your bounce problems, and proper care and cleaning of the keyboard key contacts will also help, even without the debounce software. Maybe in the near future some hardware expert will tell us non-hardware types how to debounce the TRS-80 without electronics. Anyone out there up to the challenge?

TRS-80 Cassette Recorder Modifications

A schematic of the Radio Shack CTR-41 cassette recorder, extracted from the owner's manual, is provided in Fig. 3. Four modifications are recommended, and three of them are, in my opinion, indispensable even without the use of Debounce software. These mods have been suggested in various forms by other hobbyists, most of them requiring some kind of external controlling box.

Refer to both the schematic and the accompanying printed

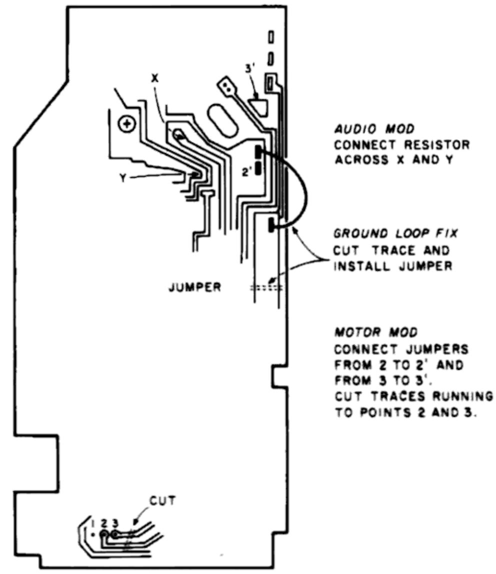


Fig. 4. CRT-41 printer circuit board.

circuit board sketch (Fig. 4) when making the mods.

1. *Audio Modification*. Connect a resistor (I used a 47 Ohm) across the top speaker wire and the top EAR connector (J2) as shown in both figures. Different size resistors will provide different volume levels. Experiment to find a suitable volume level. Fig. 4 shows where to connect this resistor on the printed circuit board.

In addition to allowing use of DEBNC audio, this resistor will also allow you to hear both CSAVE and CLOAD audio without external boxes or without the necessity of pulling plugs on the side of the recorder. With this mod, you will have no more recording surprises as a consequence of not hearing what was going into or out of the recorder. If desired, a switch can be installed in series with this resistor to defeat it.

2. *Separate Motor Control*. A switch in series with the motor as shown in the schematic will permit shutting off the motor when only the cassette amplifier is desired for DEBNC audio. This is not shown in Fig. 4 because I have not installed such a switch.

3. *Computer/Manual Cassette Control*. In Fig. 3, locate the tone control, S3. Isolating this switch from the circuit without disturbing R28 and C21 leaves the tone circuitry in the "high" mode as it should be for computer use. When properly wired, this switch can be used to get manual control of the recorder without external mods and without pulling out the Remote jack.

See Fig. 4 for instructions as to where to cut leads on the board to isolate the tone control switch. Now run two wires from the switch to the two connectors on the Remote jack as shown in both Figs. 3 and 4.

4. *Ground Loop Mod*. As long as your recorder is disassembled, this is a good time to do another indispensable mod. The stock Radio Shack CTR-41 recorder is notorious for generating hum via ground loops when used with the TRS-80. The fix is to cut the board trace and run a jumper wire as shown in Fig. 4. This fix will greatly reduce hum on computer-generated tapes and will also reduce loading problems. There are other methods for curing the ground loop problem, but this one keeps the mod inside the recorder where it belongs, out of sight.

It is my understanding that newer TRS-80 recorders have some of these mods installed, especially the ground loop fix, so it may be necessary to only perform mods one and two to isolate the cassette motor while using DEBNC with audio. You are on your own to determine your own needs. ■

NOT SATISFIED? GENERAL LEDGER INEFFICIENT?

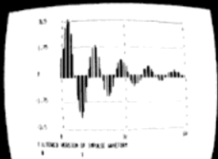
The author of ours (a public accountant) tried three and was not pleased. He wrote his own and is now using it in his business.

The program does not use bulk sorts, is fast, provides an audit trail, has 100+ accounts, has 3800+ check or general journal entries per year, prints monthly journal, prints statement of condition, prints profit and loss, prints trial balance including a listing of all entries for the year by account # (month & year posted, check or voucher #, amount, and specifically where each entry is posted to disk) and can print up to 5 departments' profit and loss statements.

A detailed manual is provided. 32K w/2 disks and printer required. Cost of the program is $175.00.

For sample printouts and more information send your name and address plus $1.00 for handling and postage to:

✓82 **Sturdivant & Dunn, Inc.,** Box 277, Conway, N.H. 03818



- ***F(ast) F(ourier) T(ransform)***
- ***Digital Filter Simulation***
- ***Linear and Exponential Curve Fit***
- ***Disk or Cassette Data & Results Files***
- ***Interactive Graphics***

■ Having this set of interactive programs in your hands is a learning experience in digital signal processing.

■ Learn by doing. Documentation includes multiple examples. Balance your checkbook with a digital filter (can you believe it?). Plot daily stock market values and their computed trend lines. Find the frequency response of a digital filter. Illustrate Nyquist sampling theorem. Perform spectral analysis on any waveform (FFT).

■ This sophisticated software, written by a professor and consultant in the digital processing field for use in teaching and research, is written in basic for ease of user understanding and modification. Runs in a minimum 16K cassette system having expanded capabilities when used with disk and printer systems.

FFT-80 DISK $30.00
FFT-80 CASSETTE $25.00

✓38

DELTA SYSTEMS
3667 Montalvo Way
Santa Barbara, CA 93105
Tel (805) 682-1270

QUALITY TRS-80 SOFTWARE

KEYWORD Indexing System

A series of programs that will create a data file on disc, build an index of all occurrences of "Keywords" in the text of the data file and allow inquiries or searches into the file using the indexed keywords. The system features:
- •flexible record lengths with location pointers
- •deletion of non keywords from index by system
- •"and" "or" "not" logic for inquiries
- •interface for user written inquiries

KEYWORD INDEX—2 disc 32K DOS system **$39.95**

SORTS for HOME and BUSINESS

No computer user should be without a versatile, easy to use sort program. The Northeast Microware in memory sort programs are written in Level II BASIC and have the following features:
- •Sort ALPHA or NUMERIC data
- •Sort on up to 5 fields simultaneously
- •In ascending or descending sequence
- •Supports kb, video or tape I/O
- •Supports seq. disk and printer I/O (SORT-IID)
- •Supports user I/O routines
- •User exits (SORT-IID only)

SORT-II—16K Level II in memory sort **$19.95**
SORT-IID—32K DOS in memory sort **$29.95**

FOR the SERIOUS GAMBLER

BLACKJACK SIMULATOR. Allows you to simulate the playing of thousands of hands of BJ and analyze the results on tape in Level II BASIC. **$19.95**

Manuals for all programs available for $3.00 ea.
(price deductible on purchase of program)

"TRS-80 is a registered trademark of TANDY CORP."

Northeast MICROWARE
BOX 6153, ✓74
SYRACUSE, N.Y. 13217

WE MEAN BUSINESS!

BUSINESS SOFTWARE, THAT IS

USE YOUR TRS-80* FOR MORE THAN FUN AND GAMES

THE DATA DUBBER $49.95
Duplicates any program tape to TRS-80 quality. Reconstructs date pulses to ensure accurate CLOADs. Permits easy loading of even poor quality commercial tapes without constant volume adjusting. Money-back guarantee if not satisfied.

THE ELECTRIC SECRETARY $75.00
A powerful word processor to turn your TRS-80 into an automatic typewriter. Features page numbering, movable margins, headers, variable page length, and title centering. Enter text, revise, correct, and output to printer page formatted, justified, even hyphenated as required. Cross-coupling files permits individually addressed form letters. Complete with upper/lower case conversion information on diskette. Specify if RS-232 adapter is installed in interface.

MAILROOM PLUS $75.00
A versatile and powerful mailing program to print labels by sequential coding: zip, city, state, customer ID code, even last name. Sorts by any code in minutes and stores sequentially in a single string (approx. 1500 records per diskette). Includes AUTOPRINT. Supplied on diskette.

MINIMAIL $50.00
A compact version of MAILROOM PLUS but without customer coding. Features alpha-lookahead for duplicates. Supplied on diskette.

FORMLET $35.00
Generates form letters from MINIMAIL records. Prepare your letter, bulletin, notice, advertisement, etc., then load the MINIMAIL files. Your printer will print the inside address, letter, and repeat for each name in the file—all properly spaced and justified. Supplied on cassette.

AUTOBOOT $15.00
Simplifies automatic BASIC program loading from your DOS. Permits sequencing through your choice of DOS commands, selects files and memory size you specify, and loads or runs selected program. Allows user to see directory and free space before program runs automatically. Supplied on cassette.

SIR ECHO $10.00
A handy program to make your printer work like an electric typewriter. Use alone or merge with your programs to make what appears on the screen echo to the printer. Supplied on cassette.

TELEFON $20.00
Make your TRS-80 a smart terminal. Communicate with time-share and other computers, bulletin boards, etc. Transfer programs over the phone. For disk systems with modem.

UPPER/LOWER CASE CONVERSION $20.00
Reprint of KILOBAUD article explaining how to modify the TRS-80 to display both upper and lower case characters. Kit contains step-by-step instructions, parts, and necessary software on cassette for case reversal, echo, and automatic line feed routines.

User group discounts available
Dealer inquiries invited

*TRS-80 is a trademark of the Tandy Corp

VISA — TERMS: Check, money order, Visa, Mastercharge. Washington residents add 5.3% for tax — master charge

THE PERIPHERAL PEOPLE
P.O. Box 524, Dep't. M
Mercer Island, WA 98040
✓43 **(206) 232-4505**

```
*********************************************
MANAGEMENT SYSTEMS SOFTWARE
ANNOUNCES
TWO NEW BUSINESS PROGRAMS FOR THE SMALL BUSINESSMAN
BY
JEROME S. OSTERYOUNG, PH.D.
RENOWNED BUSINESS EDUCATOR AND AUTHOR OF 5 BUSINESS TEXTS

*PROFORMA CASH-FLOW STATEMENT*          *LEASE-PURCHASE DECISION*
FORECASTS CASH NEEDS AND LIQUIDITY     EVALUATES LEASE VS PURCHASE DECISION
FOR UP TO 12 PERIODS                   INCORPORATES LATEST TAX IMPLICATIONS
EXTENSIVE DOCUMENTATION WITH EACH PROGRAM
CASSETTE OR DISK
WRITE FOR BROCHURE

MANAGEMENT SYSTEMS SOFTWARE
P.O. BOX 10673, ST. PETERSBURG, FL. 33733
(813)864-4347
*********************************************
```
✓87

Do not fear, Joffe is here! Calculus demystified on your computer.

Oh No! Calculus

Allan S. Joffe W3KBM
1005 Twining Road
Dresher PA 19025

Sooner or later the desire to peel yet another layer from your own personal "I wish I knew" file leads you down strange paths. If you are worried lest the word "calculus" should send you looking for the latest version of Lunar Lander, be reassured; this little trip will be thoroughly humanized and debuzzed.

The one element of calculus to be tackled is the use of the concept of integration . . . more specifically, how such use can lead to some logical determination of the area bounded by or under a curve. The curve might be any symmetrical or assymmetric open or closed curve such as a circle, a triangle or what we normally think of as a "square wave."

Finding the Area of a Circle

Integration as used here may be defined as a "process of summation." A simple example will make the basic process readily apparent. Suppose we possess a drawing compass and a ruler. We can draw a circle of a given diameter, say three inches. Assuming that we are not armed with the formula for calculating the area of this circle but possess the ability to calculate the area of a triangle, we can proceed as follows.

We can break the circle up into a series of triangles whose sides are all equal to the radius of the circle. If the bases of all the triangles are equal, the areas of all the triangles will be equal. The area of one of the triangles multiplied by the number of triangles created will then approximate the area of the circle. Fig. 1 demonstrates the idea. The smaller the base of the triangles, the closer we get to integrating the true area of the circle.

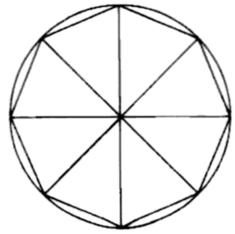


Fig. 1.

Restated, integration is the process of calculating an area by dividing it into a large number of small parts of equal area and then summing the total value of those many small divisions.

Another concept of integration might be described as mechanical integration. Suppose we add a weighing device such as a fairly sensitive balance to our compass and ruler. Let us take a square of cardboard and cut a 4 x 4 inch piece. We then weigh our 16 square inch piece of cardboard. Now draw the same 3-inch diameter circle on the cardboard and cut the circle out. Weigh the cutout circle and compare its weight to the uncut weight of the cardboard.

You can see that it is a simple matter to determine the area of the cutout circle to the original weight of the uncut cardboard by taking the ratio of the weights of the pieces. We have mechanically "integrated" the area of the circle in this simple fashion, and though the men of mathematical mystique may not agree, we have performed a function of calculus.

Integrator

There does exist a "simple" electrical equivalent of the integration variety, and being immune to Murphy's law, strangely enough it is called an integrator. Fig. 2 is a schematic of the circuit in general terms.

The thrust of our intent is to examine with the computer what a "perfect" integrator will do to a specific waveform that we will apply to it. For simplicity the selected waveform is a square wave that may have any ratio of on to off time that you wish as long as it does not

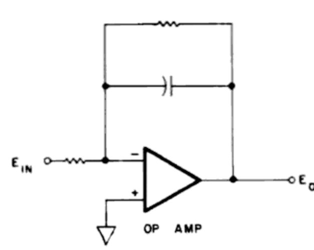


Fig. 2.

outrage the laws of nature.

We make certain basic assumptions: First, the input waveform is an ideal waveform; second, our integrator is a perfect one that is not overloaded or even verbally abused.

Fig. 3 shows our classic square wave with some annotation. The positive flat top is considered to be the *on* part of the wave and the negative (flat bottom?) is considered to be the *off* part of the wave. For ease of programming and explanation, consider that the start of the waveform is at 0 percent and the end point of the waveform is at 100 percent. Thus from start to finish it is easy to divide the X, or horizontal, axis into 100 equal parts. The Y, or vertical, axis can be considered unity as it serves no purpose to assign a specific amplitude to the Y axis. The finished program should calculate certain values that depend on the percent of on to off times and translate these values into graphics information for the TRS-80 so the output of the integrator will be graphically displayed on the monitor.

We use our integrating technique carried over from finding the area of a circle. The X axis is divided into 100 equal divisions, by which we create 100 little segments of area bounded by the waveform curve. Because we are working with a square wave, we automatically assume that the amplitude, or height, of every point on the waveform is uniform.

Thus, with segments having equal heights and equal bases, every segment is equal in area. We use this information to plot points for our graphics in the following manner. For simplicity, we can assign the area of each little segment, the value of unity.

Graphics Points

The first graphics point value is the area of the first segment. The next graphics point is the sum of the areas of segments one and two. The third graphics data point is the sum of segments one, two and three.

This summing process to form each successive data point is continued until the point on

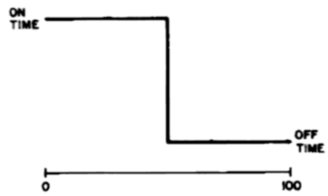


Fig. 3.

the X axis is reached where the on time ends and the off time gets into the act. When we reach this point of the waveform, the next graphics data point is the maximum-developed summing value for the on time *minus* the value of the area of the next segment. The next data point is the on maximum minus the sum of the first and second segments into the off time area of the square wave.

This process is continued until the entire 100 segments of the square wave have been integrated. The transition in the process is made necessary by the change in sign of the square wave about its axis. The on time is assumed to be positive and the off time is assumed to be negative.

We now have enough data to form a program that will accept your designated on time of the square wave to the integrator and plot out the resultant waveform. As usual, since the origin of curves drawn on the TRS-80 screen is in the upper left-hand corner where X=0 and Y=0, there is the usual problem of scaling and scale factoring to make the resultant graphics take on the same relative position as we would see them if we were using an oscilloscope.

Two rather different approaches to generating a program to fill the needs are offered. The prime reason is to show how programs embracing quite different philosophies can attain a basically identical result. Playing with these two versions will offer you a possibly finer insight into handling any graphics problems that your computer has been handing you.

The Programs

Lines 10 to 110 in Program A set the graphics to show the square wave that you have selected to run through the integrator. Lines 125 to 210 set the graphics points for the resultant integrated wave. If you want to show both the input square wave and the resultant integrated waveform on the screen together (as in the photos) eliminate line 118. Line 220 is a "holding" loop that prevents the finished graphics from being interrupted by the usual READY, which pops up when a program has run.

When the program is entered and RUN, you are asked to enter the on time of the desired square wave (i.e., 10 would mean 10 percent on time). The resultant square wave, which would be one with 10 percent on time and, naturally enough, 90 percent off time is shown. After the timing loop in line 115 times out, the screen is cleared and the integrated waveform appears.

The photographs show the results of the program for on times of 20, 50 and 90 percent. When the photographs were taken, line 118 was eliminated so that both the input waveform and the integrated waveform would be shown together on the monitor. The inclusion or exclusion of line 118 is a matter of personal choice.

You ham photographers in the group may have seen displays on the usual TRS-80 monitor and casually wondered why the images here seem so

```
10  CLS
20  REM "O" = % ON TIME OF POSITIVE PORTION OF INPUT SQUARE WAVE
30  INPUT "PERCENT ON TIME";O
40  UP=O*.01
50  X1=UP*120
60  FOR X=O TO X1
70  SET (X,4)
80  NEXT X
90  FOR X=X1 TO 120
100 SET (X,44)
110 NEXT X
115 FOR T=1 TO 1000: NEXT T
118 CLS
120 REM SETTING INTEGRATED WAVEFORMS
125 Y=4
130 FOR X=X1 TO 0 STEP -(X1/40)
140 SET (X,Y)
150 Y=Y+1
160 NEXT X
170 X=X1
180 FOR Y=4 TO 44
190 SET (X,Y)
200 X=X+(1/40)*(120-X1)
210 NEXT Y
220 GOTO 220
```

Program A.

```
5   CLS
10  INPUT "POSITIVE ON TIME" ;PSTV
15  REM SETTING THE INPUT WAVEFORM
20  FOR J=0 TO PSTV
30  SET (J,Y) : NEXT J
40  FOR J=100 TO PSTV STEP -1
50  SET (J,40) :NEXT J
60  FOR T=1 TO 350: NEXT T
70  CLS
75  REM SETTING INTEGRATED WAVEFORM
80  FOR X=PSTV TO 0 STEP -1
90  Y=Y+(30/PSTV)
100 SET(X,Y) :NEXT X
110 FOR X=100 TO PSTV STEP -1
120 Y=Y-(30/(101-PSTV))
130 SET (X,Y) :NEXT X
140 GOTO 140
```

Program B.

sharp. At the time of the picture taking, the Radio Shack monitor had died and I had resorted to my ATV monitor, which is as sharp as the proverbial tack. Score a plus for being a ham and a computer buff . . . it's a nifty combination.

The next series of comments applies to Program B. Line 10 gives you an instructional heading and allows you to input a desired value for the variable PSTV, which is the on time of the input square wave.

Lines 20 through 50 set the graphics points to have the square wave flat tops appear on the screen. It may be worth pointing out that in line 30 no specific quantity is attached to the Y dimension. The computer picks up this fact and assigns the value of ZERO, which is the value we want for the positive flat top of the square wave at the top left of the screen.

In line 50 we have put in a value of 40 (j,40) to set the negative side of the square wave and, since we want it to occupy the proper position, use the STEP-1 instruction to set it where it must go. Line 60 is the usual timing loop to keep the first graphics on the screen long enough to get a good visual impression. Eliminating line 70 will show both input and integrated waveforms on the screen together.

Lines 80 to 100 handle the setting of the points for graphics of the integrated waveform relating to the positive portion of the square wave. Notice that initially Y is ZERO and that each trip through this loop increments Y by the scaling factors set up in line 90.

The next loop in lines 110 to 130 handles the setting and scaling for the negative portion of the integrated waveform. Notice that the value of Y in line 120 is set by the last value from the first loop. This value is then decremented by the scaling factors in line 120.

The factor of 30 in the Y lines controls how "small" an on time will be accepted by the program. For instance, if the 30 is changed to 40, then the program will not accept an on time of less than six percent. With the factor remaining at 30, the program will accept on time as low as 1.3 percent, which is perfectly acceptable as it really pushes the resolution limit of the TRS-80 graphics.

In line 120, the factor 101 in the expression controls how large a value of on time will be accepted. Reducing this factor below 101 is not acceptable as 101 is the factor needed to accept an on time of 100 percent. Line 140 is a holding loop to keep the display whole when the program has run.

Since we are not using the full X axis capability (which is 127), you can actually enter an on time up to this limit, and the graphics will show just that—an on time of 127 units, which is meaningless.

If you wish to avoid this condition you may insert a line as follows: IF PSTV>100 PRINT "MAX ON TIME LIMIT IS 100": GOTO 10, which will take care of this matter nicely. This could be inserted as line 12 in Program B.

For a keener insight into what numbers the scaling factors are producing, you can experimentally insert two lines:

```
99 PRINT Y;
125 PRINT Y;
```

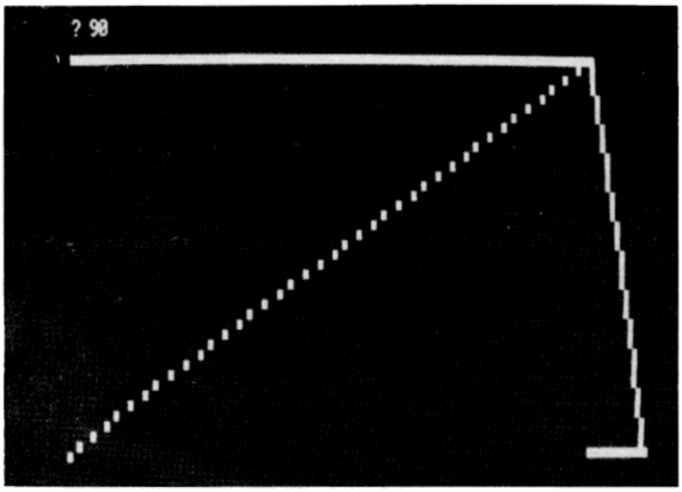
This will allow you to see how the value of Y is first incremented and then decremented to produce the graphics plotting points.

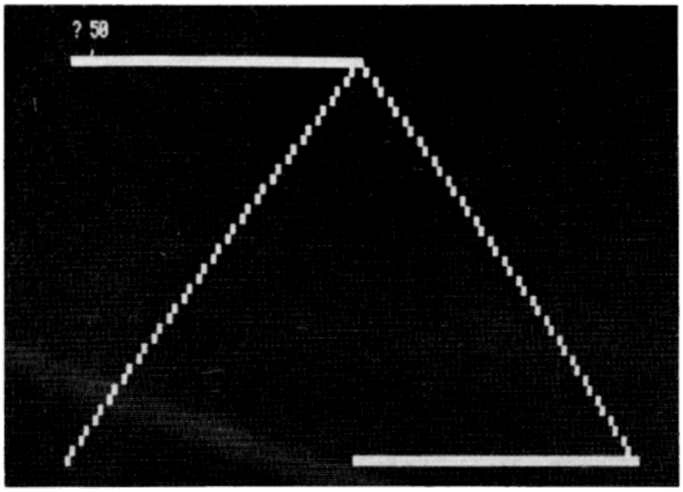
At this point you can see why the square wave is such a favorable waveform for this initial flight into integration. With its constant amplitude, the succeeding areas are all uniform and lend themselves to a simple expression for the successive additions as we go from integration point to integration point.

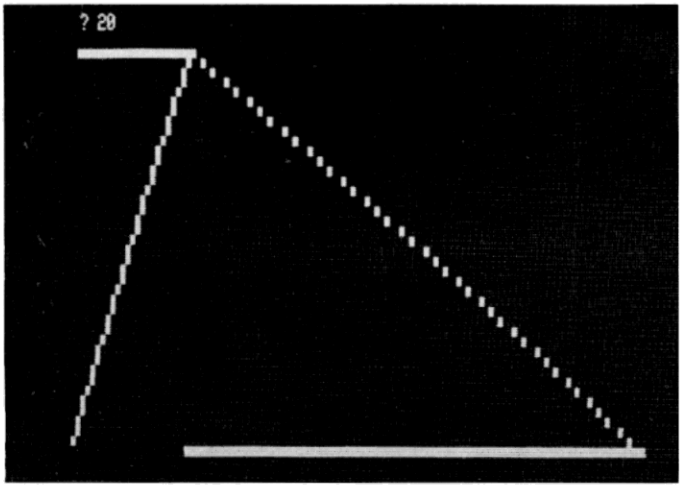
Conclusion

This simple exposition is another example of the versatility of your machine. Since the computer only does what you tell it to, its invitation is clear to expand your own horizons by acquiring new knowledge and then letting it help you implement that knowledge in a much more pleasant fashion than the old-fashioned Graphite Character Generator ever could.

You don't have to be a math buff to understand or enjoy certain concepts of geometry, trigonometry or calculus. However, coming across the right book helps. Way back in 1904, William Granville and Percy Smith authored a calculus volume that you can actually understand. Later on, books of this type became rather sophisticated, making calculus the property of the student. If you can find a used copy of this book, it may *just make calculus almost as* much fun as kissing your wife . . . or your computer! ■







Waveform results. *(Photos by Ira Joffe)*

(215) 665-1112

U-Design-It Software

For the Radio Shack TRS-80 with Optional Letter-Quality Output.

DATAMASTER—THE FLEXIBLE DATA BASE MANAGEMENT SYSTEM (DBMS)

- Designed for use by the professional or small businessman who is not a programmer
- Add, Delete, Search & Sort Random Data Files on disk
- Complex reporting features for reprinted forms
- User-specified screen formating for data input
- Reformat and add data fields to existing records
- Field to field scrolling (up and down)

COMPLETELY SELF-CONTAINED

- Provided with complete operating system and utilities
- Non-technical staff can operate immediately
- Self-prompting video screen
- Hand-holding backup and file utilities
- Power failure recovery of lost data

```
MICROTRONIX DATAMASTER VER. 2.0 - MAIN MENU
BACK - BACKUP DISKETTE               END  - END PROGRAM
CASE - CASE CONTROL                  FORM - ENTER/CHANGE FORMAT
CDAT - CONVERT DATA (1.4-2.0)        KILL - KILL FILE
COMB - COMBINE TWO DATA FILES        PREP - PRINT REPORT
COPY - COPY FILE                     RCVR - RECOVER INFORMATION
DENT - ENTER/CHANGE DATA             REFM - REFORMAT DATA FILE
DIR  - DISK DIRECTORY                RSEL - SELECT RECORDS
DREP - DEFINE REPORT                 SORT - SORT DATA FILE
? _
```

THE PROGRAMMER/OEM APPLICATION TOOL

- Complete custom applications in hours—not days
- Skeleton module for Data Base Access

- Custom packages can easily be adapted I.E., Mailing Lists, Payroll, Inventory, A/P, A/R, General Ledger, Medical & Dental Billings, Legal & Professional Cost Accounting.

HARDWARE REQUIREMENTS

- 48K TRS-80 MOD I or MOD II (Datamaster available 4/80) 2 Disk Drives (35, 40, or 77Trk)
- Parallel or Serial Printers (Optional)
- Lower Case Modification (Optional)

PRICE ONLY $500.00 (Special Introductory offer $299.00) INCLUDES:

- 2 Program diskettes
- 1 Sample data diskette (Inventory and mailing list examples)
- Looseleaf binder manual ($25.00 alone—applicable toward purchase)
- Dealer inquiries invited

COMPLETE TRS-80 COMPUTER SYSTEMS AVAILABLE

- Large selection of printers, featuring the new Olivetti ET221—4 Pitch Electronic Typewriter
- TRS-80 MOD I & MOD II

✓22 microtronix

Showrooms: 913 Walnut Street • Philadelphia, PA 19106

Microtronix Price Match Policy: We will match any currently advertised price providing the item is in stock. Shipping/handling: Software—$2.50 initial charge. $.50 each additional item. Add $2.00 for Air Service in the following states: AL, AK, AR, CA, FL, GA, HI, ID, IL, IN, IA, KS, KY, MI, MN, MT, NE, NV, NM, ND, OR, TN, TX, UT, WA, WI, WY. Hardware: Call for details. PA residents add 6%. All hardware is subject to availability and guaranteed by the manufacturer of the product. C.O.D.'s add $2.00.

✓ Reader Service—see page 147

Move it to a useful location in memory with minimal changes.

Get T-BUG High

Irwin Rappaport
24 Hemlock Hill Road
Upper Saddle River NJ 07458

T-BUGging with the TRS-80 is a fascinating exploration into the secret world of machine codes. With some previous knowledge of machine-language programming and the Z-80 (or the willingness to acquire this knowledge by doing some outside reading), you can expose some of the mystery and magic for what it really is . . . good, solid computer technology, of course!

But fascination can turn to frustration after the initial adventure. If you try to load in a BASIC program after loading T-BUG, then go back to the "bug" so you can satisfy your curiosity as to what the BASIC looks like, you will see . . . whoops . . . no more T-BUG! Loading in reverse order, with T-BUG going in last, results in the destruction of all but maybe a line or two of the BASIC. T-BUG is there, but with nothing to look at.

For some reason which escapes me, the Radio Shack people designed T-BUG to load into low area memory, just a few short bytes away from where everything else loads in as well. In the Level II machine, BASIC loads in at 42E9h (h signifies hex notation) while T-BUG begins at 4380h. Obviously, the two are incompatible except for 150 or so bytes. And T-BUG is useless for looking at or using along with the Editor-assembler program also, for very similar reasons.

Now those are merely frustrations for the mostly curious. The BUG's nest at 4380h can become a true annoyance when things start to become more serious, like when a machine-language program is to be used along with a BASIC program. Not only is program development hindered and debugging virtually impossible (even for the patient people of the world), but program loading and running can become more difficult, particularly if you would like to check registers, for example, or want to make revisions from time to time.

Block Transferring

There is no doubt that T-BUG must go elsewhere. With all those empty kilobytes in my 16K RAM and with that great memory-protect feature of Level II, why not bury it deep in high memory where it will be out of the way but still be able to function perfectly well?

So the project became "off-to-high-memory-with-T-BUG." Well, doing a block transfer is certainly no great trick, but you soon discover that there are many "intra-T-BUG" addresses in the program. It jumps and calls to the tune of 191 addresses, actually. Block transferring without making the necessary 191 internal changes is analogous to renumbering a BASIC program's lines but not making the respective changes in all the GOSUBs and GOTOs.

Having neither the patience to manually disassemble the thing (since my less-than-extensive knowledge of Z-80 code required looking up everything in the book) nor having patience or ability to write a disassembler, I looked to the one thing I did know—the excellent Level II BASIC.

First, I selected the entrance point of 7380h for the home of T-BUGI. That location offers at least three advantages: (1) plenty of working space for BASIC program applications in the 12K available from 42E9h to 7380h, (2) plenty of room beyond T-BUG HI for any machine-code routines I may want to call from BASIC and (3) more simplicity in creating the move procedure from 4380h to 7380h.

With the location decided upon, I used a direct block transfer to put T-BUG at its new position. Next, I wrote a PEEK/POKE routine in BASIC (see Fig. 1) to change all 43h through 49h values to the new 73h through 79h values starting at memory location 7380h (29568 decimal).

The program will execute fairly quickly, but the print statements will help to assure that something is happening. At the end, there should be a total of 213 locations POKEd—191 that we want and 22 that will have to be changed back again. The start PEEKing point is T-BUG HI decimal address of 29568.

Well, all of that worked fine, but T-BUG HI didn't, because some of those 43h-49h values were not high-order portions of addresses at all. How about that! . . . some were even a part of machine-code instructions. And naturally, those innocent, non-high order address bytes had to be restored to their original values. I used some PEEK/POKE trickery here, too, by making the preceding byte the action determinator, but still didn't get them all. I had to find the last two or three by pure debuggery and all the usual unorthodox methods.

Procedure

But the result of all this labor

```
10 INPUT "START PEEKING WHERE"; P
15 FOR L = 1 TO 1200
20 A = PEEK(P)
30 IF A<67 OR A>73 THEN 75
35 REM NOTE THAT VALUES ARE IN DECIMAL
40 A = A + 48
50 POKE P,A
60 K = K + 1:PRINT "JUST POKED LOCATION";P;"WITH VALUE";A
65 PRINT "SO FAR HAVE POKED":K;"LOCATIONS AND STILL PEEKING"
70 P = P + 1
75 NEXT L
80 PRINT:PRINT:PRINT "POKING AND PEEKING DONE. TOTAL OF";K;"LOCATIONS POKED"
```

Fig. 1. PEEK/POKE program to change 43h-49h to 73h-79h.

Hex address	Code	Remarks
5000	21	LD HL,NN
5001	80	Loads HL registers with the
5002	43	source of material to transfer
5003	11	LD DE,NN
5004	80	Loads DE registers with the
5005	73	destination of the transfer
5006	01	LD BC,NN
5007	B0	Loads BC registers with number
5008	04	of bytes (in hex) to transfer
5009	ED	LDI Operation to do the
500A	A0	transfer and decrement counter BC
500B	EA	JP PE,NN
500C	09	Jumps to NN (5009) if Parity
500D	50	flag set (BC not zero)
500E	00	Non-op, end
500F	CD	Back to T-BUG
5010	80	when transfer is completed
5011	43	(same as setting breakpoint)

Fig. 3. Corrections to restore values.

Hex address	After PEEK/POKE	Must be changed to	Hex address	After PEEK/POKE	Must be changed to
7381	73	43	7676	77	47
7389	73	43	76A6	77	47
73E6	76	46	76B4	77	47
73FD	77	47	76DB	77	47
74E9	74	44	76E1	73	43
74EF	73	43	771A	73	43
74F8	76	46	772C	73	43
74FE	75	45	7734	75	45
750A	73	43	773A	76	46
758C	77	47	7743	77	47
75AD	77	47	7776	77	47

Fig. 2. Block transfer routine.

is that T-BUG HI does exactly everything that the original T-BUG does and operates in precisely the same way, with the big difference that the entrance point after loading is 29568 (decimal) instead of 17280 (decimal).

At this location it does not interfere with any program or technique that I would have occasion to use.

To get your T-BUG high, you can make all the 191 changes manually, but I would suggest the following method:

(1) Load T-BUG as usual.
(2) Enter the block transfer routine (see Fig. 2) using T-BUG, starting at 5000h.
(3) Execute the transfer program with #J5000 from T-BUG.
(4) Take a look at 7380h to see that the transfer went well, then reset to return to Level II BASIC.
(5) Enter a PEEK/POKE BASIC program, such as Fig. 1, to change all 43h-49h to 73h-79h, respectively. Start at 7380h (29568 decimal) and go for 1200 bytes (that's a few extra for good measure) and remember to use decimal when working in Level II BASIC and hex when using T-BUG.
(6) Run the PEEK/POKE program and then reload the regular T-BUG.
(7) Now refer to the chart (Fig. 3) so that you can use regular T-BUG (which I call T-BUG LO) to correct new T-BUG HI.
(8) Use #M to address and correct each of the 22 locations indicated.
(9) T-BUG HI should now be operational. Try #J7380. You should be in T-BUG HI, with all rights and privileges.
(10) Jump back to T-BUG LO and use it to save the HI on a fresh tape. ■

A step-by-step procedure for converting your Radio Shack Editor/Assembler from tape to disk.

EDTASM on Disk

Robert Butler
Wayne Butler
25 Magford St.
Marblehead MA 01945

We recently purchased a mini-disk drive system for our TRS-80 and found it to be much faster and more efficient than the tape system. However, this meant converting our programs from tape to disk.

Most of the tape programs were not any problem to convert, but for a time the Editor/Assembler (E/A) program, which we purchased from Radio Shack, had us stumped. Finally, my son solved it. This procedure is not intended to pirate the E/A program of Radio Shack, but rather to make the program more convenient to the user.

The problem was that the memory used for the E/A program overlapped the memory used for the disk operating system (DOS).

The solution was to enter the E/A program, move it to a clear section of memory and add a program to move it back to the original memory locations. Enter both onto the disk. When the program is retrieved from the disk, the added program will put it in its proper memory location and start it automatically. Of course, this will cause a loss of part of TRSDOS.

Steps to Implement Solution

1. Turn on the computer with DOS. The display will look like this:

DOS READY

2. Enter DEGUG.
3. Press BREAK key.
4. Type D6580. This will display the location where you want to put the program, which will move the E/A program to a clear memory location.
5. Type M6580 (SPACE). This will allow you to change the contents of this and the next memory locations.
6. Type in Program A, which is the machine-language program to move the E/A program to a new location.
7. To get out of DEBUG, type G402D.
8. Then type DEBUG (OFF) to turn off DEBUG.
9. Enter BASIC2. Use Level II rather than disk BASIC so that the clock interrupt will not interfere with loading E/A tape.
10. Type SYSTEM (ENTER).
11. Type EDTASM (ENTER). This will load the E/A program into memory from tape.
12. When the program has been successfully loaded, type /25984. This activates the moving program to move the E/A program into the other section of memory. This works very fast.
13. Press the RESET button at left rear of computer. This puts you into the disk operating system without erasing any of the memory you have just set up.
14. Enter DEBUG again.
15. Display 7FF0 by typing D7FF0 (ENTER).
16. Now type in Program B. This adds to the E/A program an additional program that will move it back to the original location so that it will work properly.
17. Type G402D to return to DOS from DEBUG again.
18. Type DEBUG (OFF) to turn off DEBUG.
19. Type TAPEDISK. This has a dump routine that will dump to the disk starting locations under 7000 hex, unlike the dump routine in DOS, which will not.
20. After getting a "?" type in:

F EDTASM/CMD:0 65B0 7FFF 7FF0 (ENTER)

This will put everything between memory locations 65B0 and 7FFF onto the disk and give a starting address of 7FF0.

21. When you get a "?" again, type E (ENTER) to exit the TAPEDISK program.
22. Now the E/A program is on the disk with a name of EDTASM and may be recalled by using that name as with any other program on the disk. ■

```
6580            00100           ORG  6580H
1A19            00110   BASIC2  EQU  1A19H
6580   210043   00120           LD   HL,4300H   ;SOURCE ADDRESS
6583   11B065   00130           LD   DE,65B0H   ;DEST. ADDRESS
6586   01401A   00140           LD   BC,1A40H   ;PROGRAM LENGTH
6589   EDB0     00150           LDIR            ;MOVE THE PROGRAM
658B   C3191A   00160           JP   BASIC2     ;RETURNS TO A READY
0000            00170           END
00000  TOTAL    ERRORS

BASIC2 1A19
```

Program A.

```
7FF0            00100           ORG  7FF0H
468A            00110   EDTASM  EQU  468AH      ;START ADDRESS PROGRAM
7FF0   21B065   00120           LD   HL,65B0H   ;SOURCE ADDRESS
7FF3   110043   00130           LD   DE,4300H   ;DEST. ADDRESS
7FF6   01401A   00140           LD   BC,1A40H   ;LENGTH OF PROGRAM
7FF9   EDB0     00150           LDIR            ;MOVE THE PROGRM
7FFB   C38A46   00160           JP   EDTASM     ;GOTO THE PROGRAM
0000            00170           END
00000  TOTAL    ERRORS

EDTASM 468A
```

Program B.

UP TO 25% OFF

YOUR OWN TRS-80 SYSTEM AT TREMENDOUS SAVINGS

① TRS-80 Complete System

Includes: CPU/Keyboard, Power Supply, Video Monitor, Cassette Recorder, Manual, and Game Cassette.

② Line Printer
③ Mini Disk System
④ C-10 Cassettes
⑤ Verbatim Diskettes

DISK DRIVES IN STOCK!

ITEM		REG. PRICE	OUR PRICE
Level II—4k		$619.00	$575.70
Level II—16k		$849.00	$789.60
Expansion Interface		$299.00	$278.10
Mini Disk Drive		$ 495.00	$ 385.00
Centronics 730 Printer		$ 995.00	$ 850.00
Centronics 101 Printer		$1595.00	$1400.00
Anadex DP-8000 Printer		$ 895.00	$ 995.00
Memory Kit (16K) FREE INSTALLATION		$ 149.00	$ 98.00
Verbatim Diskettes ea.		$ 5.95	$ 4.95
	3	$ 17.89	$ 12.00
	10	$ 59.00	$ 37.00
C-10 Cassettes	5	$ 4.95	$ 4.50
	25	$ 24.75	$ 18.75
Paper (9½" x 11" fanfold, 3500 sheets)		$ 35.00	$ 29.95

SALE

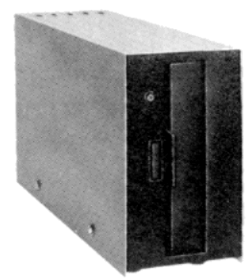
Centronics 779 $995.00
Same as Line Printer I (Tractor)

TRS—80 MODEL II $ $3626.00
- **64K RAM**
- **½ MEG DISK**

ADDITIONAL DISK DRIVE (1ST) $1069.50
ADDITIONAL DISK DRIVE (2ND + 3RD) $ 558.00

MINI DISK DRIVES NOW $385

Over $100 less than Radio Shack's!



There are new developments every day—write or call for the latest information.

VRData corp

777 Henderson Boulevard N-6
Folcroft Industrial Park
Folcroft PA 19032
(215) 461-5300 ✓31

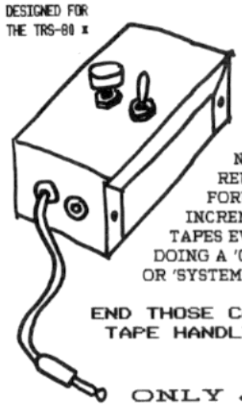
master charge
VISA

TOLL FREE
1-(800) 345-8102 *Orders only!

FOREIGN and DOMESTIC DISTRIBUTORSHIPS AVAILABLE . . .

From The Engineering Group At GSIC:

DESIGNED FOR THE TRS-80 *



CASSETTE REWINDER INTERFACE

NOW YOU CAN REWIND, FAST-FORWARD, OR INCREMENT YOUR TAPES EVEN WHEN NOT DOING A 'CLOAD', 'CSAVE' OR 'SYSTEM' COMMAND.

END THOSE CASSETTE TAPE HANDLING BLUES!

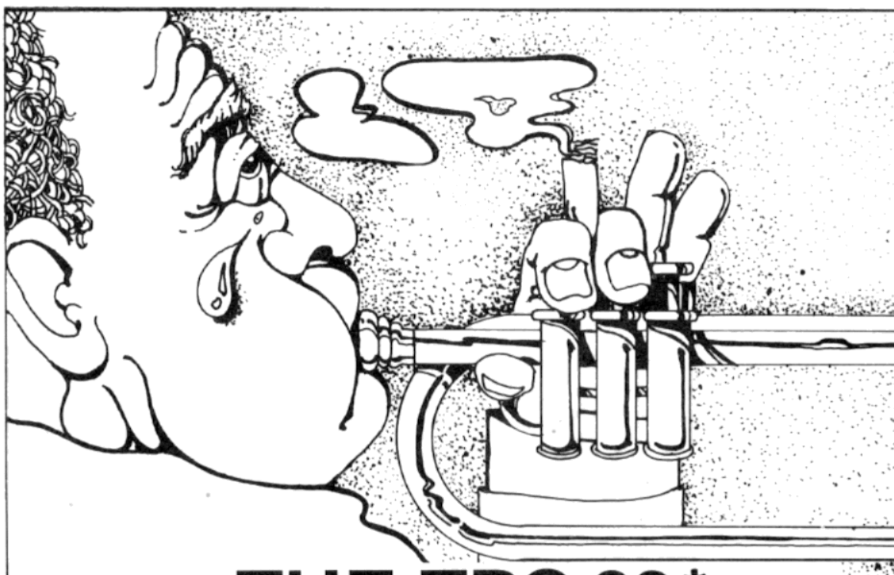
ONLY .. $14.95

WE PAY ALL SHIPPING AND HANDLING CHARGES.

FEATURES: Adapts instantly to present system, precision toggle and momentary push-button switches, durable painted aluminum cabinet, high quality micro-plug and jack, non-slip rubber feet, does not interfere with normal system operation, compact design complements your system.

SEND CHECK OR MONEY ORDER TO: ✓93
GRANITE STATE INSTRUMENT COMPANY
Box 3486
Nashua, NH 03061

(* TRADEMARK OF TANDY CORPORATION)



THE TRS-80* BLUES.

Do your TRS 80'S limitations hit a sour note? Our full line of business-oriented software can change that tune.
They're easy to use, modularly designed, and superiorly documented, giving you versatility and sophistication of a higher scale, at a price you'll sing about.
So write us, and we'll send you the score.

SMALL BUSINESS SYSTEMS GROUP
GROTON STREET, DUNSTABLE, MA. 01827 (617) 649-9595 ✓18

*TRS-80 is a registered trade mark of Radio Shack, a division of Tandy Corp.

Analyze your cassette and adjust it with this neat utility program: Eliminate CLOAD Problems.

TTape

Dennis Stevens
10895 Kemah Lane
San Diego CA 92131

Are you having difficulty reading tapes into your TRS-80? Are you uncertain whether the fault lies with the cassette recorder or the computer, or whether the tape errors occur when you are writing onto tape or reading from tape? Is the recorder head dirty or misaligned? Maybe the tape speed is wrong.

TTAPE will not always provide direct answers to these questions, but it will almost instantly feed back an evaluation of a cassette adjustment. The real-time display of the error rate and the type of errors (e.g., dropped bits) being fed from tape to the CPU will usually tie down the trouble spot.

Present Methods

In contrast, presently used methods suffer from the inability to detect low error rates, or they are not diagnostic of deteriorating cassette conditions, or they are clumsy and time-consuming.

One hardware approach to solving cassette problems involves playing an AM radio positioned over the TRS-80 label on the right of the keyboard while loading data from cassette to the computer. When the cassette volume is too high, the radio emits a steady, high-pitched noise. (It turns out that this noise corresponds to reading a string of hexadecimal FFs into the computer. If the cassette data happens to be a string of hex FFs, the radio method will not work.) At lower cassette volume settings, you hear a lower pitched, random noise, which corresponds to good code entering the computer.

In between these conditions are narrow transition regions where random errors are entering the computer. If, for some reason, you are entering a few percent bad data (or even very high error rates if the bad data is random in nature), this loading condition will go unnoticed by the radio method, and yet your program will almost certainly fail.

A similar but less diagnostic tool is an ac voltmeter connected in parallel with the cassette output. This allows setting the cassette volume to a predeter-

Listing 1. TTAPE.

```
4F00            00100          ORG    4F00H
4F00 AF         00110 WTTAPE   XOR    A
4F01 CD1202     00120          CALL   212H    ;TURN ON CASSETTE
4F04 3EAA       00130 LBL100   LD     A,0AAH
4F06 CD6402     00140          CALL   264H    ;WRITE ONE BYTE ONTO TAPE
4F09 C3044F     00150          JP     LBL100  ;END BY RESET
4F0C 21003C     00160 RTTAPE   LD     HL,3C00H
4F0F 222040     00170          LD     (4020H),HL ;HOME THE CURSOR
4F12 E5         00180          PUSH   HL
4F13 AF         00190          XOR    A
4F14 CD1202     00200          CALL   212H    ;TURN ON THE CASSETTE
4F17 CD3502     00210 LBL101   CALL   235H    ;READ ONE BYTE FROM TAPE
4F1A 08         00220          EX     AF,AF'  ;SAVE BYTE IN A'
4F1B E1         00230          POP    HL      ;RETRIEVE PREVIOUS CURSOR ADDRESS
4F1C 7D         00240          LD     A,L
4F1D FEFC       00250          CP     0FCH
4F1F C22E4F     00260          JP     NZ,SKIP ;JUMP ON L<>0FCH
4F22 7C         00270          LD     A,H
4F23 FE3F       00280          CP     3FH
4F25 C22E4F     00290          JP     NZ,SKIP ;JUMP ON HL<>3FFCH
4F28 06CD       00300          LD     B,0CDH
4F2A 10FE       00310          DJNZ   $       ;DELAY 3.3 MSEC ON HL=3FFCH
4F2C 10FE       00320          DJNZ   $
4F2E 2A2040     00330 SKIP     LD     HL,(4020H) ;SAVE PRESENT CURSOR ADDRESS
4F31 E5         00340          PUSH   HL
4F32 08         00350          EX     AF,AF'  ;RETRIEVE BYTE READ FROM TAPE
4F33 CB47       00360          BIT    0,A     ;TEST BIT 0
4F35 CA3C4F     00370          JP     Z,LBL102 ;JUMP ON BIT 0 LOW
4F38 06CD       00380          LD     B,0CDH
4F3A 10FE       00390          DJNZ   $       ;DELAY 1.5 MSEC WHEN BYTE IS ODD
4F3C CD404F     00400 LBL102   CALL   HEXDSP  ;PUSH PC
0001            00410          DEFS   1       ;RESERVE A ONE BYTE SPACE
4F40 E1         00420 HEXDSP   POP    HL      ;HL POINTS TO RESERVED BYTE
4F41 77         00430          LD     (HL),A  ;BYTE IS AT RESERVED ADDRESS
4F42 0602       00440          LD     B,2
4F44 AF         00450 HXDP1    XOR    A
4F45 ED6F       00460          RLD            ;UNPACK ONE NIBBLE
4F47 C630       00470          ADD    A,30H   ;HEX TO ASCII
4F49 FE3A       00480          CP     3AH
4F4B FA504F     00490          JP     M,HXDP2 ;JUMP ON DIGIT BETWEEN 0 & 9
4F4E C607       00500          ADD    A,7     ;ADD ON DIGIT BETWEEN A & F
4F50 CD3300     00510 HXDP2    CALL   33H     ;DISPLAY AT CURSOR
4F53 10EF       00520          DJNZ   HXDP1
4F55 C3174F     00530          JP     LBL101  ;READ NEXT BYTE
4F0C            00540          END    RTTAPE
00000 TOTAL ERRORS
HXDP2   4F50
HXDP1   4F44
HEXDSP  4F40
LBL102  4F3C
SKIP    4F2E
LBL101  4F17
RTTAPE  4F0C
LBL100  4F04
WTTAPE  4F00
```

mined "acceptable" level.

Another hardware method presently in use consists of using an automatic level control on the cassette output. This is essentially an automatic version of the above voltmeter method and is an excellent means for compensating for the different volumes at which tapes from different sources invariably are recorded. But this method will not diagnose an incorrect tape speed or a misaligned head. Nor is it diagnostic of anything, in reality.

Radio Shack has provided some firmware to help load from tape correctly. Machine-language tapes can be loaded in the monitor mode. If a loading error occurs, it will almost always be caught as a checksum error revealed by a C in the upper right-hand corner of the monitor screen. This is very valuable and works very well—as long as there are no errors.

If you get a checksum error and you suspect that the cassette volume setting is too low, for example, then after adjusting the cassette volume, it is necessary to repeat the loading procedure from the beginning. The checksum indicator cannot be made to appear and disappear as the cassette is adjusted. Repeating the loading procedure for each cassette adjustment can be very time-consuming, which renders this method impractical for use as a diagnostic tool.

The checksum is glaringly absent from the loading procedure in the BASIC mode. Level II has a CLOAD? command, which essentially eliminates the writing of undetected errors onto tape. However, this command provides very little help in detecting loading errors unless a correct copy of the code already resides in memory.

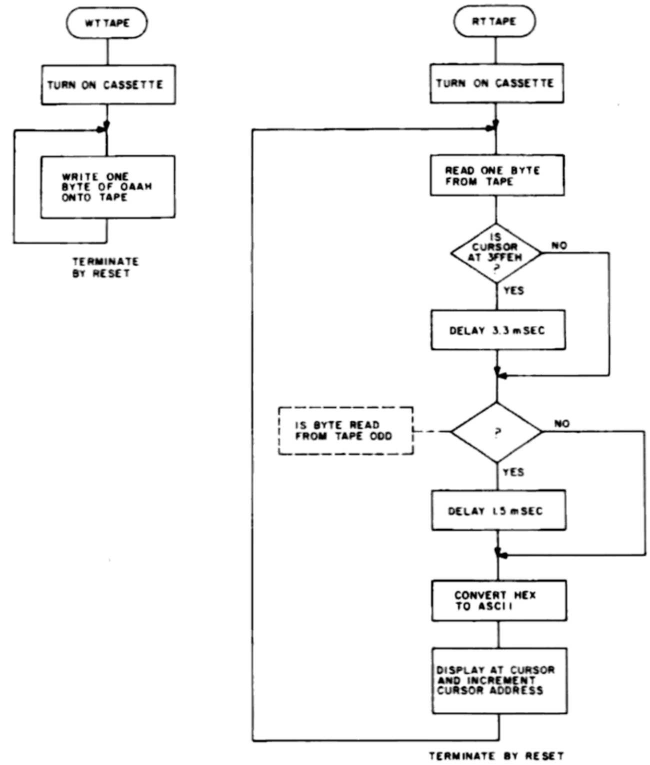


Fig. 1. TTAPE flowchart.

Listing 3. Expanded TTAPE.

```
DGDGJ4DC9         00100          ORG     4DC9H
4DC9 7E           00200 DSPLY    LD      A,(HL)  ;HL POINTS TO CHARACTER TO BE DISPLAYED
4DCA CD3300       00300          CALL    33H     ;DISPLAY CHARACTER
4DCD CD3300       00400          CALL    33H     ;INC CURSOR FOR 32 CH./LINE
4DD0 23           00500          INC     HL
4DD1 10F6         00600          DJNZ    DSPLY   ;B=# BYTES TO BE DISPLAYED
4DD3 C9           00700          RET
4DD4 3E20         00800 SPACE    LD      A,20H   ;BLANK CHARACTER
4DD6 CD3300       00900          CALL    33H     ;DISPLAY BLANK
4DD9 10F9         01000          DJNZ    SPACE   ;B=# OF SPACES TO BE DISPLAYED
4DDB C9           01100          RET
4DDC 21003C       01200 CLRSCN   LD      HL,3C00H
4DDF 11013C       01300          LD      DE,3C01H
4DE2 010004       01400          LD      BC,400H
4DE5 3620         01500          LD      (HL),20H ;BLANK CHARACTER
4DE7 EDB0         01600          LDIR             ; BLANK OUT SCREEN
4DE9 C9           01700          RET
4DEA CDDC4D       01800 TTAPE    CALL    CLRSCN  ;CLEAR THE SCREEN
4DED 3E08         01900          LD      A,8
4DEF D3FF         02000          OUT     (0FFH),A ;32 CH./LINE, CASSETTE OFF
4DF1 21003C       02100          LD      HL,3C00H
4DF4 222040       02200          LD      (4020H),HL ;HOME CURSOR
4DF7 061A         02300          LD      B,1AH
4DF9 CDD44D       02400          CALL    SPACE   ;26 SPACES
4DFC CD194F       02500          CALL    LETTER  ;PUSH PC
4DFF 54           02600          DEFM    'TTAPEREAD'
 4E00 54    4E01 41    4E02 50    4E03 45    4E04 52    4E05 45    4E06 41    4E07 44             02605
;LINE FEED
4E08 53           02610          DEFM    'S & WRITE'
 4E09 20    4E0A 26    4E0B 20    4E0C 57    4E0D 52    4E0E 49    4E0F 54    4E10 45             02615
;LINE FEED
4E11 53           02620          DEFM    'S A TEST '
 4E12 20    4E13 41    4E14 20    4E15 54    4E16 45    4E17 53    4E18 54    4E19 20             02625
;LINE FEED
4E1A 54           02630          DEFM    'TAPEBY D'
 4E1B 41    4E1C 50    4E1D 45    4E1E 42    4E1F 59    4E20 20    4E21 20    4E22 44             02635
;LINE FEED
4E23 45           02640          DEFM    'ENNIS STE'
 4E24 4E    4E25 4E    4E26 49    4E27 53    4E28 20    4E29 53    4E2A 54    4E2B 45             02645
;LINE FEED
4E2C 56           02650          DEFM    'VENS'
 4E2D 45    4E2E 4E    4E2F 53                02655 ;LINE FEED
4E30 54           02800          DEFM    'TO WRITE '
 4E31 4F    4E32 20    4E33 57    4E34 52    4E35 49    4E36 54    4E37 45    4E38 20             02802
;LINE FEED
4E39 54           02804          DEFM    'THE TEST '
 4E3A 48    4E3B 45    4E3C 20    4E3D 54    4E3E 45    4E3F 53    4E40 54    4E41 20             02806
;LINE FEED
4E42 44           02808          DEFM    'DATA ONTO'
 4E43 41    4E44 54    4E45 41    4E46 20    4E47 4F    4E48 4E    4E49 54    4E4A 4F             02810
;LINE FEED
4E4B 20           02812          DEFM    ' TAPEFIRS'
 4E4C 54    4E4D 41    4E4E 50    4E4F 45    4E50 46    4E51 49    4E52 52    4E53 53             02814
;LINE FEED
4E54 54           02816          DEFM    'T MAKE TH'
 4E55 20    4E56 4D    4E57 41    4E58 4B    4E59 45    4E5A 20    4E5B 54    4E5C 48             02818
;LINE FEED
4E5D 45           02820          DEFM    'E RECORDE'
 4E5E 20    4E5F 52    4E60 45    4E61 43    4E62 4F    4E63 52    4E64 44    4E65 45             02822
;LINE FEED
4E66 52           02824          DEFM    'R READY,  '
 4E67 20    4E68 52    4E69 45    4E6A 41    4E6B 44    4E6C 59    4E6D 2C    4E6E 20    4E6F 20
        02826 ;LINE FEED
```

```
4E70 54      03000        DEFM   'THEN PRES'
 4E71 48     4E72 45      4E73 4E     4E74 20     4E75 50     4E76 52     4E77 45     4E78 53          03002
;LINE FEED
4E79 53      03004        DEFM   'S W'
 4E7A 20     4E7B 57                  03006 ;LINE FEED
4E7C 54      03200        DEFM   'TO READ T'
 4E7D 4F     4E7E 20      4E7F 52     4E80 45     4E81 41     4E82 44     4E83 20     4E84 54          03202
;LINE FEED
4E85 48      03204        DEFM   'HE TEST T'
4E86 45      4E87 20      4E88 54     4E89 45     4E8A 53     4E8B 54     4E8C 20     4E8D 54          03206
;LINE FEED
4E8E 41      03208        DEFM   'APE, FIRS'
4E8F 50      4E90 45      4E91 2C     4E92 20     4E93 46     4E94 49     4E95 52     4E96 53          03210
;LINE FEED
4E97 54      03212        DEFM   'T    MAKE'
4E98 20      4E99 20      4E9A 20     4E9B 20     4E9C 4D     4E9D 41     4E9E 4B     4E9F 45          03214
;LINE FEED
4EA0 20      03216        DEFM   ' THE RECO'
 4EA1 54     4EA2 48      4EA3 45     4EA4 20     4EA5 52     4EA6 45     4EA7 43     4EA8 4F          03218
;LINE FEED
4EA9 52      03220        DEFM   'RDER READ'
 4EAA 44     4EAB 45      4EAC 52     4EAD 20     4EAE 52     4EAF 45     4EB0 41     4EB1 44          03222
;LINE FEED
4EB2 59      03224        DEFM   'Y, THEN'
 4EB3 2C     4EB4 20      4EB5 54     4EB6 48     4EB7 45     4EB8 4E                 03226 ;LINE FEED
4EB9 20      03400        DEFM   '   PRESS R'
4EBA 20      4EBB 20      4EBC 50     4EBD 52     4EBE 45     4EBF 53     4EC0 53     4EC1 20     4EC2 52
      03500 ;REMARK FOR LINE FEED
4EC3 54      03600        DEFM   'TO RE-ENT'
 4EC4 4F     4EC5 20      4EC6 52     4EC7 45     4EC8 2D     4EC9 45     4ECA 4E     4ECB 54          03602
;LINE FEED
4ECC 45      03604        DEFM   'ER TTAPE '
 4ECD 52     4ECE 20      4ECF 54     4ED0 54     4ED1 41     4ED2 50     4ED3 45     4ED4 20          03606
;LINE FEED
4ED5 46      03608        DEFM   'FROM BASI'
 4ED6 52     4ED7 4F      4ED8 4D     4ED9 20     4EDA 42     4EDB 41     4EDC 53     4EDD 49          03610
;LINE FEED
4EDE 43      03612        DEFM   'C II,ENTE'
 4EDF 20     4EE0 49      4EE1 49     4EE2 2C     4EE3 45     4EE4 4E     4EE5 54     4EE6 45          03614
;LINE FEED
4EE7 52      03616        DEFM   'R SYSTEM '
 4EE8 20     4EE9 53      4EEA 59     4EEB 53     4EEC 54     4EED 45     4EEE 4D     4EEF 20          03618
;LINE FEED
4EF0 41      03620        DEFM   'AND THEN '
 4EF1 4E     4EF2 44      4EF3 20     4EF4 54     4EF5 48     4EF6 45     4EF7 4E     4EF8 20          03622
;LINE FEED
4EF9 45      03624        DEFM   'ENTER'
4EFA 4E      4EFB 54      4EFC 45     4EFD 52                 03626 ;LINE FEED
4EFE 2F      03800        DEFM   '/19946 AF'
 4EFF 31     4F00 39      4F01 39     4F02 34     4F03 36     4F04 20     4F05 41     4F06 46          03802
;LINE FEED
4F07 54      03804        DEFM   'TER THE S'
 4F08 45     4F09 52      4F0A 20     4F0B 54     4F0C 48     4F0D 45     4F0E 20     4F0F 53          03806
;LINE FEED
4F10 59      03808        DEFM   'YMBOLS *?'
 4F11 4D     4F12 42      4F13 4F     4F14 4C     4F15 53     4F16 20     4F17 2A     4F18 3F          03810
;LINE FEED
4F19 E1      04000 LETTER POP    HL      ;HL POINTS TO 1ST CHARACTER
4F1A 0605    04100        LD     B,5
4F1C CDC94D  04200        CALL   DSPLY   ;DISPLAY 1ST LINE
4F1F 0622    04300        LD     B,22H
4F21 CDD44D  04400        CALL   SPACE   ;34 SPACES
4F24 061A    04500        LD     B,1AH
4F26 CDC94D  04600        CALL   DSPLY   ;2ND LINE
4F29 0612    04700        LD     B,12H
4F2B CDD44D  04800        CALL   SPACE   ;18 SPACES
4F2E 0612    04900        LD     B,12H
4F30 CDC94D  05000        CALL   DSPLY   ;3RD
4F33 0650    05100        LD     B,50H
4F35 CDD44D  05200        CALL   SPACE   ;80 SPACES
4F38 064C    05300        LD     B,4CH
4F3A CDC94D  05400        CALL   DSPLY   ;4TH,5TH & 6TH
4F3D 0668    05500        LD     B,68H
4F3F CDD44D  05600        CALL   SPACE   ;104 SPACE
4F42 0647    05700        LD     B,47H
4F44 CDC94D  05800        CALL   DSPLY   ;7TH, 8TH & 9TH
4F47 0672    05900        LD     B,72H
4F49 CDD44D  06000        CALL   SPACE   ;114 SPACES
4F4C 063B    06100        LD     B,3BH
4F4E CDC94D  06200        CALL   DSPLY   ;10TH & 11TH
4F51 060A    06300        LD     B,0AH
4F53 CDD44D  06400        CALL   SPACE   ;10 SPACES
4F56 061B    06500        LD     B,1BH
4F58 CDC94D  06600        CALL   DSPLY   ;12TH
4F5B CD4900  06700 LBL113 CALL   49H     ;WAIT FOR KEYSTROKE INPUT
4F5E FE52    06800        CP     52H     ;IS IT AN R?
4F60 CAB14F  06900        JP     Z,RTTAPE ;JP ON R
4F63 FE57    07000        CP     57H     ;IS IT A W?
4F65 C25B4F  07100        JP     NZ,LBL113 ;JP ON NEITHER R NOR W
```

Method of Approach

The program TTAPE is not intended to detect errors in actual code being loaded from any tape in general. Instead, it reads and displays test data written on a tape specifically for test purposes. The program starts with a very short section that writes a string of hex AAs onto the tape. This string of test data is terminated manually by pushing the reset button.

The rest of the program reads the test data (see the flowchart in Fig. 1). After each byte of test data is read from the tape, the cursor position is tested. If the cursor is at address 3FFF hex, then a 3.3 ms delay will be incurred. At this cursor address, a scrolling of the display is imminent. This will cause a delay that will drop a bit from the first byte read after the scroll. I found by trial and error that an additional 3.3 ms delay dropped enough additional bits to regain the hex AA data sequence.

Next, the byte is tested to determine whether it is odd or even. If it is odd, then a delay of 1.5 ms will be incurred. If one bit has been dropped during loading, then the data will appear to consist of a string of hex 55s. The 1.5 ms delay causes the loss of an additional bit, which converts the string back to hex AAs. The byte is then converted to ASCII and displayed at the cursor. The procedure is then repeated until interrupted by the reset.

```
10 DATA 175,205,18,2,62,170,205,100,2,195
4,79,33,0,60,34,32,64,229,175,205,18,2,
205,53,2,8,225,125,254,252
20 DATA 194,46,79,124,254,63,194,
46,79,6,205,16,254,16,254,42,32,
64,229,8,203,71,202,60,79,6,205,16,254
30 DATA 205,64,79,0,225,119,6,2,175,237,
111,198,48,254,58,250,80,79,198,7,205,
51,0,16,239,195,23,79
40, POKE 16553,255
50 FOR X = 0 TO 87 : READ BYTE
60 POKE 20224 + X,BYTE : NEXT
70 END
```

Listing 2. BASIC program to POKE TTAPE into memory. (Note that each of the three data statements requires an intermediate line feed. Use the enter key only at the end of the statement.)

Entering and Using TTAPE

The program is loaded into the computer from tape by entering SYSTEM and then entering TTAPE after the cassette is made ready and is put in the play mode. If you can't load TTAPE from tape, try loading it from T-BUG using the hex listing (Listing 1). If you can't even load T-BUG from tape, you will have to POKE the decimal code in from BASIC. Now you can see that TTAPE's brevity (88 bytes) adds to its utility. A brief BASIC program that will do the job is shown in Listing 2. The DATA and READ statements are used to allow proofreading the decimal code before it is poked.

After TTAPE is loaded, it can be executed by entering SYSTEM and then entering /20224 to write the test data onto tape, or /20236 to read and display test data. I use a tape loaded with 45 minutes of test data directly following the TTAPE program itself.

When you are reading test data, the most obvious knob to play with is the cassette volume control. You will notice that at very high volume settings the computer is being loaded with hex FFs. At lower volumes, there is a narrow transition where random errors are being read, and then correct data consisting of hex AAs is read below that volume setting. This transition occurs at a volume setting of about 7 on my Radio Shack CTR-41. Another transition is observed at a volume setting of about 2, where again, random errors are read in. Just before data is cut off completely, the data becomes mostly zeros.

The tape head can be aligned by setting the volume just above the lower transition. Then adjust the head alignment screw until it is about halfway between the positions at which errors appear.

The effects of changing the tape speed are similar to the effects of changing the volume setting. At high tape speeds, hex FFs are read in, and at low speeds, zeros are read in. I also observed that errors could be introduced by sharply tapping the recorder directly over the

```
4F68 CDDC4D   07200 WTTAPE  CALL   CLRSCN  ;CLEAR THE SCREEN
4F6B AF       07300         XOR    A
4F6C CD1202   07400         CALL   212H    ;TURN ON CASSETTE
4F6F 11403C   07500         LD     DE,3C40H
4F72 CD944F   07600         CALL   LBL114  ;PUSH PC
4F75 57       07700         DEFM   'WRITING T'
4F76 52       4F77 49       4F78 54      4F79 49      4F7A 4E      4F7B 47      4F7C 20      4F7D 54      07702
;LINE FEED
4F7E 48       07704         DEFM   'HE TEST D'
4F7F 45       4F80 20       4F81 54      4F82 45      4F83 53      4F84 54      4F85 20      4F86 44      07706
;LINE FEED
4F87 41       07708         DEFM   'ATA ONTO '
4F88 54       4F89 41       4F8A 20      4F8B 4F      4F8C 4E      4F8D 54      4F8E 4F      4F8F 20      07710
;LINE FEED
4F90 54       07712         DEFM   'TAPE'
4F91 41       4F92 50       4F93 45                   07714 ;LINE FEED
4F94 E1       07900 LBL114  POP    HL      ;POINTS TO 1ST CHARACTER
4F95 011F00   08000         LD     BC,1FH
4F98 EDB0     08100         LDIR           ;MESSAGE TO SCREEN
4F9A 1604     08200         LD     D,4     ;WAIT 3 SEC. FOR TAPE TO COME UP  TO SPEED
4F9C 010000   08300 LBL115  LD     BC,0
4F9F 0D       08400 LBL116  DEC    C
4FA0 C29F4F   08500         JP     NZ,LBL116
4FA3 10FA     08600         DJNZ   LBL116
4FA5 15       08700         DEC    D
4FA6 C29C4F   08800         JP     NZ,LBL115
4FA9 3EAA     08900 LBL117  LD     A,0AAH  ;BYTE TO BE RECORDED
4FAB CD6402   09000         CALL   264H    ;WRITE ONE BYTE
4FAE C3A94F   09100         JP     LBL117  ;END BY PUSHING RESET BUTTON
4FB1 CDDC4D   09200 RTTAPE  CALL   CLRSCN  ;CLEAR THE SCREEN
4FB4 21003C   09300         LD     HL,3C00H
4FB7 222040   09400         LD     (4020H),HL ;HOME THE CURSOR
4FBA AF       09500         XOR    A
4FBB CD1202   09600         CALL   212H    ;TURN ON CASSETTE
4FBE E5       09700         PUSH   HL
4FBF CD3502   09800 LBL118  CALL   235H    ;READ A BYTE
4FC2 08       09900         EX     AF,AF'
4FC3 E1       10000         POP    HL      ;RETRIEVE PREVIOUS CURSOR POSITION
4FC4 7D       10100         LD     A,L
4FC5 FEFC     10200         CP     0FCH
4FC7 C2D64F   10300         JP     NZ,SKIP ;JUMP ON L<>FCH
4FCA 7C       10400         LD     A,H
4FCB FE3F     10500         CP     3FH
4FCD C2D64F   10600         JP     NZ,SKIP ;JUMP ON H<>3FH
4FD0 06CD     10700         LD     B,0CDH
4FD2 10FE     10800         DJNZ   $       ;DELAY 3.3 MSEC ON HL=3FFCH
4FD4 10FE     10900         DJNZ   $
4FD6 2A2040   11000 SKIP    LD     HL,(4020H) ;SAVE CURSOR ADDRESS
4FD9 E5       11100         PUSH   HL
4FDA 08       11200         EX     AF,AF'
4FDB CB47     11300         BIT    0,A     ;TEST BIT 0
4FDD CAE44F   11400         JP     Z,LBL120 ;JP ON BIT 0 LOW
4FE0 06CD     11500         LD     B,0CDH
4FE2 10FE     11600         DJNZ   $       ;DELAY 1.5 MSEC
4FE4 CDE84F   11700 LBL120  CALL   HEXDSP  ;PUSH PC
0001          11800         DEFS   1       ;RESERVE ONE BYTE
4FE8 E1       11900 HEXDSP  POP    HL      ;POINTS TO RESERVED BYTE
4FE9 77       12000         LD     (HL),A  ;DISPLAY BYTE IS AT RESERVED ADDRESS
4FEA 0602     12100         LD     B,2
4FEC AF       12200 HXDP1   XOR    A
4FED ED6F     12300         RLD            ;UNPACK ONE NIBBLE
4FEF C630     12400         ADD    A,30H   ;HEX TO ASCII
4FF1 FE3A     12500         CP     3AH
4FF3 FAF84F   12600         JP     M,HXDP2 ;JP ON DIGIT BETWEEN 0 & 9
4FF6 C607     12700         ADD    A,7     ;ADD ON DIGIT BETWEEN A & F
4FF8 CD3300   12800 HXDP2   CALL   33H     ;DISPLAY AT CURSOR
4FFB 10EF     12900         DJNZ   HXDP1
4FFD C3BF4F   13000         JP     LBL118  ;READ NEXT BYTE
4DEA          13100         END    TTAPE
00000 TOTAL ERRORS
HXDP2   4FF8
HXDP1   4FEC
HEXDSP  4FE8
LBL120  4FE4
SKIP    4FD6
LBL118  4FBF
LBL117  4FA9
LBL116  4F9F
LBL115  4F9C
LBL114  4F94
WTTAPE  4F68
RTTAPE  4FB1
LBL113  4F5B
LETTER  4F19
TTAPE   4DEA
CLRSCN  4DDC
SPACE   4DD4
DSPLY   4DC9
```

TTAPE
READS AND WRITES A TEST TAPE
BY DENNIS STEVENS

TO WRITE THE TEST DATA ONTO TAPE FIRST MAKE THE RECORDER READY, THEN PRESS W

TO READ THE TEST TAPE, FIRST MAKE THE RECORDER READY, THEN PRESS R

TO RE-ENTER TTAPE FROM BASIC II, ENTER SYSTEM AND THEN ENTER /19946 AFTER THE SYMBOLS*?

Example 1.

tape head.

The TTAPE program can be put onto cassette tape by using T-BUG or the Editor-Assembler, or the BASIC program in Listing 2 can be saved by the CSAVE command.

Expanded TTAPE

An expanded version of TTAPE on cassette tape is available from the author for $4.95. This version allows reading or writing test data by keyboard command, and operation of the program is explained by visual prompting. The program resides at the end of memory for 4K Level II machines.

The expanded version accepts the keyboard commands W to write test data, or R to read and display test data. This version is easy to operate: Just enter SYSTEM, then after making the recorder ready (in play mode), enter TTAPE, which will load the program into your TRS-80. Then enter / to execute the program. This will cause the instructions in Example 1 to appear on the screen.

You will probably begin by writing test data onto tape. This is done as instructed by making the recorder ready (record mode) and pressing W. The statement "WRITING THE TEST DATA" will appear on the screen as the recorder starts and hex AAs are recorded on tape. This mode continues indefinitely until the reset button is pushed. This allows writing a test tape of any length. After reset, the machine will be in BASIC.

To read the test data, return to the monitor mode by entering SYSTEM, then enter / (or a / 19946 if another machine-code tape has been entered). Rewind the tape to the beginning of the test data, put the recorder in the play mode and press R on the keyboard. The recorder will start, and the hex AAs will be read from tape and displayed on the monitor screen. This read mode is also indefinite in length and is terminated by pushing the reset button. While reading and displaying test data, the cassette recorder can be adjusted, and the effects of the adjustment can be observed on the screen. A listing of this version is shown in Listing 3.■

PROBLEMS?

Are you having trouble with advertisers or products?
If such is the case, please write to the firm
giving complete details and send a copy of your letter to
Wayne Green, *80 Microcomputing*,
Peterborough, New Hampshire 03458.
Be polite—no matter how badly you're shafted.

We're looking for names and addresses of all the **TRS-80 newsletters** out there. If you produce or are affiliated with such a newsletter, please drop a line to:

Jim Perry, Managing Editor
80 MICROCOMPUTING
Peterborough NH 03458

Thanks.

✓ Reader Service—see page 147

What's the big deal about disk drives? Find out how they work and what they can do.

A Disk Primer

William O'Brien
11 Dongan Place
New York NY 10040

If it hasn't happened yet, rest assured that it will. There you are, sitting smugly in front of your TRS-80, having just zapped the last Klingon battleship, your screen is flashing a well deserved 'Congratulations, Captain. You have saved the Federation.' Looking at those words, suddenly a thought will flash across your mind: 'Is this what I spent a thousand dollars on?'

Yes it is; but sit back and relax. You haven't wasted all that money, you've just stopped a little short of what you wanted.

Yes, I know that those ads with the kid in front of the Christmas tree doing evaluations of his stock portfolio on a TRS-80 made you a little unsure of exactly what you were getting into, but then Radio Shack wasn't too familiar with what their product could really do. For years all their ad men had to do was promote Archer Kits, CB's and stereo components. A computer was another novelty and they had to sell it.

A Radical Decision

So now you have a 4K Level I, or perhaps a 16K Level II, if you outsmarted them, and you're fed up with games and cassette recorders in general. What do you do?

Let me become very radical and tell you straight out: Either buy yourself a practical arcade machine that will give you reasonable graphics and a wider selection of games or else start saving your dollars and finish up what you've already started.

I'm talking about that most dreaded monster, the Disk Drive, that costly peripheral that also adds the expense of an expansion interface to your system.

Firstly, I warn you, the decision to upgrade will mean abandoning a lot of those games, since Level II tapes are not compatible with the disk operating system.

Secondly, you must realize that a 16K machine will only be sufficient for a small disk system. If you want to realize the full power of a disk operating system, it requires a minimum of 32K of user available memory. Since you TRS-80 owners must have the Radio Shack DOS disk present in Drive 0, there is not much room left for storage (for instance, RS's 2.2 DOS leaves only 18 kilobytes of storage of a possible 89.6 kilobytes). That means two disk drives.

And friend, that's where we separate the dedicated computerist from the arcade game freak. Now that those with little intestinal fortitude and even less cash have gone on to the latest version of battleships, I'll start at the beginning.

Dumb Programming

Let me preface by saying that I am a Dumb Programmer; much like a Dumb Terminal, I can perform after some mental mulling and more often than not get a concept correctly translated into BASIC.

And, if you're anything like me the first question you might ask is, what is a disk?

As the name implies, it's a circular object made of the same material as recording tape. But unlike tape, a disk has no real beginning or end and allows access to all of itself at once.

Using Radio Shack's statistics, that means that the length of tape holding a 13,000 byte program will take about three and a half minutes to load. The same program, loaded from the disk, takes only about twenty seconds. Now, I realize, this will mean no more quick trips to the refrigerator while your asterisks are flashing, but then, according to the Department of Health, most Americans are overweight anyway.

Generally, disks come in two sizes, an eight-inch version used by the TRS-80 Model II and a five and a quarter-inch mini disk or diskette used in the Model I.

As supplied, the TRS-80 Model I uses four mini disks and a drive storage of approximately 300,000 bytes. The four eight-inch disk drives on the Model II can store almost 2,000,000 bytes.

Now how, you ask, can a round piece of plastic hold so much information? It's not easy, but it's not really that difficult. Look at Fig. 1.

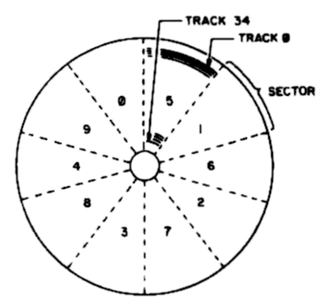


Fig. 1. Diskette

Thirty-five tracks on a diskette are divided into ten sectors, each sector can hold 256 bytes of data. (Under DOS 2.1 you were limited to 255 but 2.2 changed that.) These 350 sectors at 256 bytes per sector, can store up to 89.6K bytes of information.

Unfortunately, under the easiest method of file storage—sequential access—DOS is not memory efficient. Physically, the diskette holds 70 files, each five sectors long. When you write something to the diskette the number of files is reduced by one. Whether or not the data is long enough to fill the file, all five sectors (called granules) are used.

No partial granules are available so that one file three and a half sectors long (roughly 4.5K) uses five complete granules (a little under 7.5K of space).

To compensate for this, another type of data storage is available called Random Access. This allows the programmer to use all his sectors by defining the length of the data to be stored.

Learning to Drive

What is a disk drive?

In the TRS-80 Model I, Radio Shack uses a Shugart SA-400. When you receive it from them, if you are dumb enough to take it apart, you'll probably see something that looks like Fig. 2.

The most important features are the drive motor, the stepper motor and the read/write head.

The index LED and detector keep the stepper motor from going too far in either direction, while the protect switch acts like an electronic version of the tabs on the back of cassettes, only in reverse. When there is no tab on the disk, you can write; when you cover the notch the switch would fall into, you cannot.

The stepper motor guides the head across the face of the diskette, reading the electrical impulses. The magnetic head is similar to the heads in a tape recorder, only it performs both read and write functions. The drive motor makes sure the diskette turns at the proper speed.

The power supply, write/read electronics, and the controller logic are also important—if they don't work, the drive won't work. However, if the head is misaligned, the stepper motor does too fancy a two step, or the drive motor speed is off, you'll get anything from garbage reads to intermittent errors or total dysfunction. Thankfully, this doesn't happen very often, but those are the pitfalls.

You now have your 16K system and a disk drive. Did you notice when you turned on the computer, the disk operating system didn't automatically load? All you see is MEMORY SIZE and Level II **BASIC.**

Now you know why they sell expansion interfaces.

Somehow, the computer has got to be able to control the actions of the disk drive. It must tell the drive when to read, when to write and when to load the operating system. That's the purpose of the expansion interface.

The E/I houses the disk controller that, like a ring master at a circus, runs the whole show. It gets the orders from the CPU, determines if it is a valid command and tells the disks what to do. It holds up to 32K more of memory, has a clock (which under DOS advances day and date) and lets you hook up a second cassette—heaven forbid.

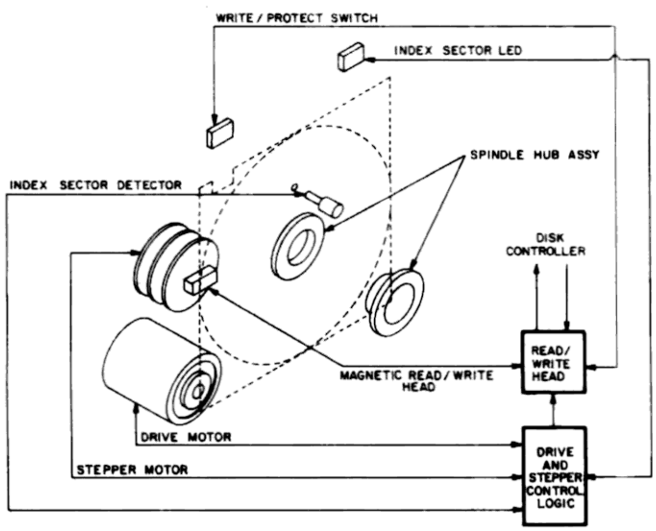


Fig. 2. Shugart SA-400 Disk Drive

The Disk Operating System

Aside from robbing you of about 10K of memory and about ½K of dollars, if you are just a keyboard puncher with a few good ideas like me, most of the power of DOS will be lost on you for the time being.

What is important to know is that the DOS and Disk BASIC systems let you implant and retrieve more information at a faster rate of time than possible with a cassette. The DOS is the software patch that allows you to utilize (and remember, there is a small but important distinction between *utilize* and *use*) the disk drive.

Radio Shack lists a long line of commands contained in DOS and Disk BASIC, ranging from APPEND and USRn, but if you are just beginning to use your drives, there are only a few you'll really need.

DIRectory: Lists all stored programs on the drive you select.

TEST1: Tests all RAM and ROM memory locations, just in case it's not a power line spike that crashed your system.

TEST2: This test puts your disk drive through the ringer and if something's wrong, it will tell you.

BASIC: Just that, it loads Disk BASIC.

BASICR: This is the same as above but with one important feature, it lets you run a BASIC that has built-in renumbering abilities.

BASIC• and BASICR•: These allow you to get back to the version of BASIC you were in before the system crashed or locked up and still have the program you were using in memory. Unfortunately you can't write a program under BASIC, exit to the DOS and return to BASICR. BASICR is just a bit longer and does wonders to your first two lines if you try it.

CMD"S": Lets you take a look at the DOS for whatever reason without having to reset the system or power up again. It's great if you can't remember program names. You can return to the version of BASIC you were in by using the BASIC• or BASICR• commands.

BACKUP: Probably the most 'thanked' utility. It will let you make a complete copy of a diskette, just in case you leave the original too close to the left side of the monitor or near that great new speaker system with the two pound magnets you just bought. With one drive it's boring because you must remove the source disk, insert the destination disk, remove the destination disk, reinsert the source disk, etc.

SAVE: Not too different from CSAVE except that it writes your program to disk.

RUN: Loads and runs a called program.

KILL: As the name implies, it kills or deletes the named program or data file.

OPEN: This opens for use any of the fifteen buffers or holding areas in the TRS-80 (each 256 bytes long) that can be used to hold data before writing it to disk.

WRITE: This will write the buffer, whose number you select, onto the disk.

INPUT: This reads the selected information from the disk into the buffer you have selected. The three commands above, OPEN, WRITE and INPUT must be used in a set format to let the computer know if it is input or output to or from the disk and into which buffer number it is to be read or out of which buffer it is to be written. These are the techniques for sequential file access.

FREE: Used under DOS to display the amount of free storage space available. The answer is in units called GRANS which equal about 1280 bytes.

BASIC2: After you've been frightened out of your wits by your first shot at DOS and Disk BASIC, this command will let you slip quietly back to Level II BASIC and full memory capabilities. It's nice if you just want to go back and play Star Trek too.

Weigh Your Needs

Just to recap the most important things, remember, the monetary investment is sizable. You will, more likely than not, become disenchanted with the limitations of a single drive and 16K of memory. Yes, I know the Radio Shack advertisements tell you there is 65K of storage on the DOS drive, but DOS 2.2 leaves only 15K and version 2.3 will probably leave less (when I requested my copy of 2.3, the salesman asked how many drives I had).

I also recall that the salesman said that 16K of memory was enough, but after TRSDOS and Disk BASIC get through taking pieces, there'll only be about 6K left.

To upgrade a Level II system with 16K of memory will require an expansion interface with 16K of memory ($448) and two drives ($499 each), for a grand total of $1446.

When I was twenty years old that was a little less than half the price of a brand new Chevy Belair four door sedan with power steering and power brakes.

Personally I can justify it in terms of using the expanded and enhanced data file storage techniques for my business, but I never entered into the expense with a distant hope of 32K Disk Star Trek in mind.

And if, even after all of this, you are still going to order the E/I and Disk(s), then get going. You'll have about five months with nothing to do while your order is being filled.

My advice is to buy a copy of the DOS manual and start looking over the various commands and utilities the system offers. Sit down and write out some theoretical programs that you think might work in Disk BASIC, programs that utilize file placement and retrieval.

Best of all, go into your nearest Radio Shack store or authorized outlet and ask the manager if you can try writing some programs in Disk BASIC. Tell him you are thinking of buying a 16K E/I and two disks. Ask him to let you get a feeling for a disk based computer. Buy a blank disk from him, if he doesn't offer one. Don't let him sit you down with a disk full of games. If you have any questions after using the system, ask him. If he can't answer you, call (800) 433-1679, that's the Radio Shack Computer Services' toll free number. Ask them about their instructional classes.

Keep in mind that with a disk system your TRS-80 will not leap tall data files in a single bound. But what you are going to see, perhaps for the first time since you purchased the system, is what a real computer is capable of doing. ■

HARDWARE

Quick, dirty but cheap solution to most CLOAD problems.

Cheap CLOAD Fix

Scott D. King
7905 59th Ave. N.
New Hope, MN 55428

How many times have you had to reload your TRS-80 tape programs all because you had the volume control set in the wrong place? Or because your tape head was getting dirty, and wasn't giving out that same signal each time.

Well, if your machine is anything like mine, the answer is TOO OFTEN!

You can live with the problem until you start working with programs that take five or six minutes to load.

Here is a quick and dirty (but cheap) fix for this problem, shown to me by an engineer friend named Dick Wood. All that's required is a 100 pf capacitor and two germanium diodes.

Place the diodes back to back, (cathode to anode), across the input of the cassette signal to the op amps (see fig. 1). Place the 100 pf capacitor across R 42, a 1 megohm feedback resistor on Z4 (see Fig. 2).

These diodes limit the input signal, allowing you to crank the volume control on your tape recorder all the way up without overdriving the circuitry.

The capacitor works as a high frequency filter blocking unwanted noises that could cause data errors. The volume will work anywhere between 1/2 and full. This should solve most tape problems. ■

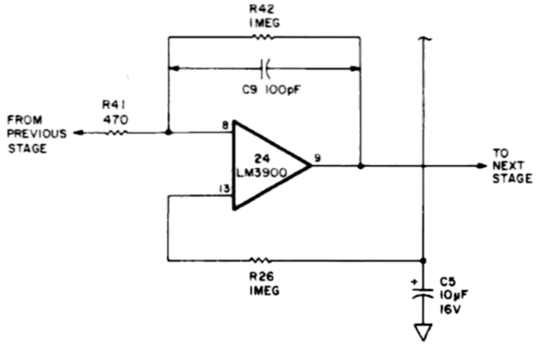


Fig. 1.

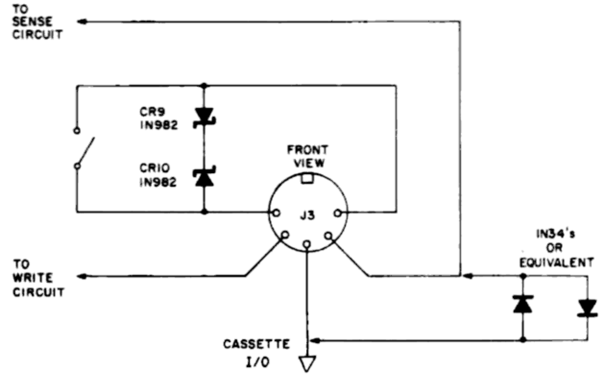


Fig. 2.

ITS PROTECTIVE COVERS . . . . .

A COMPLEMENT TO ANY DECOR

- *Attractive, durable, leather grained vinyl*
- *Custom designed fit*
- *Double stitching and corded seams*
- *Choice of colors (Gold, Chestnut Brown, Olive Green or Black)*

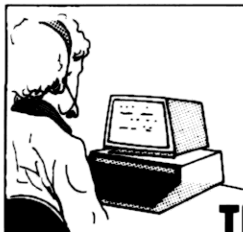
PRICES TRS – 80 COVERS

	Cover	Shipping
1. Keyboard	9.95	.75
2. Monitor	14.95	1.00
3. Monitor (with Interface)	15.95	1.00
4. Cassette	5.95	.75
TRS–80 COVER SETS A THROUGH D (See Descriptions Above)		
A. 1 & 2	20.95	1.25
B. 1 & 3	21.50	1.25
C. 1, 2, 4	24.50	1.35
D. 1, 3, 4	24.95	1.35
Apple - II	12.95	1.00
Apple Disk - II	8.95	.75
PET 2001 and PET-CBM	18.95	1.25

VIRGINIA RESIDENTS
Add 4% Sales Tax
METHODS OF PAYMENT
Cash/Check, C.O.D.
Visa or Master Charge

"TRS-80 is a registered trademark of TANDY CORP."

International Technical Systems, Inc.
P. O. Box 264, Woodbridge, Va. 22194
Telephone (804) 262-9709 ✓86



Talk to your TRS-80

Now, Scott VET/I, an exciting new product of Scott Instruments, permits you to talk to your TRS-80, allowing voice control of basic programs or games. The Scott VET/I offers:

- User initialized vocabulary of up to 40 words
- Fast recognition
- High accuracy
- Demonstrated performance comparable to systems costing $10,000 or more at hobbyist prices.
- Multiple user Capability with no increase in storage requirements
- Easy to train and use (user programs can be written in Basic)
- Requires TRS-80 (c) Model I with expansion interface, 16 K of additional memory and dual disk drives/ cassette (c) Trademark of Tandy Corporation

For more information call
(817) 387-1054 or write

Scott Instruments ✓91
815 North Elm, Denton, Tx. 76201

For Your TRS-80*

DUAL DISK-32K
BUSINESS SYSTEM

QUALITY DTI PAYROLL

No. 410

DATA TRAIN'S many years of small business computer experience in accounting program products brings to your business all of the quality features, functions, screen displays, standard reports, user designed reports and operator reference manuals; allowing you to efficiently manage the payroll of your company.

- 50 employees per mini-disk.
- Runs in all states.
- You maintain the P/R product without programing.
- Flexible, easy to use.
- No maintenence fees.

$235.

Product Info & License/Order Form.

FROM.....

DATA TRAIN INC.

PHONE
(503) 476-1467 ✓44
840 N.W. 6th STREET, SUITE 3
GRANTS PASS, OREGON 97526

Available Soon-

- General Ledger Accounting
- Accounts Payable
- Accounts Receivable
- Fixed Asset Accounting

**Trademark Radio Shack, Div Tandy Corp.*

TRS-80* STRUCTURED BASIC

Now available for TRS-80 disk systems

SBASIC

A pre-compiler adding performed procedures, case structures, repeat while, until and many other structured language statements to the already powerful Level II BASIC. Diskette $50.00, Listing $35.00, Manual purchased separately $10.00.

ULTIMATE COMPUTER SYSTEMS

313 Meadow Lane
Hastings, Michigan 49058
(616) 945-5334
(Dealer inquiries invited)

✓84

**TRS-80 is a trademark of Tandy Corp.*

AT LAST

USEFUL HOME PROGRAMS For Your Level II 16K

CALCULATOR PLUS turns your TRS-80 into a printing calculator! Excellent as an on-screen calculator without a printer. All functions double precision, chain and mixed calculations, 5 memories with titles—can be used as constants, item count in add and subtract with on-screen or printed review of entries, optional dollar format. Prints all calculations, results and memories, ideal for use with Quick Printers. Loads fast, easy to use.
MPC-1 $9.95

CHECKBOOK PLUS solves the problem of monthly bank-statement reconciliation. No cumbersome tape record keeping. Just do your checkbook once a month, and let Checkbook Plus handle all the details and find the errors.
MCB-1 $9.95

CALCULATOR PLUS and **CHECKBOOK PLUS** both on one cassette.
MC-2 only $14.95

Each cassette includes a bonus Loan Payment program with clean dollar-format on-screen monthly or yearly display of payment, interest, capital and balance.

"TRS-80 is a registered trademark of TANDY CORP."

MANHATTAN SOFTWARE, Inc.
P.O. Box 5200 Grand Central Station
New York City, New York 10017

✓90

How to get dozens of extra variables in Level I.

Extra Variables

Gary E. Clark
7505 Democracy Blvd., A-123
Bethesda MD 20034

In the TRS-80 Level I BASIC, the number of variable names is severely limited. There are 26 single-value variables, named A through Z, and the array A(I). (String variables are an even bigger problem, but that is not the subject of this article.)

After you have used up the 26 letter variables, which don't require a large program, you can use A(0), A(1), etc., just as though they were single-value variables with names four bytes long (or longer if you go to A(10) and higher). However, most frequently the low-order array is already being used. Anyway, it is cumbersome to be forced to use four or more bytes for the name of a variable when you can't choose letters that indicate a meaningful word.

The Plan

We will use the variables from the last half of the alphabet, N through Z, in the normal manner. We will use the variable A as an index and set up multiple sets of the 12 variables B through M. That is, when A = 0, we will be using the zero-order set of B through M variables; when A = 1, we will be using the first-order set, another completely independent set of B through M variables; and so forth up to the limit of space available in the memory for storing the values for all sets of variables.

When we wish to switch from the current set of B through M variables to another set, we will call an exchange subroutine to automatically store the current values and switch to the desired set based on the value that we have given A. It is helpful to think of the zero-order sets as B0, C0, D0, etc., the first-order set as B1, C1, D1, etc., and so on.

It is obvious that we must store these sets of variables somewhere, and the only available place in Level I BASIC is the A(I) array. You will recall that we set out to reserve the low-valued subscripts for program arrays, so the choice is to go to high-value subscripts. The highest available subscript value is MEM/4-1, which we will reserve for the current index number, 0 for the zero-order set of B through M variables, 1 for the next set, and so on. This will free the variable A to be used as we wish until we actually exchange variable sets.

Our subroutine will require the use of one working variable. We can use A(MEM/4-2). The different sets of B through M variables will be stored in the high end of the A(I) array at successively lower values of the subscript.

The Exchange Subroutine

Line 15010 of the exchange subroutine temporarily stores the new index variable in A(MEM/4-2). This frees A to be used as a working variable. In line 15020, the value of the array subscript that corresponds to the location for the B variable of the current set is calculated and placed in the working variable A.

In the next 12 lines, the current values of B through M are stored in the appropriate locations. In line 15160, the value of A(MEM/4-1) is updated to the new index number, which was temporarily stored in A(MEM/4-2).

The type of calculation that was done in line 15020 is again performed in line 15170, but this time with the new index number. In the next 12 lines, the old values of the B through M variables are replaced with the values from the new set. The value of A is restored before leaving the subroutine.

In the listing, each statement is placed on a separate line to make it easy to read. When you use the exchange subroutine in a program, you should, of course, place several statements, separated by colons, on each line. Also abbreviate MEM and RETURN. The reduced subroutine uses less than 340 bytes.

At the start of your program, you must initialize the index value stored in A(MEM/4-1). This is shown in line 10. The number of high-order array elements used is two plus 12 times the number of sets of variables. Therefore, you must be certain that the maximum array subscript used for other purposes is no greater than MEM/4-3-12*N, where N is the number of sets of variables, the maximum value of A plus one.

Applications

A practical use of an alternate variable set is for operator constants. In Level I, we can allow keyboard inputs of individual alphabetic characters that, as mnemonic devices, are the first letters of meaningful words corresponding to the desired operations.

The choice of letter variables *must first be set equal to a set* of numbers that will be used in an ON statement, for example,

```
A$ = ","
PRINT#A;A$;B;A$;C;A$;D;A$;E;A$;F;A$;G;A$;H;A$;I;A$;J;A$;K;A$;L;A$;M
```

Example 1.

to direct program control to tl. corresponding operations. Store these operator constants in an alternate variable set, the A = 1 set, for example. Then at the INPUT statement, use the following:

```
A=1  :  GOSUB 15000
INPUT X
A=0  :  GOSUB 15000
```

This will preserve and restore the main set of B through M variables with the least amount of effort.

A dozen extra variables can come in handy when a subroutine needs some working variables. For example, if you have tried to use the exponentiation subroutine in the back of the Level I manual, you have discovered that, in addition to the output variable P and the input variables X and Y, of which X, unfortunately, is altered, the subroutine uses E, L, A, B and C. This is a total of eight variables that are tied up out of the 26 normally available. If your program uses the exponentiation subroutine, it is probably of such a size that every single-letter variable is needed; but you must be ever vigilant that E, L, A, B, C and X aren't currently being used for storing variables that should not be changed.

The perfect subroutine returns the value of the output variable without altering the input variables and without altering any other available variables. This can be done by using the exchange subroutine. As you enter the exponentiation subroutine, or any subroutine that needs working variables, execute an exchange with A = 2 : GOSUB 15000, for example. That gives a dozen working variables. At the end of the subroutine, exchange back to the previously stored main set of B through M variables with A = 0 : GOSUB 15000.

True, it takes time to do these exchanges (about 0.4 seconds for the subroutine), but we can make up for this by also storing constants in the alternate variable set. In the example of the exponentiation subroutine, we need only E, L, A, B and C, so that leaves room for seven constants. Scanning through the log and exponential subroutines, which are used by the exponentiation subroutine, we find several constants: .598979, .961471, 2.88539, .693147, 1.32988E-3, 1.2131E-4 and .707107.

At the beginning of our main program, we can initialize some of the variables in the alternate set to these constant values and replace the numbers in the subroutines with the corresponding single-letter variables. Operations on single-letter variables are much faster than conversions of decimal numbers to internal form each time they are needed.

When alternate variable sets are used in a subroutine, the input and output variables must be included among the N through Z variables, which are not exchanged. Otherwise, the calculations in the subroutine would be done with wrong values in the variables. This shows the advantage of having a permanent set of N through Z variables that can interact with many sets of B through M variables. If the exchange subroutine were to exchange all 26 variables, this interaction would not be possible.

Here's another example where an alternate set of variables is useful: If you use the PRINT# statement to store variables on tape, you should know that at the beginning of each PRINT# operation a leader of over 130 bytes is put on the tape before the first variable is recorded. That leader uses up about four seconds per tape record on both output and input.

I have found that the most common variables I need to

✓ Reader Service—see page 147

record are arrays. Unfortunately, when PRINT# A(I) is placed in a FOR-NEXT loop, each value is treated as a separate record with a four-second leader. The solution is to switch to an alternate variable set with the exchange subroutine. That gives 13 free, single-letter, working variables. (Remember: A is free to use once the exchange is made.)

With a small subroutine put 13 elements of the array into A through M. Then do a tape output with Example 1. This is treated as one record with only one four-second leader for the entire set. That saves 48 seconds for the tape output and, correspondingly, another 48 seconds for a tape input.

For large arrays, the time saved is enormous. It takes about one hour and 15 minutes to tape an array of 1000 elements with one element per tape record; but when each record contains 13 elements, the total time is only about seven minutes.

To reverse the procedure for tape input, exchange variable sets and execute INPUT# A,B,C,D, ,F,G,H,I,J,K,L,M. Place these values in the array and proceed to the next record to get all the data. On conclusion of the tape reads, exchange back to the main set of B through M variables. Note that the procedures just described have speeded up tape operations without disturbing the main set of B through M variables.

Conclusion

A word of caution: The memory locations occupied by the different variable sets depend on the program size as inversely reflected in MEM. If a program statement is rewritten with a different number of bytes while debugging the program, the alignment of variable sets is destroyed, and you must begin at the start of the program to initialize the alternate variable sets.

Let's face it, Level I BASIC is inadequate in many ways! However, with a little effort, it can become a useful system. I devised the technique of alternate variable sets out of necessity. Solving such problems is what makes programming so much fun.■

```
   1Ø  A(MEM/4-1) = Ø : REM INITIALIZATION
15ØØØ  REM B THRU M ALTERNATE VARIABLE SETS
15Ø1Ø  A(MEM/4-2) = A
15Ø2Ø  A = MEM/4-3-12*A(MEM/4-1)
15Ø3Ø  A(A) = B
15Ø4Ø  A(A-1) = C
15Ø5Ø  A(A-2) = D
15Ø6Ø  A(A-3) = E
15Ø7Ø  A(A-4) = F
15Ø8Ø  A(A-5) = G
15Ø9Ø  A(A-6) = H
151ØØ  A(A-7) = I
1511Ø  A(A-8) = J
1512Ø  A(A-9) = K
1513Ø  A(A-1Ø) = L
1514Ø  A(A-11) = M
1516Ø  A(MEM/4-1) = A(MEM/4-2)
1517Ø  A = MEM/4-3-12*A(MEM/4-1)
1518Ø  B = A(A)
1519Ø  C = A(A-1)
152ØØ  D = A(A-2)
1521Ø  E = A(A-3)
1522Ø  F = A(A-4)
1523Ø  G = A(A-5)
1524Ø  H = A(A-6)
1525Ø  I = A(A-7)
1526Ø  J = A(A-8)
1527Ø  K = A(A-9)
1528Ø  L = A(A-1Ø)
1529Ø  M = A(A-11)
153ØØ  A = A(MEM/4-1)
1531Ø  RETURN
```

The exchange subroutine.

Frustrated by the limitations of INKEY$? Find out how to use it more effectively.

Keyboard Information

Daniel Lovy
2398 Hulett Road
Okemos MI 48864

Most real time arcade style programs written for the TRS-80 use the INKEY$ function of Level II. This allows a single character to be input without stopping the program. Unfortunately the INKEY$ function cannot sense if a key is being held down, only if it has been pressed.

For example, if you define the "F" key as the key that will cause your animated "whatever" to move forward, it must move one unit at a time, or another key must be designated to stop it. You cannot move "whatever" as long as the key is pressed, then stop it when the key is released, not using INKEY$ anyway.

In the memory map in the back of the Level II manual, locations 14336 to 15359 are listed as "TRS-80 KEYBOARD MEMORY". To better understand what goes on in this hinterland, enter this short program.

```
5 CLS
10 FOR Z= + 14336 TO 14650
20 PRINT PEEK(Z);
30 NEXT Z
35 PRINT CHR$(28)
40 GOTO 10
```

The program will display a section of the keyboard memory. While the program is running, press the "A" key and hold it for a few seconds, then let go. The zeros should have changed to alternating 2's during the time that you had the "A" pressed. Experiment with different keys for awhile.

As you probably guessed a pattern emerges. Each group of keys effect certain memory locations in this section of RAM. The keys are grouped together as in Example 1.

The first key in each group puts a 1 into that group's particular set of locations. The second key in each group places a 2, the third places a 4, the fourth an 8, the fifth a 16 and so on.

Group	1	@ A B C D E F G
	2	H I J K L M N O
	3	P Q R S T U V W
	4	X Y Z
	5	0 1 2 3 4 5 6 7
	6	8 9 : ; , - . /
	7	ENTER CLEAR BREAK ↑ ↓ ← → SPACE

Example 1.

Overlapping Locations

When the locations overlap or when two keys are being pressed at the same time their values are added together. Most of the memory locations do overlap, meaning they are used by more than one group. Here is a list of locations used by only one group.

Group	1	14336 + 1
	2	+ 2
	3	+ 4
	4	+ 8
	5	+ 16
	6	+ 32
	7	+ 64

Let's write a short program that will display an "F" on the screen only as long as the "F" key is depressed (this could be equated with firing an engine or moving an animated "whatever" forward).

First determine its group (1), then its position in that group (7) and finally the memory location used by its group (14337). Since F is in the seventh position it puts a 64 into memory location 14337. Now, let's test for it. The program might look like Example 2.

```
1 CLS
10 A=PEEK(14337)
20 IF A=64 THEN PRINT@0,"F" ELSE PRINT@0," "
30 GOTO 10
```

Example 2.

This information could have been documented by Radio Shack, but, if they had, all the fun would have been taken out of the hunt. Everyone needs to shout "Eureka" once in awhile. ■

Disks and Printers for the TRS-80 . . .

Add 1 or 2 megabytes of disk storage to your TRS-80 with a Micromation MEGABOX® interface — double-density 8" drives

(2 megabytes by utilizing dual-headed double-sided drives).

The Micromation MEGABOX® provides for up to 4 megabytes of storage; it has provision for adding 32K of RAM to your TRS-80, giving you the opportunity of having a 48K system without purchasing an expansion interface. The Micromation MEGABOX comes with an RS232 serial interface, which can be used by a serial printer or a modem.

1-Megabyte MBII-80 MEGABOX for the TRS-80, List $2295 **OUR PRICE $1949**

2-Megabyte MBIV-80 MEGABOX, List $3095 . . . . **$2629**

CP/M Operating System for above . . . $ 150

Microsoft Patch to ROM BASIC . . . . . $ 150

PRINTERS

NEC SPINWRITER for TRS-80, 55 cps, letter quality

5530 RO/Centronics Interf., $3275	2475
ANADEX80-col dot matrix, $995 . . .	$895
INTEGRAL DATA IP-125 w/1210 op. .	$754
IP-225 w/1210 & 1250 op., $984 . .	$889
IP-225 w/1210,1250,1241 (graphics) options, List $1098	$949
IP-440 Paper Tiger, List $995	$895
w/graphics op. incl. buffer $1194	$1069
CENTRONICS779-1, List $1245 . . .	$ 949
779-2 w/tractor, List $1350	1049
780-2 w/tractor, List $1725	1295
701-2 w/tractor, List $2205	1895
702-2 w/tractor VFU, List $2480 . .	1995
703-2 w/tractor VFU, List $2975 . .	2395
704-2 w/tractor, List $2350	1995
730-1 parallel interface, List $995 . . .	895
730-3 RS232 interface, List $1045 . .	939
753-1, List $3095	2785
T.I. 810 basic unit, List $1895	$1695
810 serial/parallel, List $1940 . . .	$1735
DATA PRODUCTS M200-017, $3465	$3119

DISK DRIVES

Fully compatible with expansion interface and TRSDOS software – VISTA, PERCOM, LOBO, and others

SAVE OVER $100

Your choice **$369** ea.

CORVUS 10 megabyte hard disk — ONLY $4795

CONVERT YOUR EXISTING SELECTRIC TYPEWRITER TO A LETTER-QUALITY PRINTER . . .

Power Supply & electronics, assembled & tested. You make only a simple solenoid installation (or have the factory do it). Manufactured by Escon.

Parallel version, List $525	$469
TRS-80 cable	$ 20

RADIO SHACK TRS-80's™ EXPANSION INTERFACES AND DISK DRIVES at 10% DISCOUNT

Full warranty at any Radio Shack store

26-1051 4K RAM, Level-I BASIC, List $499	$449
26-1053 16K RAM, Level-I BASIC with numeric keypad, List $729	656
26-1054 4K RAM, Level-II BASIC, List $619	557
26-1056 16K RAM, Level-II BASIC with numeric keypad, List $849	764
26-1140 Expansion Interface with 0 Ram, List $299	269
26-1141 Expansion Interface with 16K RAM, List $448	403
26-1142 Expansion Interface with 32K RAM, List 597	537
26-1160 First Disk Drive (includes DOS, cable), List $499	449
26-1180 Voice Synthesizer, List $399	359

SHIPPING, HANDLING AND INSURANCE: Add $5 for Selectric Converter, $6 for disk drives, $10 for Megabox. Centronics printers shipped freight collect. Contact us for shipping information on other terminals and printers.

All prices subject to change and all offers subject to withdrawal without notice. Prices in this ad are for prepaid orders. Slightly higher prices prevail for other-than-prepaid orders, i.e., C.O.D., credit card, etc.

— WRITE FOR FREE CATALOG —

MiniMicroMart, Inc.

24

1618 James Street, Syracuse, NY 13203 (315) 422-4467 TWX 710-541-0431

✓ Reader Service—see page 147

COMPUCRUISE

Put a computer in your car, which gives you the most effective and functional cruise control ever designed, plus complete trip computing, fuel management systems, and a remarkable accurate quartz crystal time system.

So simple a child can operate, the new CompuCruise combines latest computer technology with state-of-the-art reliability in a package which will not likely be available on new cars for years to come • Cruise Control • Time, E.T., Lap Timer, Alarm • Time, Distance, Fuel to Arrival • Time, Distance, Fuel to Empty • Time, Distance and Fuel on Trip • Current or Average MPG, GPH • Fuel Used, Distance since Fillup • Current and Average-Vehicle Speed • Inside, Outside or Coolant Temperature • Battery Voltage • English or Metric Display. $199.95, without cruise control $159.95.

FLOPPY DISK STORAGE BINDER

This black vinyl three-ring binder comes with ten transparent plastic sleeves which accommodate either twenty, five-inch or ten, eight-inch floppy disks. The plastic sleeves may be ordered separately and added as needed. A contents file is included with each sleeve for easy identification and organizing. Binder & 10 holders $14.95 Part No. B800; Extra holders 95¢ each. Part No. 800

GAME PADDLES & SOUND

For Your TRS 80 (Level II or Disk)

Includes: 2 game paddles, interface, software, speaker, power supply, full documentation including: schematics, theory of operation, and user guide; plus 2 games on cassette (Pong and Starship War). $79.95 Complete Part No. 7922C

DISKETTES

Box of 10, 5" $29.95, 8" $39.95.
Plastic box, holds 10 diskettes, 5" - $4.50, 8" - $6.50.

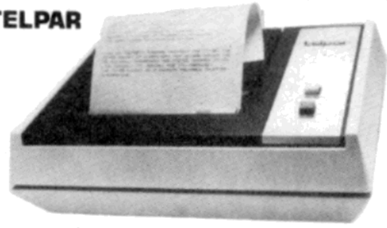
LEEDEX MONITOR

12" Black and White • 12 MHz Bandwidth • Handsome Plastic Case • $139.00

DIGITAL CASSETTE

5 min. each side. Box of 10 $9.95. Part No. C-5.

TELPAR



TTL parallel, TTL serial, RS232C serial, 20 mil loop serial. Quiet, clean — no ribbons or ink; 24 characters/second; Upper/lower case 96 ASCII characters; Signalling rates: Serial Mode — 110 or 300 Baud; Parallel mode — up to 960 characters/second; For applications over 48 columns, automatically prints on next line; Prints 1 character at a time — ideal for interactive applications; Local line feed for advancing paper; Parity check (odd, even or none); ASCII control codes: back space, carriage return, line feed (implemented as a new line), vertical tab (1 line), negative vertical tab (on receipt of ESC, VT), Bell (9 microsecond pulse on receipt of BEL) PS-48C Printer $699, with Keyboard, $895. Paper roll 5½"x164' $3.00

ASCII TO CORRESPONDENCE CODE CONVERTER

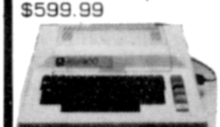
This bidirectional board is a direct replacement for the board inside the Trendata 1000 terminal. The on board connector provides RS-232 serial in and out. Sold only as an assembled and tested unit for $229.95. Part No. TA 1000C

DISK JACKET™

Made from heavy duty .0095 matte plastic with reinforced grommets. The mini-diskette version holds two 5-1/4 inch diskettes and will fit any standard three ring binder. The pockets to the left of the diskette can be used for listing the contents of the disk. Please order only in multitudes of ten. $9.95/10 Pack.

ATARI 800

Computer with 8K $995.00, disk drive $549.00, printer $599.99



TRS-80 SERIAL I/O

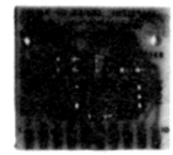
• Can input into basic • Can use LLIST and LPRINT to output, or output continuously • RS-232 compatible • Can be used with or without the expansion bus • On board switch selectable baud rates of 110, 150, 300, 600, 1200, 2400, parity or no parity odd or even, 5 to 8 data bits, and 1 or 2 stop bits. D.T.R. line • Requires +5, -12 VDC • Board only $19.95 Part No. 8010, with parts $59.95 Part No. 8010A, assembled $79.95 Part No. 8010 C. No connectors provided, see below.

EIA/RS-232 connector Part No. DB25P $6.00, with 9', 8 conductor cable $10.95 Part No. DB25P9.

3' ribbon cable with attached connectors to fit TRS-80 and our serial board $19.95 Part No. 3CA840

RS-232/20mA INTERFACE

This board has two passive, opto-isolated circuits. One converts RS-232 to 20mA, the other converts 20mA to RS-232. All connections go to a 10 pin edge connector. Requires +12 and -12 volts. Board only $9.95, part no. 7901, with parts $14.95 Part No. 7901A.



COMPUCOLOR II

Model 3, 8K $13 95, Model 4, 16K $15 95, Model 5, 32K $18 95. Prices include color monitor, computer, and one disk drive.

PET COMPUTER

With 32K & monitor - $1195. Dual Disk Drive - $1195.

APPLE II or APPLE II PLUS

16K - $995, 32K $1059, 48K - $1123. Disk & cont. $589

CASSETTE TAPE ERASER

REMOVES RECORDINGS IN ONE SECOND! The process eliminates static positive / negative ions and maintains original tone quality with minimal tape hiss • To improve tone quality • To reduce hissing • For quick and easy to erase • No battery or liquid required • Powerful and effective action • Unconditional 2 year guarantee. ERASER-8 $19.95.

16K RAMS

For the Apple, TRS-80 or Pet $8 each Part No. 4116/2117.

APPLE II HOBBY/PROTOTYPING CARD

$14.95 Part No. 7907

T.V. INTERFACE

● Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple ● Power required is 12 volts AC C.T., or +5 volts DC ● Board only $7.60 part No. 107, with parts $13.50 Part No. 107A

PARALLEL TRIAC OUTPUT BOARD FOR APPLE II

This board has 8 triacs capable of switching 110 volt 6 amp loads (660 watts per channel) or a total of 5280 watts. Board only $15.00 Part No. 210, with parts $119.95 Part No. 210A.

To Order: Mention part no. description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders in the U.S. only, or a VISA or Master Charge no., expiration date, signature, phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling. Payment must be in U.S. dollars. Dealer inquiries invited. 24 hour order line (408) 448-0800

Send for FREE Catalog . . . a big self-addressed envelope with 41¢ postage gets it fastest!

ELECTRONIC SYSTEMS Dept. 80 P. O. Box 21638, San Jose, CA USA 95151

✓26

80 microcomputing book shelf

•INSTANT BASIC – BK1131 – by Jerald R. Brown. For the personal computer enthusiast or the user of DEC's BASIC PLUS language, here is a new book to teach you BASIC. It teaches BASIC to beginners using interesting programming ideas and applications that will be easily understood by the home computer programmer. BASIC PLUS users know that the two languages are very similar, so this book can be used by them as well. This is an "active participation" workbook, designed to be used with your home computer so you can learn by doing! Ideas are slowly introduced in a nonmathematical context so the beginner can quickly learn good programming techniques. $6.00.*

•YOUR HOME COMPUTER – BK1172 – by James White, is an introduction to the world of personal microcomputing. This book tells you everything you want to know about home computing and gives the computer novice a painless introduction to microcomputer technology and terminology, beginning with what computers are and how they work. This basic book requires no prior knowledge or experience in electronics or computing. It provides information about home computer kits; guidelines for selecting and building your own microcomputer, how to use your home computer and what you can do with it; lists of computer stores, clubs, periodicals; and answers to many more of your questions about microcomputers and the jargon surrounding the personal computing scene today. $6.00.*

•HOW TO BUILD A MICROCOMPUTER – AND REALLY UNDERSTAND IT – BK7325 – 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.*

•MICROCOMPUTER DICTIONARY – BK1034 – This microcomputer dictionary fills the need to become quickly acquainted with the terminology and nomenclature of the revolution in computers. There is also a comprehensive electronics/computer abbreviations and acronyms section. $15.95.*

•MICROCOMPUTER PRIMER – BK1035 – by M. Waite and M. Pardee. Describes basic computer theory, explains numbering systems and introduces the reader to the world of programming. Describes the world of microcomputing in "real world" terminology. $7.95.*

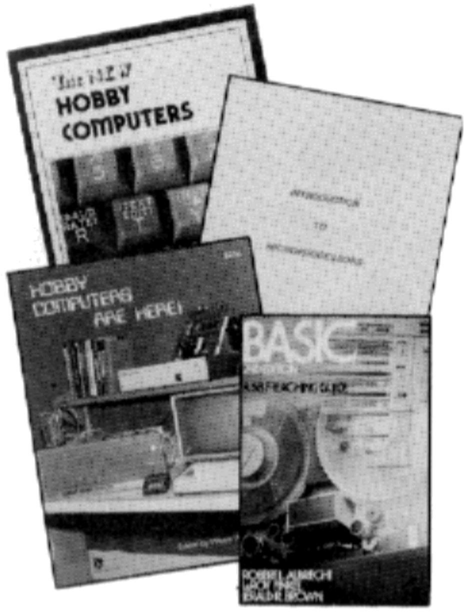
•THE SECRET GUIDE TO COMPUTERS Parts 1, 2 and 3 by Russ Walter. **Part One** describes computers in general, and after reading for ten minutes you will be writing simple BASIC programs! **Part Two** discusses computer applications. It's one thing to master the syntax of the language such as BASIC and another to solve problems using the new tool. **Part Three** describes programming languages. Ever heard of APL and QLISP? BASIC is not the only language used to program computers. 7th Edition. Part I – BK1050 – $2.75*; Part II – BK1051 – $2.50*; Part III – BK1052 – $3.50.*

•SOME COMMON BASIC PROGRAMS – BK1053 – 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. $9.50 paperback.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
80 Microcomputing Book Shelf • Peterborough NH 03458. Be sure to include check or detailed credit card information.
All orders, add $1.00 handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing book shelf



•HOBBY COMPUTERS ARE HERE! – BK7322 – If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

•THE NEW HOBBY COMPUTERS – BK7340 – 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.*

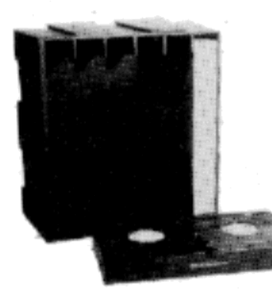
•INTRODUCTION TO MICROPROCESSORS – BK1032 – by Charles Rockwell of MICROLOG, is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Specific systems are not described and programming is only briefly discussed. $17.50 U.S. and Canada; $20 foreign (in U.S. funds).*

•BASIC NEW 2ND EDITION – BK1081 – by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $6.95.*

•HOW TO SELL ANYTHING TO ANYBODY – BK7306 – According to *The Guiness 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. $2.25.*

•PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM – BK1009 – Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $9.95.*

•HOW TO MAKE MONEY WITH COMPUTERS – BK1003 – In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*



•ADD 'N STAC – BX1003 – If you've got programs lying around and want them organized in easy-to-locate fashion Add 'N Stac is the answer. Each unit stores eight cassettes. Each module locks together with the next and grows with your collection. Build yourself a software library with Add 'N Stac by ordering from the Book Nook. As your library grows you'll need more of these handy units so order more than one today. Several colors are available and you can mount them to your wall, desk, table or keep them loose for taking with you. Colors and prices are: Smoke $3.00*; Black, Dark Blue, Orange, Brown, White, Red $2.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: **80 Microcomputing** Book Shelf • Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders, add $1.00 handling.

FOR CUSTOMER SERVICE CALL 603-924-7298

80 microcomputing book shelf

•AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0—BK1130—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. $7.95.*

•VOL. I—BK1030—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. $9.50.*

•VOL. II—BK1031 (loose leaf)—$25.00*; BK1040 (with binder)—$30.00*—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips.

•VOL. III—BK1132 (loose leaf)—$15.00*; BK1133 (with binder)—$20.00.* Contains descriptions of all support devices that can be used with any microprocessor.

•WHAT TO DO AFTER YOU HIT RETURN—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

•BASIC COMPUTER GAMES—BK1074—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. $7.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: **80 Microcomputing** Book Shelf • Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders, add $1.00 handling.

80 microcomputing book order form

Qty.	Cat. #	Description	Unit Price	Total
			Add $1.00 Handling	
			Total	

Enclosed $________ ☐ Check ☐ Money Order

☐ Master Charge ☐ VISA ☐ American Express

Credit card # ________________ Exp. date __________ Signature ______________________

Name ______________________________

Address ______________________________

City ________________ State ________ Zip ________

80 microcomputing™
peterborough nh 03458

Coupon expires in 60 days

1/80

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 PREVIEW

WHAT TO LOOK FOR IN THE FEBRUARY 80

MAILING LISTS

Most small businesses and organizations need a mailing list system. Next month we have in-depth reviews of not one but FOUR systems. From Galactic Software, Radio Shack, Small Business Systems Group and The Bottom Shelf. All the systems are disk based and priced in the $100 range. Find out whether they delivery the goods —1st class or 4th.

FLOPPY PIMS

How to convert the Scelbi Personal Information Management System to run on disk BASIC.

VIDEO TUNE-UP

A focus on your monitor, make it both electrically safer and visually sharper.

BABYBUG

Frustrated with loading & reloading your editor/assembler? Fed up with hand assembly? Or perhaps just plain baffled by all those machine code bytes? Try BABYBUG, a simple machine code monitor that allows you to create and load machine code programs—easily and quickly.

SCOOP

The TRS-80 Disk And Other Mysteries, written by Harv Pennington, is a new book from the International Jewellers Guild. It covers everything you ever wanted to know about disk organization and file structures, including how to salvage unreadable data and "killed" files. The Apparat utility program "SUPERZAP" is used to dig deep into the heart of your disks. In the February issue of 80 we have exclusive extracts for your perusal.

FASTER, FASTER

Do your programs seem to take forever to run? Try this software controlled speedup modification.

ALL THIS PLUS MUCH MORE!

80 ADVERTISERS

• Reader Service inquiries not honored. Please contact advertiser directly.

STARTER KIT

EXATRON STRINGY FLOPPY FOR THE TRS-80

Recommended initial purchase:

Exatron Stringy Floppy with Level III BASIC	$249.50
5 Wafers each: 5′, 10′, 20′, 50′	40.00
Bus Extender, 2-for-1	15.00
ESF Machine Language Monitor	9.95
(Plus shipping and tax if applicable)	$314.45

SPECIAL PRICE FOR THIS STARTER KIT	$299.50
Sales Tax (California only)	______
Shipping and Handling	3.00
TOTAL	______

For more information see the current Exatron Stringy Floppy Owners Association Newsletter in *Microcomputing*.

If you have any questions about the product, about Exatron, or ESFOA, please call the Hot Line. Address letters to ESFOA, 3559 Ryder St., Santa Clara, CA 95051.

Stringy Floppy is a trademark of Exatron Corporation.

HOT LINE (For Calls Outside CA) 800-538-8559

excellence in electronics

exatron

3555 Ryder Street ▪ Santa Clara, California 95051
(408) 737-7111