$3.00 PER COPY
$3.50 in Canada

80-U.S.

The TRS-80 Users Journal

Volume III, Number 5

Sept/Oct 1980

TRS-80* Computer Compatible . . .

Quality is the real difference. Low price is merely a dividend.

High quality.
Competitive pricing.
And a proven track record.

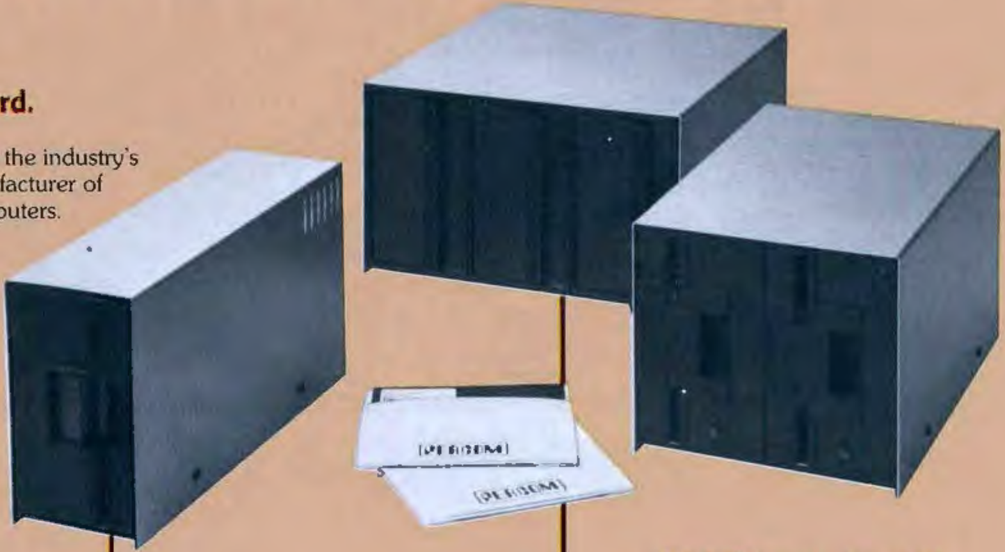
Three reasons why Percom is the industry's number one independent manufacturer of mini-disk systems for microcomputers.

And if you're looking at mini-disk drives, extra storage capacity is an added bonus.

In fact, you store almost one fifth more data on Percom TFD-100^{tm} drives and over two-and-one-fourth times as much on TFD-200^{tm} drives.

Besides extra testing and superior design, you get free, with each system, a software patch on minidiskette that not only upgrades TRSDOS* for operation with the newer 40- and 77-track drives, but also deglitches version 2.1.

Available in 1-, 2- and 3-drive configurations, Percom drives for the TRS-80* computer start as low as $399.



The gift of speech

Called Speak-2-Me-2^{tm}, this clever interface module makes a Texas Instruments' Speak & Spell† the voice of your computer — announcing, imploring, commanding with expressions and sentences created from the Speak & Spell† vocabulary.

Speech is controlled either at the keyboard or by your own Level II BASIC programs. Or by Percom minidiskette word games (available soon).

Speak-2-Me-2^{tm} is installed in the battery compartment of your Speak & Spell†, and power is provided from an ordinary calculator power pak. Supplied with an interconnecting cable, operating software and a comprehensive users manual, Speak-2-Me-2^{tm} costs only $69.95.

the Separator:^{tm} End "CRC error. Track locked out!"

This plug-in adapter virtually eliminates data read errors, a problem that plagues TRS-80* computer systems. The SEPARATOR^{tm}, so called, is installed in the Expansion Interface without modifying the host system. When installed, data and clock signals are reliably separated during playback, an essential function that the separator circuitry of the host computer performs very poorly. Price is only $29.95.

Note. Opening the Expansion Interface may void the Tandy limited 90-day warranty.

OS-80^{tm} uses Level II BASIC commands

With OS-80^{tm}, Level II BASIC commands are used for DOS and Disk BASIC functions.

You extend the OS-80^{tm} utility set indefinitely with your own BASIC disk-resident utilities.

OS-80^{tm} resides in less than 7-Kbytes of RAM.

The program is supplied on minidiskette along with a simple file manager, BASIC disk utilities and an OS-80^{tm} "Handbook" that you expand and maintain.

All are helpful programming examples.

Price, with instruction manual, $29.95.

NEW! OS-80^{tm} Machine Language SAVE / LOAD Utility. Only $14.95 on minidiskette w/instructions.

Quality Percom products are available at Percom dealers nationwide.
Call Percom's toll-free order number, 1-800-527-1592, for the address of your nearest dealer or to order direct.

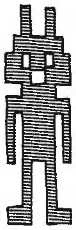
PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

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

PERCOM

^{tm} trademark of Percom Data Company, Inc.
* trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.
† trademark of Texas Instruments, Inc.

Missing some back issues? Check what's available (near the back of this issue.)



RETURN ADDRESS

____

____

____

Place
Stamp
Here

To:
80-U.S.Software
3838 South Warner Street
Tacoma, WA 98409

RETURN ADDRESS

____

____

____

Place
Stamp
Here

TO:

RETURN ADDRESS

____

____

____

Place
Stamp
Here

TO:

80-US SOFTWARE ORDER FORM 9/80

Order software by name(s) ______________________
____
____

Payment inclosed __________ OR Bill to my
□ M/C □ BankAmericard □ VISA

Signature ______________________ Exp date ________

Card No. ______________________

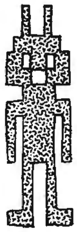
Name (type or print) ______________________

Address ______________________

City State Zip ______________________

WA residents add 5.0% sales tax All Software sent First class post paid

SOFTWARE ORDER FORM



+

80-US READER INQUIRY CARD – 9/80

Sirs:

I have seen your product/service in the 80-US JOURNAL and would like more information

ON______________________

Please send to:

Name ______________________

Address ______________________

City St Zip ______________________

80-US READERS:

Use the reader inquiry cards for further information on any of our advertisers products/services. Simply print the name and address of the advertiser on the reverse side and stamp and mail it.

+

80-US READER INQUIRY CARD – 9/80

Sirs:

I have seen your product/service in the 80-US JOURNAL and would like more information

ON______________________

Please send to:

Name ______________________

Address ______________________

City St Zip ______________________

FOR MORE INFORMATION ON ADVERTISED PRODUCTS OR SERVICES

HOMES for the TRS-80

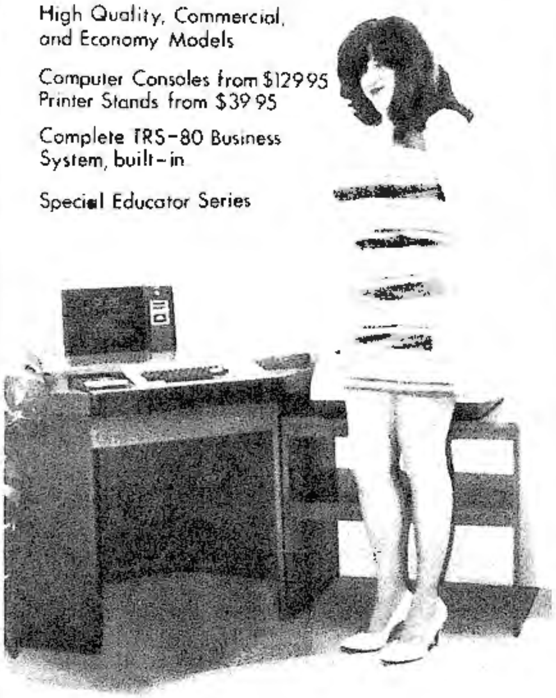
Featuring

High Quality, Commercial, and Economy Models

Computer Consoles from $129.95
Printer Stands from $39.95

Complete TRS-80 Business System, built-in

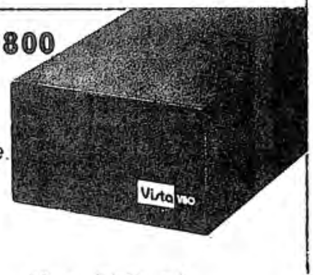
Special Educator Series



Custom furniture for the TRS-80 office or home decor.

PRODUCTS for the TRS-80

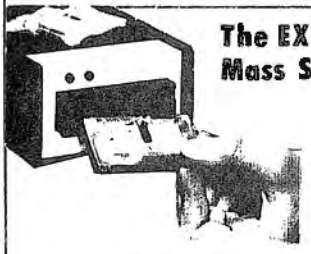
The VISTA V-80 & V-800 Disk Drive System



- *40 & 77 Track Units*
- *12 msec Access Time.*
- *High Reliability.*
- *Double Density.*
- *394K per Diskette.*

List $395.00

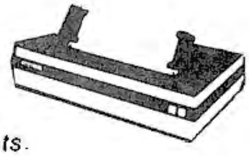
The EXATRON Stringy Floppy Mass Storage System



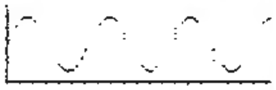
- *Access 15 x Cassette.*
- *Simplicity of Operation.*
- *Direct Keyboard Cont.*
- *Uses only 1K Memory.*
- *60K per Waffer.*

List $249.50

The BASE II Graphic Line Printer



- *Tractor/Friction Feed.*
- *5 Charactor Densities.*
- *Programable Aux. Fonts.*
- *2K Buffer*
- *72V x 99H Dot Res*



List $699.00

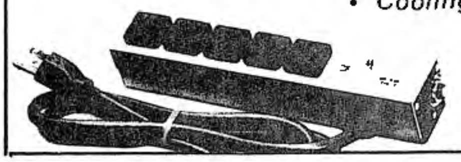
The AVS Green Screen Thing SC-80

- *Improved Contrast*
- *Reduces Eye Fatigue.*
- *Enhances Legibilty.*
- *Decreases Glare.*
- *System Looks Prof*

List $7.95
(Add $1.00 for shipping & handling, and California residents add 6% sales tax)

Miscellaneous System ACCESSORIES

- *AC 3 Wire, Multi-Outlets.*
- *AC Line Filters*
- *Cooling Fans*



AVS is a manufacturer, dealer, and consultant who specializes in products for the TRS-80 and their uses. You will note we do not handle multiple brands of each type of product. When we decide to handle a product, we try to analyze all the manufactures of that product. Then we select the manufacture we feel provides the greatest features, performance, and reliability relative to the cost and needs for the TRS-80. Though many times there may be more than one manufacture that meets our requirements, we prefer to only handle the one, and give our customers maximum support in the use of that product relative to the TRS-80.

AVS Custom Computer Furniture: Home for the TRS-80 takes the orphan modules of the TRS-80 and turns them into one homogenious unit. Our economy, commercial, and high quality consoles all house the keyboard, interface, cassette, and monitor. In addition, we have a special series for the schools. Our lines consist of computer consoles, line printer stands, storage hutches, carrols, and booths.

VISTA Disk Drives: The Vista drives are preferred by many system specialists, including ourselves, for their speed and reliability. They are available in both 40 and 77 tracks, providing memory in excess of 197K per disk, In addition, a module is available that allows the TRS-80 to run the Vista drives at double density, thus, providing more than 394K of memory per disk.

EXATRON Stringy Floppy: The stringy floppy is the ideal mass storage system for most home, school, and some small business systems, because of its simplicity of operation, high reliability, low cost, and speed. Unlike disk, it does not require the expansion interface nor the 11K of computer memory, to perform its many similar functions.

BASE II Line Printer: This printer performs many functions not even availabe in printers three times its price. Besides having a vertical density of 144 half dots to the inch, you can program your own character fonts. In addition, we have a special modification that allows the printer to run without an expansion interface.

AVS Green Thing: Our green screen works on both models I and II of the TRS-80. It not only performs the same tasks, but costs one-half to one-third of its competition.

Miscellaneous Accessories: For our customer convenience, we provide a series of AC outlet strips, line filters, and cooling fans.

AVS
AUDIO-VIDEO
SYSTEMS

2485 AUTUMNVALE AVE.
SAN JOSE, CA. 95132
PH. 408-946-1265

80-U.S.
JOURNAL

EDITOR/PUBLISHER
I Mike Schmidt
ASSOCIATE EDITOR
T R Dettmann
TECHNICAL EDITOR
James W Crocker
SPECIAL PROJECTS EDITOR
Larry S Panattoni
PRODUCTION MANAGER
Cindy Wood
CIRCULATION MANAGER
Margaret Farrell
SOFTWARE DEVELOPMENT
Leo Christopherson
Roy Groth
Kristi Schmidt
REVEWERS
Cameron C Brown
Pat Perez

PRINTED BY
Peninsula Gateway Press
Gig Harbor, Washington

DISTRIBUTORS
CANADA
Micromatic Systems Inc
1303 Powell Street
Vancouver, B.C. Canada
V5B 1G6
EUROPE
Hofacker Verlag
Tegernseer Strasse 18
D-8150 Holzkirchen/Obb
West Germany
UNITED KINGDOM
The Software Exchange
146 Oxford Street
London, W. 1 England

EDITORIAL

It isn't really that hard to make it in today's software market. Here are a couple of ways to do it.

First, you must realize that a great number of TRS-80 owners honestly feel they have a right to any software they can lay hands on. Many also have an insatiable appetite for *any* program, even if they cannot use it themselves - it makes great trading material.

Don't try and write anything yourself. After all, it is too hard, you don't have any good ideas anyway, and if you do, someone may steal it. Besides, there are good software authors, and all you need do is figure how to cash in on their work.

Set yourself up as a "Program Review Agent", or maybe as a "Software Library". Buy a legal copy of all the software worth having, make a list of it and offer it to your clients.

Realizing that a large amount of what passes as software is bad, you can appeal to the common user complaint that you have to buy it before you find out it is no good. Sell memberships in your organization for "X" bucks per head and offer them the chance to "look" at the software before they buy.

If Joe Doaks in Podunk checks out one of your offerings and makes his own copy before returning it, there is nothing you can do about it, right? And you did not *sell* the software to him, did you?

In the middle of the night, when you wake up, *you* know that this is exactly what you expected him to do, else you would have no reason to exist. Never admit to this in public.

So there you have a way to do it. There is yet another way:

Contact the original owner of the software you wish to sell. Negotiate any sort of license agreement with him - agree to all his terms. After the agreement is signed, go ahead and make and sell his software - and ignore the licensor completely. Also ignore all the terms of the contract, and sell as much and as quickly as possible.

After some time, the licensor will void your agreement due to non-performance (if he wrote it correctly in the first place). But now it is up to him to prove how many copies you have sold. He has to weigh the price of legal action against what he may be able to recover. You, of course, have kept no records and when he calls you are always "out".

If you can live with yourself and can sucessfully dodge all your licensors, this is one way to do it.

In case you are beginning to wonder - we do not condone either of the above methods. But this sort of thing is going on daily in the software industry.

Right, wrong or indifferent, here is what may happen if it keeps up: Authors (and there are very few really good ones) are simply going to quit writing. Some already have. They are going back to bricklaying or welding or whatever they were doing before. If they stay in the business, it will be to write custom business programs for one customer and charge excessive prices, knowing full well that one copy is all they can sell.

After everyone has a copy of all the present software, there will be little more. Users can then play Startrek till it comes out of their ears, or give up and learn tennis or bridge.

The software industry may be going to hell in a handbasket, and if it does, personal microcomputing may well go with it.

If that thought dosen't bother you, then go ahead and take that "free" software - while there is still some to take.

The fact that we create our own reality could not be more amply demonstrated than by this situation.

Mike

The 80-U.S. JOURNAL
VOL III Number 5
Sept/Oct 1980

ISSN Publication # 0199-1035
Published bi-monthly in Jan, Mar, May, Jul, Sep and Nov by **80-Northwest Publishing Co, 3838 South Warner St., Tacoma, WA 98409. SUBSCRIPTION price in the United States is $16./1 yr, $31./2 yrs, $45./3 yrs.**
Canadian Subscribers contact Micromatic Systems Inc 1303 Powell St., Vancouver, BC Canada V5B 1G6. **ALL OTHERS** $24./1 yr, $47./2 yrs, $68./3 yrs remitted in US funds. All except US, Canada and Central America are sent via Airmail.
Printed in the United States of America
POSTMASTER: Please send change of address form 3579 and undelivered copies to 80-U.S. Journal, 3838 South Warner St., Tacoma, WA 98409. **Return postage is guaranteed**
Second Class Postage PAID at Tacoma, Washington

80-U.S.

The JOURNAL for TRS-80* Users

*TRS-80 is a trademark of the Tandy Corporation

Vol III Number 5 Sept/Oct 1980

CONTENTS

FEATURES

REVIEWS

DEPARTMENTS

No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this publication, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of any information contained herein. Please address all correspondence to: **80-US JOURNAL, 3838 South Warner Street, Tacoma, Washington 98409 (206)475-2219.** **ADVERTISERS:** The JOURNAL will accept limited relevant commercial advertising which pertains to, or is for use on or with, the TRS-80 Microcomputer Write for a current rate schedule. **WRITERS:** We constantly seek material from contributors. Send your TRS-80 related material (except that which has been previously published) You may send programs on disk or cassette - they will be returned provided you include return postage. Generous compensation is made for non-trivial works which are accepted for publication. The JOURNAL pays authors upon acceptance rather than on publication. Allow 3 to 4 weeks for review of your submitted manuscript/programs

Letters to the Editor

At the time I wrote HEXDUMP (80-U.S. Mar-Apr 80) all I needed was a plain memory dump, and it just didn't occur to me to provide an ASCII character display. However, the other day I was printing out a very long disassembly which wasted an awful lot of paper by trying to disassemble individual ASCII characters in tables as separate Z-80 instructions, and this letter is the result.

The following quick modifications to HEXDUMP provide all printable ASCII characters for the machine code being dumped, in addition to the original hex:

ADD the following instructions:

```
255 AS$="      ": REM CLEAR ASCII
    STRING
375 REM GET ASCII CHAR IF
    PRINTABLE, OTHERWISE DIT
376 IF ME > 31 and ME < 128 THEN
    AS$=AS$+CHR$(ME) ELSE
    AS$=AS$+"."
```

REPLACE the following instructions:

```
110 CLEAR 100
450 IF Y$="Y"THEN LPRINT AS$ ELSE
    PRINT A$
```

CHANGE 430 by removing the ending " "; in order to make room on the CRT display. This removes all blanks in the hex part of the dump, so you may want to add the following to put one blank in the middle:

```
435 IF X=7 THEN PRINT " ";
```

Louise H Frankenberg, Pasadena, MD

Thanks for publishing the announcement of our club meetings. So now we have this problem -- our TRS-80 Group has outgrown the meeting room at the Bellaire Chamber of Commerce where we met for a year and a half, but people still keep coming to the old place. And so does the mail.

To help visitors and members find the right place at future meetings, will you be kind enough to print the following?

THE 80-USERS OF HOUSTON club meets the first Wednesday of each month at 7:30 PM. Anyone interested in microcomputers - for personal or business use - is welcome. For the location contact Ben Taylor, (713) 664-5823. (3723 Purdue, Houston, TX 77055)

Our former notice of the Group meetings gave the street address. So the Bellaire Chamber of Commerce has been getting lots of mail about little computers - and they do not like it! If your mailing list had "The TRS-80 Microcomputer Group of Southwest Houston" on it, you might revise the name and address.

B C Taylor, Houston, TX

(Nice to see you are growing! (Ed)

I'm really getting a charge out of all of you jumping on the band-wagon about software pirates. The idea that someone would give a copy of a program away! Well I'll tell you guys something - if someone at a meeting gives me a program, I don't care if it sells for 7.95 or 150.00, *I'm taking it!*

I'm fed up with all your magazines (or rags) printing the same stuff only a month later with a few changes. Do you think I like getting 5 different ones in the mail each month showing me 5 different ways of doing the same program?

And how about purchasing a program from Softside only to see it *free* in 3 magazines later. And that free book for going with Super Wayne's 80-Micro - well you get what you pay for (0=0).

I'll stop taking software when 1, copy machines are taken out of libraries. 2, you or any member of your staff gets rid of a Betamax for pirating movies. 3, any one of you stop tape recording music off the radio and 4, when any member of any of your families goes to a quick copy center to get 40 copies of sheet music for "church choir" instead of purchasing 40 original copies at the music store. Get off your soapboxes and do a magazine - I can get sermons on sunday!

Unsigned, Philadelphia, PA

(Enjoy your copying, as long as there is still something out there to copy. Attitudes like yours will make authors turn to something else, and will eventually cause the quiet demise of home and hobby microcomputing. After a few years you will find you have nothing to do with your expensive toy except play the same old pirated games, over and over. I don't own a Betamax, don't use the copy machine at the library, tape music off the air and my name is Mike Schmidt, what's yours?? If you honestly feel you have a just cause you could sign your name to your letter - or do you feel guilty about something? -Ed)

I own an electronics shop, am manager of a pawn shop, and juggle half a dozen other semi-businesses and hobby activities. The number one stumbling block and massive time-burner of the whole mess is simply keeping track of what's going on where, when, and at what price. TRS-80 is slowly but surely relieving that workload - and 80-U.S. is nursing and goading me as I plug away at the keyboard. No matter what information I stumble across, I've needed to do some tailoring to get it to spit back results suitable to my picky applications, and 'tis a hell of a lot easier to work with information written directly for the TRS-80 than it is to translate ideas from 14 other breeds of machine.

Wayne M Davis, Bellingham, WA

Please accept my hearty congratulations on a most enjoyable and helpful Journal. I especially appreciate the "new" View from the Top of the Stack. Excellent.

I am a theology professor at Inland Empire School of the Bible in Spokane. We use a 32K 2 disk system (Mod I) for mailing list and bookkeepping applications. I'm writing all the programs myself, and find helpful material in every issue of the Journal. Please.. keep it up! Bi-monthly is perfectly all right. It takes me about that long to absorb it all anyway!

Bruce W Gore, Spokane, WA

I would like to obtain information about interfacing a TRS-80 to an old model of IBM Selectric (Model 71). I would also like to know if any commercial interfaces are available for this purpose. If you or any of your readers have this information I would appreciate receiving it.

N Vijayan, 1332 Notre Dame Dr. Davis, CA 95616 (916) 756-4955

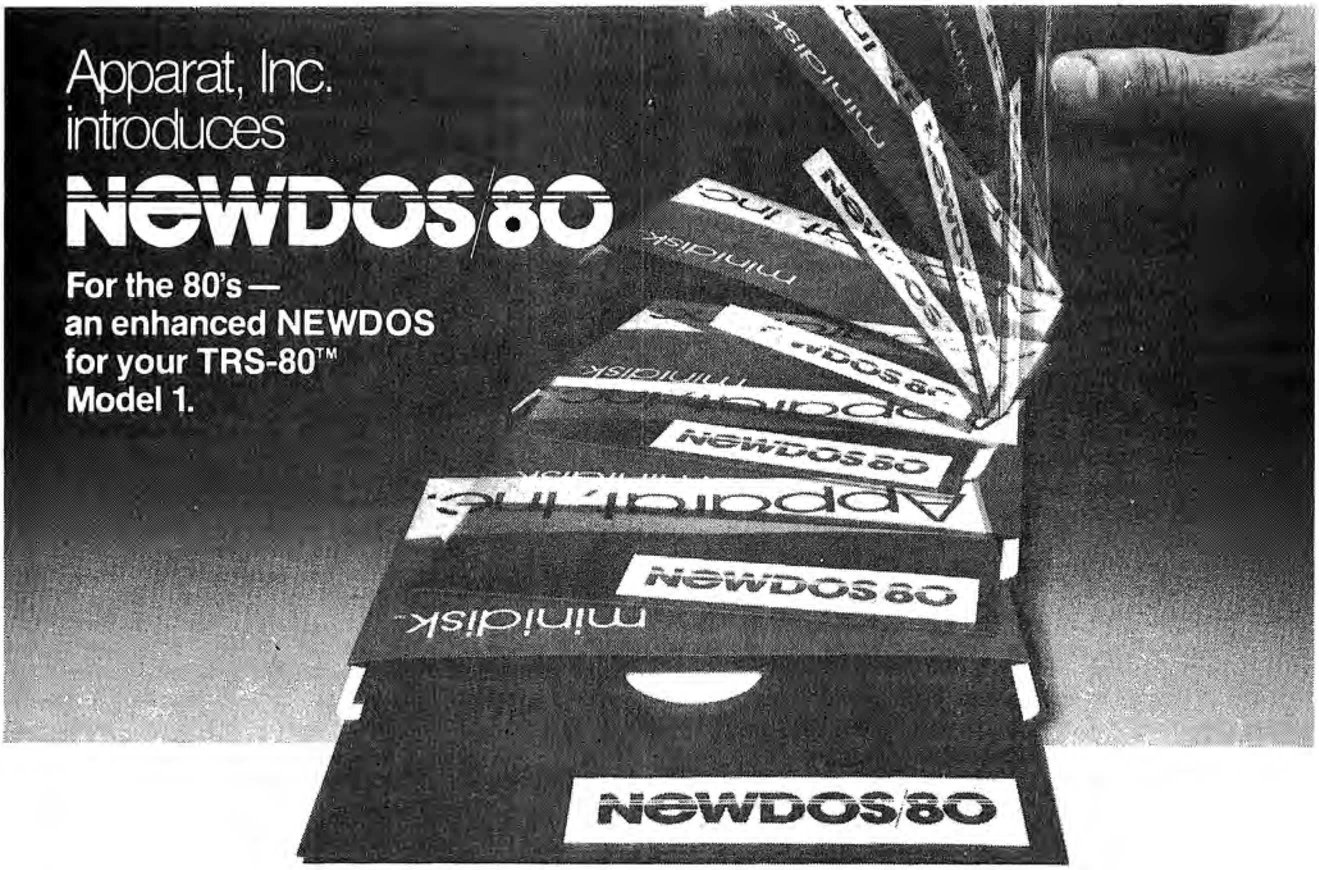
With a little bit of effort I've learned to use the numeric keypad by touch. This has been especially helpful for speeding up the entry of long data statements, i.e., some of the music and sound programs published lately. However, data input still requires a juggling act. With one hand on the keypad and the other hand for the comma key, a major benefit of the keypad is lost.

This prompted my search for a better way and I think I've found a solution. First I enter all the data statements in the program, but instead of using the comma between numbers I use the period (decimal point) on the keypad. I then enter the following temporary program using line numbers higher than those used for the data lines. This short routine will convert the periods in the data lines to commas.

```
1000Y=PEEK(16634)*256+
    PEEK(16633)
1001 FOR X=17134 TO Y
1002 IF PEEK(X)=46 POKE X,44
1003 NEXT X
```

Typing RUN 1000 followed by a list command will show the periods have been converted to commas. After

(Continued on page 6)



Apparat, Inc. introduces NEWDOS/80

For the 80's — an enhanced NEWDOS for your TRS-80™ Model 1.

Apparat, Inc., announces the most powerful Disk Operating System for the TRS-80®. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2.1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2.1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2.1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:

- New BASIC commands that supports files with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disks drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOS READY" or ">READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST".
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful chaining commands.
- Print Spooler.
- DFG function; simultaneous striking of the D, F and G keys will allow the user to enter a mini-DOS to perform some DOS commands without disturbing the resident program. (e.g. dir while in scripsit.)
- Upward compatible with NEWDOS 2.1 and TRSDOS 2.3.
- Includes machine language Superzap/80 and all Apparat 2.1 utilities.
- Enter debug any time by pressing 123 keys. Also allows disk I/O.
- Diskette "Purge" command.
- Specifiable system options (limited sysgen type commands).
- Increased directory capacity.
- Copy by file commands.

NEWDOS/80 with all of the NEWDOS + utility programs, many of which have been enhanced, is priced at just $149.00 and is available at most TRS-80 dealers.

As with 2.1, NEWDOS/80 relies on the TRSDOS and Disk Basic Reference Manual published by Radio Shack. NEWDOS/80 documentation supports its enhancements and upgrades only.

VISA | master charge THE INTERBANK CARD

Apparat, Inc.

MTI MICROCOMPUTER TECHNOLOGY INCORPORATED

TO PURCHASE NEWDOS/80, COMPLETE AND MAIL TO:

Apparat, Inc.
4401 S. Tamarac Parkway
Denver, CO 80237
303/758-7275 303/741-1778

--OR--

Microcomputer Technology, Inc.
3304 W. MacArthur Blvd.
Santa Ana, CA 92704
714/979-9923

□ Check □ Money Order □ Master Charge □ Visa

Card No ____________ Expiration Date ____________

Colo residents add 6 5% sales tax Cal residents add 6% sales tax
Add $10 00 postage and handling

Please rush ________ NEWDOS/80 @ $149 EACH TO:

Name ____________________________

Address ____________________________

City ________ State ________ Zip ________

Phone ____________

U

(Continued from page 4)

deleting the comma conversion subroutine the remainder of the original program can be entered.

Andy Kell, Palm Springs, FL

(Clever idea! (Ed)

Over the past year I have grown to expect the highest quality in the articles you include in your magazine. I must admit to being a little upset by the SCRIPSIT article by Barry Kornfeld. I run two timesharing users groups on MICRONET and THE SOURCE and one of the biggest headaches for me is trying to correct wrong information which keeps finding its way to various members of the groups. We have written extensively about both PENCIL and SCRIPSIT and we use both programs in our daily activities which deal with the time sharing systems. For those readers who are not members of any such group I would like to correct some errors and some very misleading statements which appeared in your SCRIPSIT review.

In comparing the PENCIL and the RS lowercase mods, Mr Kornfeld tells us that PENCIL has a toggle switch which SCRIPSIT handles through software. He goes on to say that this *somehow* screws up the the video RAM. *Somehow?* First of all it is not handled through software, it is handled through hardware! This is why RS replaces the character generator ROM, not just to get "decenders which look nice". The PENCIL mod if left switched on without a lowercase driver will produce garbage (control codes) on the screen The RS character generator prints the correct uppercase letter instead of the garbage. This lets the lowercase mod be on constantly whether or not a driver is present. It is important to note that the new ROM does not convert the control codes to uppercase letters, it just generates uppercase letters when one of these codes is encountered. This is why the 'JKL' function of NEWDOS, and other routines which PEEK video RAM, malfunction when the RS mod is running without a lowercase driver being active. APPARAT has fixed the NEWDOS 'JKL' problem and most software authors have included a little routine which PEEKS video RAM and adds 64 if it finds a value under 32. This accounts for another fact not even mentioned in the article which is that PENCIL with the RS mod will print M's and L's instead of right-arrow and down-arrow for C/R and Form Feed.

As far as the VTOS PENCIL/FIX making PENCIL reasonably compatible with the RS mod, let me say that it makes it totally compatible. My user group published the same type of patch for NEWDOS and TRSDOS thereby making PENCIL running under any DOS compatible with the RS mod. With this patch PENCIL can load any ASCII file with any filename or extension. The most recent patch changes PENCIL to use the BREAK key as the Control Key so no extra hardware modification is needed beyond the RS mod. These patches have been reprinted in user group publications around the country.

I have taught many people to use both PENCIL and SCRIPSIT and there is no extra time needed for SCRIPSIT. It should have been mentioned that once a file has been printed with SCRIPSIT, the print formatting commands remain as part of the file and nothing has to be done to print the file at a later date.

As far as editing Basic programs, this was not something to be skipped over so lightly! Most of the people I know who use SCRIPSIT also use it for editing their Basic programs. There is an important difference here when using PENCIL and SCRIPSIT. PENCIL will crash if it encounters a line longer than 60 characters without a space. This makes PENCIL very awkward indeed for editing Basic programs. SCRIPSIT can handle any line length without crashing but the user must be certain to use the ASCII switch when saving the edited programs (S,A).

The review was written by someone with a Diablo printer running through the RS232. This is certainly not the majority of us. As far as I know the line feed problem Mr Kornfeld mentions is peculiar only to the RS232 driver. I have never heard of a similar problem when using the parallel driver. Indeed, Mr Kornfeld has it backwards when he tells us that only C/R's are output. A real problem is the fact that when issuing form feeds and during vertical centering, only line feeds are output. This creates a problem for Selectrics which will ignore the line feeds. Selectrics do not run fine with SCRIPSIT as Mr Kornfeld tells us they do. There is a very small patch for this problem:

All references are relative sector positions.
At F000046 Change 3E 0A 32
(X'5242) to 3E 0D 32
At F01FF0 Change 3E 0A CD
(X'7170) to 3E 0D CD
At F02004 change 3E 0A C3
(X'7184) to 3E 0D C3
These should even work for a Selectric connected to the RS232 port.

Another important omission is the fact that SCRIPSIT will not work with NEWDOS. APPARAT has been sending the patch to all who ask for it. Write to APPARAT for all the latest NEWDOS patches. Their address is APPARAT Inc 6000 E Evans Ave Bldg 2, Denver, CO 80222. What you get from them will include the patch to make SCRIPSIT work correctly with NEWDOS.

Also left out of the review was the fact that SCRIPSIT Disk I/O is at least three times faster than PENCIL's and the very important fact that SCRIPSIT uses a FIFO keyboard driver which means that no characters are lost when you approach the end of the line and the text is adjusted. Typists must slow down at the end of every video line when using PENCIL.

As a matter of personal opinion I disagree with Mr Kornfeld's choice of the major omissions in each Word Processor. Listing the lack of a page-wait as a major omission seems a little overstated and again reflects the reviewer's own hardware setup. If this is a problem, patches for this have also been published by user groups (including my own). The major omission from SCRIPSIT we have already discussed is the line feed problem for Diablo Serial printers.

In my opinion, the one huge omission from SCRIPSIT is the lack of any disk file functions without leaving the program. There is no way to get a directory or KILL a file without exiting the program. The major omission from PENCIL is the lack of being able to edit text files which contain long lines without spaces.

It seems that Mr Schrayer thinks the SCRIPSIT advantages good enough to include in his new PENCIL for the Model II. As for me, I use both programs every day and I would not want to give up either. I just wanted to get some facts into this discussion that I felt were important for people who are familiar with the TRS-80 to know.

Richard Taylor, Union City, NJ

(Judging from your comments and those of several others, it appears we were amiss in editing that review. Thank you for your comments (Ed)

In two short years the TRS-80 has become the equivalent of the "IBM" of microcomputers. As a result of their dominance, a host of products have been developed by clever and creative people to give the S-80 support. As a direct result, we all now have the opportunity to do things with our computers besides turning them on and off.

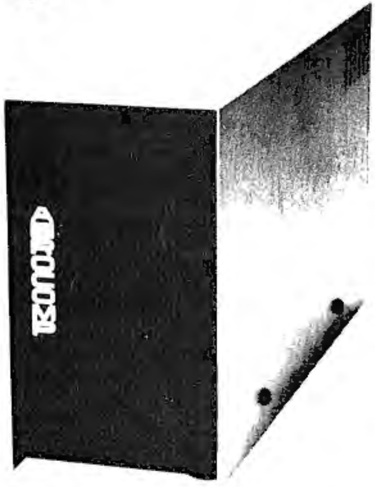
This list of things also includes a myrad of business application programs that closely fit almost any reasonable need; tutorials to teach us more about our computer, our environment and ourselves; all of which has the capacity to expose us to new concepts and skills in a wide range of interests from social sciences to technical pursuits - and of course, the fun stuff, games.

While all of this has occurred, we can still be considered on the ground floor of the microcomputer industry. A lot of talented people have been creating a pile of software intended to make life easier, or uplift our intellect. I consider most of them competent people who are realistically trying to satisfy many needs with quality products. This seems to have spawned a fringe industry of some who are trying to cash in on the software boom. You know them: The ones really not contributing to our growth, the ones who have their ads prepared by word merchants which describe products that nowhere near approach the product itself, the ones that have no sensitivity to their customers' needs.

Although these cases are few in number, they are still sufficient to cause any consumer to harbor pessimism for any future advertising - even though it truthfully describes a product.

(Continued on page 8)

NEW DISC DRIVES FOR TRS—80* "FLIPPY"—250K bytes/disc FAST—5ms Track to Track Access



REAL VALUE

AEROCOMP offers the best value in microcomputer disc drives on the market today! Reliability, features and cost tough to beat. We deliver...and we stand behind our products, as evidenced by the only FREE TRIAL OFFER in the industry. Examine your systems needs and order today!

MYSTERY REMOVED

There appears to be some confusion in the terminology used to describe disc drives and their features. Here's what we mean:

•FLIPPY Allows the use of both sides of a diskette with a single-headed drive by simply turning the diskette over (Model 40-1).

•TRACK DENSITY specified in tracks per inch (TPI). Refers to the number of tracks per radial inch on the diskette. Typically 48 TPI=40 usable tracks and 96 TPI=80 useable tracks.

•DOUBLE DENSITY refers to recording density in bits per inch (bpi). Typically single density means data can be recorded up to 2,938 bpi; double density means data can be recorded up to 5,876 bpi.

•DOUBLE-SIDED refers to number of read/write heads. Single-sided is one head, read/write one side only; double-sided is dual heads allowing read/write operations on both sides of the diskette. A double sided drive appears as two separate drives to the controller.

•CAPACITY unformated capacity is the total amount of storage space available on a diskette. Typically 125K bytes on a 40 track 5.25in. diskette. Formated capacity is the total USABLE storage space on a diskette. Typically 102K bytes on a 40 track 5.25in. diskette.

•ACCESS TIME the time required for the head to move from one track to the next. Typically 5 to 40 milliseconds (ms).

FREE TRIAL OFFER

Order your AEROCOMP Disc Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return it, packed in the original shipping container, for a FULL REFUND. We have complete confidence in our products and we know you will be satisfied! ORDER TODAY!

WARRANTY

We offer you a 90 day unconditional warranty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes necessary, our service department is fast, friendly and cooperative.

100% TESTED

AEROCOMP Disc Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!

ORDER NOW

MODEL 40-1 **DISC DRIVE $339.95ea.**
Single-sided, "Flippy", 48TPI
(40 track; single density unformated 125K bytes/side; double density unformated 250K bytes/side).

MODEL 80-2 **DISC DRIVE $439.95ea.**
Double-sided, 48TPI.
(80 track/40 per side; single density unformated 250K bytes; double density unformated 500K bytes).

MODEL 80-1 **DISC DRIVE $439.95ea.**
Single-sided, "Flippy", 96TPI
(80 track; single density unformated 250K bytes/side; double density unformated 500K bytes/side).

MODEL 160-2 **DISC DRIVE $595.95ea.**
Double-sided, 96TPI.
(160 track/80 per side; single density unformated 500K bytes; double density unformated 1 megabyte).

All models are capable of single or double density and are complete with power supply and silver enclosure. Send for information on AEROCOMP 2- and 3-drive systems available in 40 and 80 track.

SELECT EITHER A 2-DRIVE OR 4-DRIVE CABLE FOR USE WITH YOUR DRIVE(S):

2-DRIVE CABLE (for use with 1-or 2-drive systems) **$24.95ea.**

4-DRIVE CABLE (for use with 1-, 2-, 3-or 4-drive systems) **$34.95ea**

Add $1.25 shipping and handling

MINI DISKETTES (5.25 in), Box of 10 **$29.95** add $1.25 shipping and handling

DISC OPERATING SYSTEMS	
NEWDOS+(40 TRACK)	**$109.00**
NEWDOS/80 (80 TRACK)	**$149.00**

SPECIAL COMBINATION OFFER

Model 40-1 Disc Drive	$339.95
2-Drive Cable	24.95
Disc Operating System (NEWDOS+)	109.00
Reg.	**$473.90**
Special	**$399.95**

SPECIAL OFFER AS ABOVE EXCEPT WITH NEWDOS/80

Reg.	**$513.90**
Special	**429.95**

Free Freight with Specials

To order by mail, specify Model Number(s) of Drive, cable, etc. (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request COD shipment. Texas residents add 5% sales tax, add $4.00 per drive for shipping and handling. Please allow 2 weeks for personal checks to clear our bank. No personal checks will be accepted on COD shipments-cash, money orders or certifed checks only. You will receive a card showing the exact COD amount before your shipment arrives. Be sure to include your name and shipping address. WE SHIP PROMPTLY! In the event there is a slight delay, you will be notified of the shipping date and we will NOT deposit your money order or charge your bankcard until the day we ship!

FOR FAST SERVICE ON VISA/MASTERCHARGE or COD ORDERS CALL TOLL FREE NOW! (800) 824-7888, ASK FOR OPERATOR 24.
In California dial (800) 852-7777, Operator 24. In Alaska and Hawaii dial (800) 824-7919, Operator 24.
TOLL FREE LINES WILL ACCEPT ORDER ONLY!

For Applications and Technical information, call (214) 337-4346 or drop us a card

COMPARE AND BUY AEROCOMP!

	"FLIPPY"	ACCESS TIME (track to track)	HEAD LOAD SOLENOID	DISC EJECTOR	CAPACITY (unformated single density)	EASY-ENTRY DOOR	FREE TRIAL
AEROCOMP	**YES**	**5ms.**	**YES**	**YES**	**250K bytes (both sides)**	**YES**	**YES**
RADIO SHACK*	NO	40ms	YES	NO	109K bytes	NO	NO
PERCOM	YES	25ms	YES	NO	250K bytes (both sides)	YES	NO
MPI	NO	5ms	YES	YES	125K bytes	YES	NO
SHUGART	NO	40ms	YES	NO	109K bytes	NO	NO
SIEMENS	NO	25ms	YES	NO	125K bytes	YES	NO
TANDON	NO	5ms	NO	NO	125K bytes	NO	NO
PERTEC	YES	25ms	YES	NO	250K bytes (both sides)	NO	NO
BASF	NO	12ms	YES	NO	125K bytes	NO	NO

Factual material from current manufacturer s data sheets is believed reliable but cannot be guaranteed comparing Aerocomp Model 40 1 to similar models

The TRS-80* expansion interface limits the track to track access time to 12ms

*Trademark of Tandy Radio Shack

AEROCOMP, INC.

Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

Having recognized an emerging need to review software, Dick Clope launched his "80 Software Critique" to give us a chance to read some in-depth software profiles, impartially written as a software survival manual. He even got the approval and endorsement of several prominent sources in the industry. I sadly note that Dick will close down his critique after Issue 4, and this service to all of us, suppliers as well as consumers, will end.

Recently a new "service" has been brought to my attention which proports to let the subscriber "...try a Level II program before you buy it". This appears to be a "lending library" in disguise. A lending library is an activity which makes programs available to their subscribers. These "services" are not authorized to make programs available to a third party by program author, and tend, by their nature, to deprive the author of the program royalties to which he is entitled. This clearly is not contributing to the advancement of the industry.

This will fill the greedy need of some subscribers to the "service". There are "Program-a-holics" who cannot see that there is a reasonable limit to how much software you actually need.

There is not much incentive to develop software if there lurks in the shadow those who would deprive the author of the royalties they are entitled to.

The industry has now reached an important crossroad. If something effective is not done to contain the efforts of those engaged in unethical activities in our hobby, it stands a good chance of going down the drain. If serious legal action becomes necessary to resolve copyright matters, the micro revolution may be squashed in the process. By creating hurdles sufficient to discourage any clearly questionable activity from thriving, we can enjoy the freedom to use our computers as we wish.

After having carefully considered what lays before us, I propose a simple but effective plan to assist both the software author as well as the consumer. This plan would call on the authors to subsidize evaluations of their own products and fairly assess them in several areas using a 1 to 10 scale. This assessment would cover the obvious categories of interest to a purchaser of the product and could take the following form:
Name of program/name of supplier/ease of use/instruction rating/description fits ad/can learn from it/instruction understandability/hardware needed/etc...

A one-line evaluation would cost 50¢/program and would be objectively evaluated by people in various backgrounds who have no interest in ripping off the author.

Tentatively, the ABS Quality Control Foundation will provide this service which will be subsidized by the authors themselves to have their programs appear on the list permanently and will be available to anyone interested in the list free by sending a self-addressed, stamped envelope.

This proposal will need the support of every publisher, since this type of non-profit service will require constant, visable exposure, notifying the readers of the existence of this service and how to take advantage of it. A few lines tacked to the bottom of the masthead as a standard practice is suggested.

It is easy to see that a simple idea like this, after running for 6 months, would be effective because you could safely assume anyone doing business and not listed could have something to hide.

Unethical is an antiseptic way of describing something we all acknowledge as being crooked, and it is only due to legislative oversight that it is not, in fact, illegal.

If a critter has webbed feet, feathers, looks like a duck, waddles and quacks - I'm going to call it a duck!

I welcome comments, criticism and suggestions regarding the above.

Thomas Frederick
PO Box 8297
Ann Arbor, MI 48107

from The Programmers' Guild

action PROGRAMS

Are you fed up with namby-pamby computer simulations that offer no challenge? Are you disgusted with utilities that have only limited applications? THE PROGRAMMERS' GUILD has the answer for you NOW.

IRV*

The most powerful keyboard utility program ever conceived for the S-80. Allows up to 255 characters per key to be programmed into ANY KEY ON THE KEYBOARD (including enter and break). Full cursor control allows awesome editing features including individual line renumber and block movement inside a basic listing. AND THAT'S ONLY FOR OPENERS! Full control of anything that appears on the video display means that with certain keystrokes a whole graphics display CAN BE SAVED AS PACKED STRINGS! And with the numbering of your choice! There are so many features to IRV that it is impossible to describe in a single paragraph.

Order #U100 Tape $25.00 Disk $29.95

ELECTRA-DRAW

Here is a disk based sketch art program for the true computer artist. Featuring all linear modes available to the S-80, (horizontal, vertical, and diagonals) plus white or black background screen, it is possible to create unlimited drawings, maps, graphs, charts, title pages and other graphics endeavors with simple key-arrow manipulation. Once completed, ELECTRA-DRAW allows the user to save the screen to disk in the form of user numbered packed strings for reuse or merging. Now the painstaking job of illustrating programs is a thing of the past. Simply ELECTRA-DRAW them.

Order #U200 Tape (for Disk) $19.95 Disk $24.95

LOST DUTCHMAN'S GOLD*

$14.95 on tape $19.95 on disk

Deep in Arizona's Superstitution mountains, under the watchful eye of Apache Indians, the fabled LOST DUTCHMAN'S GOLD MINE awaits the adventurer bold enough to challenge the fates. With only a mule and a Winchester, and the ghost of an old gold prospecter named BACKPACK SAM to guide you, do you dare to venture into the misty mountains, where it is said that the Devil himself, protects the LOST DUTCHMAN'S GOLD?

DEATH DREADNAUGHT

This mind chilling triller will freeze the blood in your veins. You are trapped aboard an ancient alien battlecruiser whose entire crew was murdered brutally. You are the scout team's only survivor. Your ship's been smashed, but one alien shuttlecraft remains intact. As you move about the wreck the danger increases. You must secure air, food, fuel and batteries. Now, as precious seconds of your life tick away you suddenly realize YOU ARE NOT ALONE! The creature that ripped this ship apart is ALIVE and waiting for you to make a mistake. CAN YOU ESCAPE? Or will YOU be the creature's next victim.

$14.95 on tape $19.95 on disk

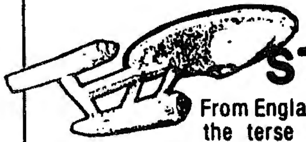
WARNING: DEATH DREADNAUGHT IS RATED R due to EXTREME descriptions of VIOLENCE

SPIDER MOUNTAIN ADVENTURE

Beyond the desolate outback of GRAYLOCKLAND lies the lair of the spider SHELOB. Once a wizard's cavernous fortress, SPIDER MOUNTAIN is now a deadly trap from which only the most skilled adventurers emerge. Treasure awaits the fortunate. DEATH stalks the foolish.

Can YOU solve the riddle of the SPIDER MOUNTAIN ADVENTURE, and come back alive?

$14.95 on tape $19.95 on disk



STAR TREK 4.0

From England's hottest micro-programmer comes the terse Starfleet Command, "DESTROY ALL KLINGONS". Many unique features make STAR TREK 4.0 a winner including COMMAND PREVIEW and REALISTIC COMBAT EFFECTS. More STAR TREK than ever thought possible in a 16K program. In military lingo, 4.0 means perfect. If perfection is what you crave in your TREK programs then look no further. STAR TREK 4.0 is for you.

Order #S100 Tape $14.95

SPECIAL

TRIPLE ADVENTURE DISKETTE
$35.00—Save $10.00

DOUBLE TAPE OR DISKETTE
(ADVENTURES OF YOUR CHOICE)
$24.95

DEALER INQUIRIES INVITED

The Programmers' Guild

P.O. BOX 66, PETERBOROUGH, NH 03458 (603) 924-6065

SEND CHECK OR M/O (SHIPMENT WITHIN FIVE DAYS OF RECEIPT) C.O.D. ACCEPTED

ITEMS At Random

We start our third year of publication with this issue, and just to prove that things haven't changed *that* much, here are some corrections for last issue:

CORRECTIONS

Page 26, Jul-Aug 80, *Fast Array Save & Load, I Barry Geller,* line 00100 following line 00130 (in the side by side listing) does not belong there at all, throw it out. Also on that page in the Basic listing, line 30020, the last B2% should be a B4%. On page 35 of that issue, the two listings are not labeled, and the lower one is Figure 1 while the upper one is Figure 2. On page 48 of that issue, the remark after line 00120 (0B00H FOR 32K) should be (0B000H FOR 32K).

Ranger Available on Tape

Rudolf Salinger, who wrote the RANGER program in our Jul-Aug 80 issue (page 60) has informed us that for those who tire of typing in such long programs, he will make it available on cassette for $5.00. Address your requests directly to him, at 5312 Bloomfield Drive, Midland, MI 48640

Subscription Renewal

We have been talking about our labels for some time now. Finally, if everything goes right, you will find a control number on your label. This number lets us find your record fast. Please refer to that number when writing or calling about your subscription.

Renewing early is a sure way to keep the JOURNAL coming to you on an uninterrupted basis. It cuts down on the paper work and hassle for us too. And by the way, have you thought of giving the JOURNAL as a gift?

In this issue

John Hind starts the thing off with a way to add new functions to Level II Basic, and shows that there is a way to understanding ROM and such.

Bill Atchison gives us a neat Basic Z-80 Disassembler that occupies just 6K. For a Basic program, it does a rather quick job, but you may note it takes a little longer to decipher the IX and IY instructions. It also gives the answers in Hex, Decimal and when possible, in ASCII

Bob Liddil (alias Captain 80) makes his debut with 80-U.S. in this issue. See his offering on page 34. We hope his stay with us will be rewarding to all, and wish him welcome.

PASCAL seems to be getting more and more play these days. Is it the language of the future? Or is it here now? Our Associate Editor, Terry Dettmann, takes another look at it and draws a comparison between it and Basic, starting on page 40.

Anatomy of the Program is missing this issue. Seems another mag stole our thunder with an identical article just one month prior. It will return in the next issue.

There is an Anatomy of sorts, for those who can't live without it, in our Business offering this issue. R A Shmina starts a two part article covering a personal finance system, including listings and an anatomy.

Phil Pilgrim is back this issue with System/Command, and has a clever way to have two (count'em, 2) Basic programs running in memory at the *same* time! We don't know where Phil gets these ideas, but we like it!

Brad Hoza has figured out how to give Level II owners the "JKL" feature of NEWDOS. He calls it "234", and it works on Level II, 16K. See page 48.

Our resident reviewer (and full-time teacher) Cam Brown, has found a way to get 48K with Disks for free. He tells about it on page 52. And, he has just told us that the Jogathon mentioned in that article grossed over $38,000.00 for the school. Not too shabby for "toy" computers, what?

Jim Crocker does a thing with the Model II in this issue by making PEEK and POKE work. No, not a substitute one, where you have to run a machine language program in first. This is the real thing. The PEEK and POKE are a real part of Basic, and can be used exactly like the Model I did. Read about it in NOTES.

For all of you who have purchased the Radio Shack 779 printer (or the Centronics 779), and now find that it has no resale value, Larry Panattoni has just the thing for you. On page 72, he tells us how to convert that 779 to upper/lower case. Note the captions on the figures in that article, they were printed on his 779. Although the schematics may look forbidding, there is really not that much to it. The additional chip from Centronics costs around $60.00, the rest should be readily available from that proverbial "well stocked junk box".

On page 84, David Bohlke tells us how to make Big Caps on the video screen. We tried it, and it does indeed catch your attention.

Mr Havenhill tells how to pack numerical data into strings for graphics on page 86, and Mr Williams tells how to get control of the RESET button on 88.

Leo Christopherson has a short article on page 89, telling how to get Radio Shack's Scripsit on an Exatron Stringy Floppy wafer (it works for other machine code programs too).

Bob Labenski does a nice job on reading a machine language program out of memory and automatically making a Basic program out of it.

Our friend from the May-Jun 80 issue, Teersaty, is back with more antics and useful information. Bill Wilson (Teersaty's legal guardian) will field questions on systems, see page 92.

That is our offering for the first issue of our third year. We enjoyed putting it together, and hope you enjoy reading it.

Tell them you saw it in 80-U.S., and remember that nice days are made not had.

Mike

introduces

THE TRS-80* MODEL I SYSTEM DESK

Recognizing the need for a completely integrated system desk for the TRS-80 Model I, **Computer Business Systems** introduces an operating console to eliminate the unsightly and confusing maze of cables, power cords, and switches to make the Model I an easier computer to use. Solid ¾″ wood construction throughout with a beautiful walnut-grain laminated finish that resists burns, mars, etc., it blends perfectly into any office or home decor.

$399.00

FEATURING

- Turn-key power up, provides security access and controls power up to all components. The CPU and all its attachments power up together, ready for processing.
- Built in cooling fan keeps disk drives, expansion interface, and CPU keyboard in optimal operating range.
- Built in power outlet strips (10 grounded outlets, UL approved) leave room for complete expansion of the Model I computer. One AC power cord (10 ft external of desk) powers all components.
- Removable top allows access to all cables, power cords, and electrical connections, but keeps them completely hidden from view and inaccessible to tampering.
- Holds the full compliment of TRS-80 Model I components:
  up to 4 mini-disk drives
  expansion interface
  keyboard
  CRT display
  2 cassette player-recorders
  almost any printer
- Built in and recessed "RESET" button extension eliminates fumbling with pens or pencils trying to restart your system, yet does not attach to your TRS-80 in any way.
- Absolutely no modification whatsoever to TRS-80 hardware; your warranty remains intact.
- Forms may be fed thru printer from bottom of desk (ample slots are provided) or rear of printer. If printer is not used, a matching filler is provided for slot in desk top.
- So compact the dimensions are the same as a normal desk: 60″L × 30″W × 29″H.

FOR MORE INFORMATION CALL OR WRITE TODAY

COMPUTER BUSINESS SYSTEMS
329 South Highland Avenue
Lombard, Illinois 60148
(312) 932-1344

*TRS-80 is a trademark of Radio Shack a Division of the Tandy Corporation

★ New Products ★

LEARN TO COMPUTE

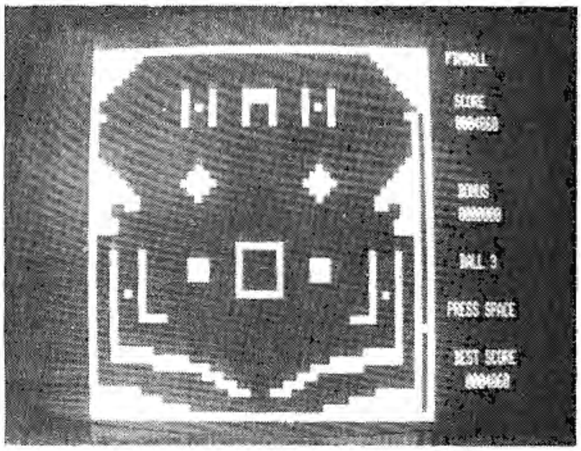
Now available from Radio Shack is TRS-80 Level II Basic, a self-teaching guide for learning to program and use a Level II TRS-80 with no previous computer experience. This 351 page step-by-step manual is written by Bob Albrecht, Don Inman and Ramon Zamora. The authors believe that learning to program can be an enjoyable experience. The material in the book gets more challenging as you move through the chapters. The book is available from participating Radio Shack stores and dealers, priced at $9.95

Cat. No. 62-2061
Nine Dollars and Ninety-Five Cents
Radio Shack
TRS-80
Level II
BASIC
By Bob Albrecht, Don Inman and Ramon Zamora
a self-teaching guide

QUILL DRIVER

Quill Driver is a text processor for the Model I or II TRS-80 with at least one disk drive. QD both enhances capabilities of existing text processors and acts as a stand alone system. A 50+ page manual provedes extensive documentation. Other features include commands for accessing alternate input files, right justification, handling problem formats such as tabulation and hanging indents, various forms of pagination and more. Thirty three commands in all. A program is also included to allow the transposition of PENCIL, BASIC and EDTASM text files to any one of the three file types.

The package also includes a driver which allows toggling between upper and lower/shift upper and lower case, a repeating key and other features. Lower case may be input whether or not user has a lower case mod. Formatted output may be directed to screen, printer or disk. The entire Quill Driver package is only $39.95 on diskette, and is available from The Alternate Source, 1806 Ada St., Lansing, MI 48910



PINBALL

Acorn Software Products Inc., announces the release of PINBALL, a real-time, arcade game for the Level II 16K TRS-80, written in machine language. PINBALL includes flippers, bumpers, rollovers, runs and bonus points. The space bar on the TRS-80 releases the ball at various speeds under player control. Once in play, both the speed and acceleration of the ball depend on the contact with various features on the board, including the mysterious "Bermuda Square". PINBALL is priced at $14.95 on cassette or $20.95 on disk. Acorn Software Products Inc., 634 North Carolina Ave SE, Washington, DC 20003 (202) 544-4259

ZOOM 3.6

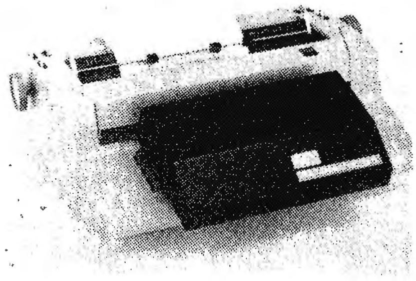
Is an electronic "black-box" that connects in between a Level II TRS-80 (or expansion interface) and a CTR-41 or CTR-80 cassette recorder. With the ready-to-run software supplied with it, tapes can be written and read in a special format at 3600 bits per second. No soldering or modifications are required. Zoom 3.6 is wholly transparent to all XRX mods (and to any other signal processor connected to the cassette port), and to the CLOAD, CSAVE, SYSTEM, and PUNCH functions. All cables can be left plugged in permanently. For further information send a SASE to ZOOM! PO Box 3766, Nashua, NH 03061 (603) 889-0901

NEW GAME SOFTWARE

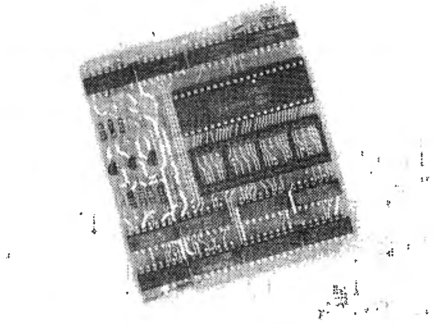
Simulation Software announces the release of two programs for TRS-80's equipped with Level II and at least 16K of RAM. DUNGEON EXPLORER 2.0 is an extensively revised single player game of fantastic combat adventure. A player tries to become a superhero by battling the vicious monsters within the Dungeon of Xanadu. The game features a streamlined game command input routine (using INKEY$), improved combat sequences, additional monsters and mapping graphics. COSMIC TRADER is a multiplayer game of interstellar trade. Up to four people try to amass a fortune by commanding their own star freighter in a quadrant consisting of nine star systems with nine categories of trade goods. Players must negotiate all transactions with alien merchants (the computer). Players must cope with sudden changes in the marketplace and in market prices. The user can adjust the game length to play for just half an hour or for an entire evening. Both programs are on cassette and come with complete instructions for $12.95 plus $1 per order for shipping. Simulation Software, PO Box 1368, Warren, MI 48090 (313) 758-0798

SELECTRIC INTERFACE

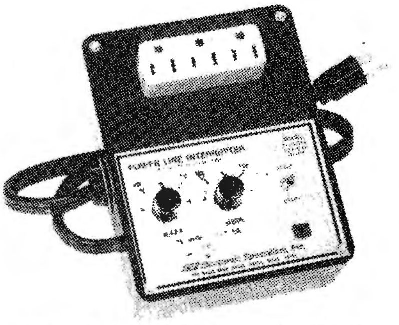
Kogyosha, the largest manufacturer of DC solenoids in Japan, announces production of the KGS-80 Keyboard Actuator. This new peripheral turns an IBM Selectric Typewriter, or its equivalent, into an economical, high quality printer for the TRS-80. In seconds, the KGS-80 is positioned on the keyboard, plugged into the expansion interface, or directly to the CPU using the interface cable, and the Selectric is ready to start printing all Level II print commands. No modification to the. typewriter is necessary, nor does the KGS-80 require any software to operate. Cost is $599.00, without the power supply. For more information contact Mark Nakanishi, Kogyosha USA Office, 179 Riveredge Road, Tenafly, NJ 07670 (201) 569-8769



DOUBLER™



Percom Data Company has announced they have begun production of a double-density disk controller adapter for the TRS-80 Model I computer. Using the DOUBLER™ a TRS-80 computer owner can now store almost four times more data on a five-inch disk - up to 354 formatted Kbytes. The DOUBLER™ adapter plugs into the controller chip socket of the computer expansion interface. No circuit mods are required. Price, for the DOUBLER™, DBLDOS™ - a TRSDOS compatible double-density operating system - and a utility for converting TRSDOS, Percom OS-80™ and other single-density files and programs into double density format, is $219.95. Percom Data Co, 211 N Kirby, Garland, TX 75042 (214) 272-3421



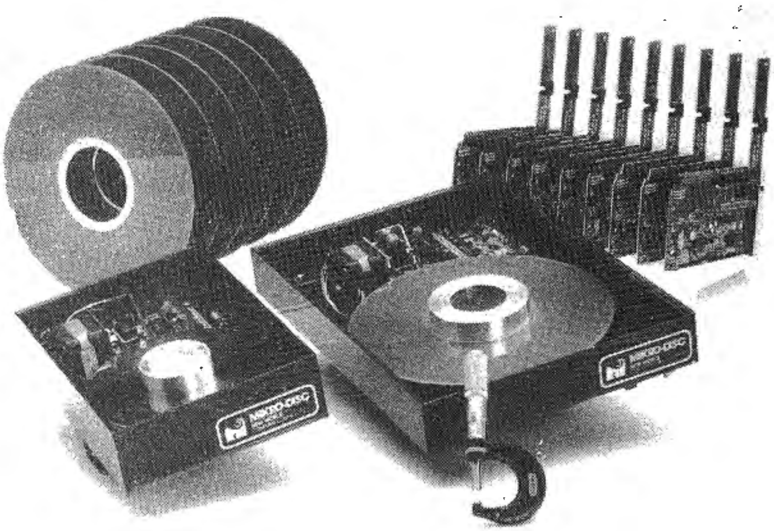
POWER LINE INTERRUPTER

Electronic Specialists announced the introduction of the POWER LINE INTERRUPTER. Should AC line voltage be disrupted or exceed user selectable limits, the device disconnects power from controlled apparatus. Front panel controls provide UNDER/OVER voltage interrupt level selection and Power Reset. Other features include integral Spike/Surge Suppression and response delay to prevent false interrupts. Intended for specialized Micro Computer applications, where equipment is subject to periods of unattended operation, the POWER LINE INTERRUPTER will provide safety and protection for equipment as well as personnel. The unit can accommodate a 15 Amp resistive load or a 10 Amp inductive load. Model PI-15-O/U, Over and Under Voltage - $142.95, Model PI-15-U, Under voltage only, $127.95. Electronic Specialists Inc., 171 South Main St., Natick, MA 01760 (617)655-1532

PMC-80™

Personal Micro Computers, Inc (PMC), is a newly formed subsidiary of an 11 year old firm, Recortec, Inc. Personal Micro Computers, Inc has been created to broaden the Microsette Co product base in the micro computer market. Microsette Co has been selling blank cassettes and program duplication services for the past three years and will continue to do so under the Microsette Co name.

Personal Micro Computers, Inc., introduced the Add-On-80™ product line which contains five peripherals to the TRS-80 Level II computer. Now, through an exclusive marketing agreement with a Hong Kong manufacturer, Personal Micro Computers Inc is offering a software and hardware compatible equivalent of the Radio Shack Model I, Level II, TRS-80™. The new computer has a cassette tape recorder, 16K RAM, Level II Microsoft Basic interpreter in ROM, power supply, computer and keyboard all in one attractive cabinet. The PMC-80™ will display on either a TV monitor or on a standard TV set using a built-in VHF Channel 3 modulator. All peripherals designed for the Radio Shack parallel port will interface to the PMC-80™ 50 pin bus through a 40 pin interface adapter available from PMC Inc. Warranty service for the equipment will be performed by PMC Inc, at it's Mountain View, CA factory. For further information, contact Personal Micro Computers Inc., 475 Ellis St., Mountain View, CA 94043 (415) 968-1604



MIKRO-DISC™

New World Computer Co Inc announced the addition of two new members to the highly successful MIKRO-DISC™ family of quick-access Winchester technology drives: the MINIMIKRO-DISC™ V Series of 5¼" fixed disk drives; and an enhanced double density system identified as the MIKRO-DISC™ VIII-1TF. The V-1TF is a low cost, quick access drive based upon a modified Winchester technology. It has a multiple head assembly that provides 8 read/write heads, 128 tracks per surface, and places the storage capacity of a 5 inch floppy disk drive immediately beneath the heads at all times and accessible within one disk rotation. For additional information contact New World Computer Co Inc., 3176 Pullman St, Suite 120, Costa Mesa, CA 92626 (714) 556-9320

VIDEOTEX

A new two-way information retrieval system for home or office use was unveiled by Radio Shack. Purchasers of the equipment will be able to carry on keyboard question and answer "conversations" with a central computer over telephone lines, and view the result on their color or black-and-white television sets. Called TRS-80 VIDEOTEX, the system will soon be available in several configurations. One option, designed for owners of the estimated 400,000 personal computers now in use, is a software package which converts the computer into a VIDEOTEX device. Also demonstrated and slated for a late October 1980 delivery, was a VIDEOTEX terminal for non-owners of microcomputers. The unit attaches to the telephone line and the antenna terminal of any TV receiver. The user dials a pre-determined number to access the data base, then selects the data desired by pressing specified keyboard numbers and letters.

Answers appear on the TV screen. Some of the types of information expected to be available to VIDEOTEX users through such data bases as H & R Block's CompuServe Information Service include: weather, sports, news, transportation schedules, stocks, commodities, classified ads, newsletters, library data, medical and agricultural information, mail-order ads, electronic mail and electronic banking. Both the TRS-80 VIDEOTEX software package for use with microcomputers, and the information terminal keyboard, will be sold exclusively by Radio Shack Computer Centers and participatir ʒ Radio Shack stores and authorized retail dealers.

CHATTERBOX

The Micromint Inc announces its latest data communications product, the "Chatterbox". It is a unique packaging combination of the presently available COMM-80 I/O interface for the TRS-80 and an acoustic modem. This one box is all that is required to turn even a bare-bones 4K TRS-80 into a full timesharing terminal. The unit, priced at $259.95 is available from The Micromint Inc., 917 Midway, Woodmere, NY 11598 (516) 374-6793

CPYALL

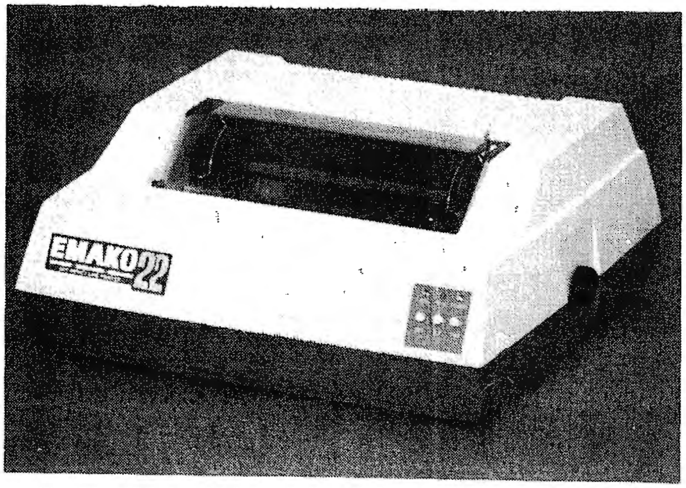
Is a machine language program that will allow you to copy almost any 500 baud tape. Most all TRS-80 owners have experienced the difficulty of loading cassette tapes not made on their own cassette recorder. While patience and experimentation with the volume control will usually lead to success, the process is very time consuming and frustrating. With CPYALL, you need to find that volume setting just once. After the program is read into CPYALL and verified, as many back-up copies as desired can be dumped onto your standard kind of tape. CPYALL displays an ASCII or graphic representation of every byte read in, verified, or punched out. This one program will copy almost any tape: Basic, machine language, data, source (from the Editor/Assembler), programs with special loaders and others. It requires Level II, 4K and up. Priced at $7.00 from Byte Miser Software, 720 West Haven Blvd. Rocky Mount, NC 27801

FLEX-FILE SYSTEM

A new product, the Flex-File Page, provides a better way to file, store and protect your flexible disks. It is a non-glare vinyl page having pockets on each side to house two 8" diskettes plus a center pocket to store standard 8½ X 11" sheets of paper, computer printouts or other documentation. The pages are three hole punched for convenient and economical storage in standard three-ring binders. Flex-File Pages are priced at $8.95 for a package of 10 pages and are available from BIS, Inc., PO Box 969, Brentwood, TN 37027

NEWPATCH

Using other software with Radio Shack's new upper/lower case hardware mod need not be limited by using NEWPATCH, from Lords. Cost is $20.00 provided on your DOS. The PencilC Mod allows use of pencil on the same hardware, and has additional features. Cost is $35.00 provided on customer's original Pencil Cassette. (Not available for VTOS). Lords, PO Box 99, Port Angeles, WA 98362 (206) 457-3064



LTM
INCORPORATED

TO ORDER CALL TOLL FREE
1-800-321-2037
IN OHIO CALL COLLECT (216) 566-9130

master charge
VISA

TRS 80 and Radio Shack are registered
trademarks of Tandy Corp

THE CHESTERFIELD BUILDING • 1801 EAST 12TH STREET, SUITE 222 • CLEVELAND, OHIO 44115

★NEW★ MTC AIDS-III* ★NEW★

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

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

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

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge.

- COMPATIBLE with AIDS-II data files and AIDS subsystems.
- Move up from AIDS-II and EXPAND to 20 field capability WITHOUT REENTERING DATA
- AIDS-II (Model I or II) owners may UPGRADE FOR ONLY $25.00.

*WARNING! This program is written in BASIC and can be listed in the normal manner Modification of program code is NOT RECOMMENDED due to its extreme complexity

NEW the MAGIC WAND™ NEW

$349.95

the most powerful, most flexible, most reliable, most useable word processing software available for a CP/M® based TRS-80® model II.

MAGIC WAND™ can do more work in less time with high quality than any other product you can buy.

The command structure is simple, logical and complete. The programs are crash-proof and completely reliable.

The system is supported by what users say is the best user's manual ever produced for microcomputer software.

FEATURES

- Full screen text editing
- Full text formatting commands
- Merging with external data files
- Up to 128 variables
- Conditional commands
- True proportional spacing

QUOTES FROM THE June, 1980 Microcomputing article "Super Work Processors" by Rod Hallen

"Of all the word processors I have used (and that includes a dozen or more), the Magic Wand is the most versatile. The Wand has almost all of the features of other processors, plus many new ones of its own. It measures up to even the word-processing software running on the largest mainframe computers."

. . ."Magic Wand is an outstanding example of the new levels of software that are being written for the small businessman, although I can't imagine a business of any size that couldn't use software of this quality."

MAGIC WAND - will also operate on Oasis based systems
- will operate on 16k but we recommend 32k for adequate operating memory
- is available on 5¼ and 8 diskettes

MAGIC WAND is a copyrighted program by **Small Business Applications Inc., TRS-80 is a** registered trademark of Tandy Corp., CP/M is a registered trademark of Digital Research Corp.

MTC AIDS - II

Ailing information? Doctor it up with AIDS-II. This Automated Information Directory System offers twelve user-defined fields with full feature editing when adding or changing records. Selective Loading, Updating, Deleting, Printing and Saving records may be accomplished using any of six relational comparisons. Also features machine code assisted sorting (200 records in about 5 seconds) by any combination of fields, and much more! Unique "windowing" capability allows directories of unlimited size. Window size is typically 200 or more records in 32K. Can be used for mailing lists, client reference reporting, appointment "calendars", inventory records and other information systems. Easy to use. Defining a system takes about a minute. MAPS-I (MTC AIDS PRINT SUBSYSTEM) is included at no charge. MAPS features full AIDS-II selection capabilities, prints user-specified fields down the page, produces user-specified columnar report formats with automatically generated column headings and paging, and allows user-defined print formats for custom forms, labels, etc. Add subsystems for additional capabilities. May be upgraded to AIDS-III when required.

MTC AIDS-II	**$ 49.95**
For Model II	**$ 79.95**

MTC CALCS $24.95 **AIDS** **For Model II $39.95**

CALCULATION SUBSYSTEM (CALCS)

MTC's most popular AIDS subsystem. Use for report generation involving basic manipulation of numeric data. Prints user-specified fields in titled, columnar report format, automatically generating column headings, paging and (optionally) indentation.

Apparat, Inc. introduces

NEWDOS/80

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

NEWDOS/80	**$ 149.95**

CALL REGARDING OUR NEWDOS+ UPGRADE PRICING.

★ PRODUCT PREVIEW ★

General Business System for Model II

This product will be a full-feature, professional-grade business system. with fully integrated General Ledger, Accounts Receivable and Accounts Payable. A Payroll subsystem will be ad-[illegible]

MAKES EVERY BYTE COUNT

IN YOUR TRS-80® MODEL I OR MODEL II DISK SYSTEM

PROGRAMMING TOOLS

SAVE

Any 3, $49.95

For Model II........... $74.95

TDAM $19.95
For Model II $29.95

Includes MTC QUE Card!

Having trouble with RANDOM FILES? With MTC's Table-Driven Access Method (TDAM) you'll never fret over FIELDing again. No knowledge of random access files is required. Insert the TDAM "interpreter" into any BASIC program and type in a few DATA statements describing the information in your files. TDAM does the rest! Reads and writes fields and records of any type (even compresses a DATE field into 3 bytes!). Features automatic file buffer allocation/deallocation, memory buffering, sub-record blocking/deblocking, and handles up to 255 fields per record. Super fast and super simple! Complete with TDAM interpreter, instructions and demo program. Requires programming experience.

DIVERGE $19.95
For Model II $29.95

Compares two BASIC program files, showing the differences between them. Identifies & lists lines which have been inserted, deleted, & replaced. Use for version control.

REBUILD $19.95
For Model II $29.95

Reorganize programs for adding program code, faster execution, readability. Much more than simple renumbering. Rearrange groups of statements within a program - automatically updates references to line numbers. Use with SUPERSEDE and MINGLE for maximum effect.

SIFTER..................... $19.95
For Model II................. $29.95

Twelve in-memory high-speed sorts for use in any BASIC program: stable, non-stable, with/without tags, for numeric or string data. Random File Sort included. Some sorts written in machine code. Includes sort subroutines, demo programs and instructions. Relocate as needed with REBUILD. Requires programming experience.

SHRINK.................... $19.95
For Model II................. $29.95

Makes Every Byte Count! Make programs smaller and faster! Combines lines & removes unnecessary code including remarks, without altering program operation. Typically reduces program size 25% to 40%.

SUPERSEDE $19.95
For Model II................. $29.95

A "must have" for the professional programmer or the serious amateur. Probably one of the greatest time-savers available. Write programs in shorthand - change variable names - generate program documentation - use with REBUILD and MINGLE to build new programs from old ones.

MINGLE-II.................. $19.95
For Model II................. $29.95

Merge up to 14 files (Program or Data) into a single file. Data files may be merged in ascending or descending sequence with the ordering based on a user-specified comparison field. A very handy utility for consolidating data files.

Single sided, Single density, Soft-sectored

DISKETTES

Verbatim 5¼-inch

SEPTEMBER SPECIAL*

$23.95

Box of 10

Quantity 10 Boxes $23.50

Hard-sectored (10-hole), Box of 10....... $26.95

8-inch FLOPPIES

Single-density, Box of 10................ $29.95

Double-density, Box of 10............... $39.95

PLASTIC LIBRARY CASES

5¼-inch or 8-inch diskette case.......... $3.00

FACTORY FRESH, ABSOLUTELY FIRST QUALITY
Minimum order 1 box, NO order limit!

*Price in effect Aug. 1 through Aug. 31, 1980

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

NEWDOS +

$69.95 by Apparat

40 TRACK VERSION $79.95

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

MTC QUE Card only $1.50

Let Your TRS-80® Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

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

REMDISK-1 .... $29.95

Let your TRS-80® Teach You

ASSEMBLY LANGUAGE

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

REMASSEM-1 .... $69.95

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

"TRS-80 DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

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

TRS-80 DISK $19.95

All products guaranteed for replacement only. Prices, Specifications & Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

QUANTITY DISCOUNT INQUIRIES INVITED

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

- Add $2.50 for shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

TO ORDER CALL TOLL FREE
1-800-321-2037
IN OHIO CALL COLLECT (216) 566-9130

LTM
INCORPORATED

master charge

VISA

TRS-80 and Radio Shack are registered trademarks of Tandy Corp

THE CHESTERFIELD BUILDING • 1801 EAST 12TH STREET, SUITE 222 • CLEVELAND, OHIO 44115

THE LEAST EXPENSIVE PROGRAMS YOU CAN BUY.

Up to 77 high-quality programs for TRS-80, only $8!

THE $15.50 COMPILER:

TINY PASCAL FOR TRS-80

At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16 K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:
● tiny Pascal compiler ● complete text editor for writing your programs ● complete tiny Pascal monitor ● sample Pascal programs ● user's manual (TRS-80 Computing issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36 K RAM and a solid knowledge of programming).

With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

Here is a partial list of People's Pascal features: recursive procedure/functions ● for (loop) ● case if/then/else ● one-dimensional arrays ● write ● read constant ● repeat/until (loop) ● "peek & poke" ● plot (graphics for TRS-80)

DEALER INQUERIES INVITED

People's Pascal 1 (tape 3) is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles P codes more slowly and is harder to use than Pascal 2, but its P codes can be translated into Z80 native code and saved as System tapes. Pascal 2 requires that Pascal be resident at run time—Pascal 1 does not. Other People's Software tapes $8.

TAPE 1 LEVEL 2

Mortgage calculations, Dow Jones Industrial, cash flow, inventory-change, California income tax, journal ledger (8K), loan amortization, perpetual calendar, bio rhythm, payroll, diet planning, speed reading, touch typing, sales receipt tally, decision maker, mail addressing, straight depreciation, double-declining depreciation, and revolving charge account.

Also, math problems, queen, Star Trek I, number guessing, wheel of fortune, World War II bomber, rock-scissors-paper, seek, Star Trek II, Red Baron, mini-Trek, strategy, pilot, battleship, "On A Snowy Evening", mastermind, tic-tac-toe, grand prix auto race, capitals, etch sketch, hangman. Total programs: 34; Level 1 version available: 24 programs. $8. $8.

TAPE 2 Some Common Basic Programs (lev. 2)

Fully documented in **Some Common Basic Programs** by Lon Poole & Mary Borchers (Osborne & Associates, 630 Bancroft way, Berkeley CA 94710—or from CIE—$12.50 postpaid from CIE, via UPS, CA residents add tax (to $13 25)):

Investment, future value regular deposits,; regular deposits; regular withdrawals, initial, minim (for withdrawals); nominal interest, effective & earned-interest; depreciation rate, amount depreciation; salvage value; discount com'l paper; loan principal, regular & last payment, remaining balance, term-loan; mortgage amortization; greatest common denom.- integer prime factors; polygon area; triangle parts; analysis, operations two vectors; radian-degree., degree-radian conversion; coordinate, polar equation, functions plot; linear, curvilinear interpolation; Simpson's & trapezoidal rules, Gaussian quadrature integration; derivative.

Side 2—quadratic equation, polynomial (Newton) & half-interval-search roots; trig polynomial; simultaneous equations; linear programming; matrix addition, subtraction; scalar multiplication, inversion; permutations & combinations; Mann-Whitney U test; mean, variance, standard deviation; geometric mean & deviation; binomial, Poisson, normal, Chi-square distribution; Chi-sq., student's T-distribution test; F-distribution; linear correlation coefficient; linear, multiple-linear, Nth order, geometric, exponental regression; system reliability; future projections; Federal withholding taxes; tax depreciation schedule; check writer; recipe cost; map check; day of week; days between two dates; anglo to metric; alphebitize $8

TAPE 3 People's Pascal Development System 1

Pascal 1 compiler-program development system. $15.50.

TAPE 4 LEVEL 1

Election returns, business percentage, ups and downs of business, index, inventory control, sales receipt tally, gas mileage, driving distance, mixed monthly sales report, payroll, annual earnings, speech recording aid, and double-declining depreciation.

Also, math problems, cash register, chase, snoopy, commander-in-chief, Christmas graphic, air raid, balance scale, stock market, tic-tac-toe and **On A Snowy Evening.** $8.

TAPE 5 LEVEL 2

Memory test, mortgage payments, tension breaker, lineprinter-screen & vice-versa utilities, Federal income tax, election returns, business percentage, vacation planner, car pool(disk), diet planning 2, mailing list(disk) and first aid.

Also spelling bee, Star Trek 3, mind bender, tachistoscope, chase, common factor, klingon capture, spelling practice, Hamurabi, animals, Snoopy, cryptogram, starship, ants, **Yesterday**, and Pilot(disk). Pilot is the language of computer-aided instruction (CAI) .

TAPE 6 People's Pascal 2

Pascal 2 compiler-program developemnt system. $23.50.

TAPE 7 LEVEL 2

Disassembler, Pilot, roster, dropout, memory loader, memory sort, inventory control, graph, land surveying, mixed monthly sales report, shopping list, diet planning 3, loan progress chart, hex-decimal conversion.

Also Star Trek 4, states and capitals, battleships 2, spelling practice 2, number guessing, hangman 2, snark, slot machine, cipher, target, surround, adder, termites, lunar lander, multiplication exercise, five-in-a-row, Bastem, and write. A number after a program indicates there are other similar People's Software programs. Pilot is the same as the disk pilot on tape 5, except runs on 16K tape systems. $8.

—CA Residents Add 6 per cent TAX ($8.45, $16.40 & 24.88). Quoted Prices INCLUDE $.50 Shipping—

COMPUTER INFORMATION EXCHANGE
Box 159, San Luis Rey CA 92068 (714) 757-4849

Feature

Adding New Functions and Commands to Level II Basic

John Hind
Raleigh, NC

If you are a Level II owner you have probably found one of the several limitations of the Basic interpreter, and have at least once wished for some new function or command which would make programming a bit simpler. You can make your wishes come true with an understanding of the ROM and a few lines of Z-80 code.

In this article, we will explore some of the more important conventions and entry points of the ROM, and show how the user can use the Disk Basic RAM exits to add new and personalized language features. Two examples will be given here; one which implements a set of integer functions with a non-standard syntax, and one which implements a cassette tape merge command.

The key to the ROM software can be found in the power-on initialization sequence which starts at location 0, and which tailors the interpreter by initializing the reserved area of RAM between 4080H and 41E5H. When the computer is operated without a disk (or BASIC2 is called from DOS READY), this sequence stores a number of RETurn and JUMP to 12DH (L3 error) instructions, thus ef'ectively 'turning off' the DISK BASIC functions. However, if the user understands the linking conventions, he or she can add their own functions by making use of these vectors.

To gain an understanding of the linkages, one must decode the existing language elements, which assumes you know where to look. My search was long and quite frustrating, ending by accident in success. The major problem with tracing out the ROM software was following the non-structured programming style mandated by the 12K memory limit. This style includes such practices as jumping into the middle of an instruction to synthesize another one.

The ROM can effectively be broken down into 5 segments. The initialization routines are only called on power-up. These are the routines that load the Device Control Blocks, set MEMORY SIZE, and generally get things ready for BASIC. Some sections of the initialization are used during the execution of certain commands, such as NEW, CLEAR, and DIM.

The second segment is the INPUT/COMMAND monitor. This is the segment that issues the READY prompt, takes keyboard input, compresses the BASIC text, and stores it in memory.

The third segment is the EXECUTION monitor. This section is entered by the RUN and CONT commands, and is responsible for keeping track of which line of BASIC text is being executed, and CALLing the appropriate keyword execution routines, along with general housekeeping chores. The execution monitor also includes the error invocation routines.

The fourth segment is the actual subroutines to execute the various BASIC statements, and includes special sections to evaluate mathematical expressions.

Finally, segment five consists of the various subroutines necessary for execution. These include the special RST's, keyboard scans, video I/O, etc.

When a line of BASIC text is entered by the user (signified as a line of text vs. a command line by having a line number), it is compressed by the INPUT/COMMAND monitor. Every BASIC keyword (see p. A/15 in the old Level II manual, or pages E/1-E/2 in the new one.) is compressed into a one-byte code before being stored in the RAM text area. This step not only saves memory, but also speeds execution by allowing speedy decoding of these keywords at RUN time.

These keyword codes are cleverly arranged in such a manner that their value may be used to decode the address of the corresponding execution routine from one of two tables. The first of these tables begins at address 1822H, and represents the addresses of the routines for the keyword codes 80H-BBH. The relative position within the table may be arrived at by the formula (CODE#-80H)*2+1822H. Throughout execution, the HL register pair is used as a sort of "program counter", so before control is passed to the execution routine, the HL is advanced to the next non-blank character in the text.

The second set of keywords are those with code numbers BCH to D6H. These are functions that would be handled from within other commands. For example, TAB and USING are handled by the PRINT command. TO is taken care of by FOR, and most of the rest are handled by the expression evaluator starting at 2337H and worming its way through much of the ROM. A rather lengthy comparison table may be seen starting with address 24AEH. It is because of this rather convoluted method that such strange statements as "A=B=C" will result in the variable A containing the results of the question "does B=C?" (A=-1 if B=C, and 0 if not) which can later be used in an expression such as "IF A THEN 50".

The third set of keywords are those that must be called from within another routine. The most common requirement for these codes (D7H-FAH) is they be on the right side of an equals sign, or similar syntax. CINT, for example, would cause a syntax error, while A=CINT(A) would not. In this case, the CINT routine would be called by the mathematical evaluation routine, not the execution monitor.

The jump table for these secondary commands resides at address 1608H, and the relative address within the table is calculated using the formula (CODE#-D7H)*2+1608H.

After completion of the requested function (which may require several other subroutines to be invoked), the command executor may choose one of three return paths;

1) Control may be passed back to the execution monitor with the mock "PC" (HL) updated to the next statement.

2) The INPUT/COMMAND monitor may be invoked (as in CLOAD, NEW, and STOP).

3) The error routine may be invoked.

Control may be returned to the execution monitor by a simple return instruction.

The INPUT/COMMAND monitor may be invoked by a jump to address 1A19H or a number of similar jumps. The INPUT/COMMAND monitor automatically checks to see if it was called as a result of a syntax error, and the EDITor is called if it was.

The error routine may be invoked by loading the E register with the appropriate error code (the TRUE error code, not the "ERR/2+1" code), and executing a jump to address 19A2H. The error routine automatically checks to see if an error trap ("ON ERROR GOTO...") is present, and forces a GOTO that line if so.

If control is returned to the EXECUTION monitor it will check to see if it is a colon (:); in which case execution will proceed as though nothing unusual had happened which, in fact, is the case. If a colon is not found, a zero (which signifies the end of the line of text) is checked for. If one is found, several pointers are updated to the value of the new line number. Meanwhile, these 2 characters are checked to see if they are both zeros. This being the 'end of text' flag, if they are, control is passed to the INPUT/COMMAND monitor. If the next line is not the end of text flag, the monitor checks to see if trace is on (signified by the value AFH in address 411BH), and displays the line number to the screen if so.

After all this, the next text character is read, and compared to see if it is a valid command (greater than 80H). If it is not, it is checked to see if it is a valid variable name (by the LET command), and execution continues with the mathematical expression evaluator.

If it is a valid command, it is checked to see if it is within the range of 80H-BBH. If not, it is checked to see if it is the MID$ command, which can be modified by Disk Basic, and that is taken care of. If not, a syntax error is generated.

If, after all this, the command syntax is valid, and we have passed all the tests, the formula mentioned above is executed, some further housekeeping is done, the return point (1D1EH) is PUSHed onto the top of the stack, and the execution routine is called. Simple, right?

Let us now look at some of the 'common subroutines' mentioned earlier. Some of the most important of these make use of the Z-80's restart (RST) instruction to reduce the code requirements for the most frequently used routines. In Level II, these RST instructions are used for the following:

RST 8 is used to check for the syntax of the next character in the text. On entry to this routine, the HL points to the text character to be checked, and the stack pointer (SP) points to the character to be compared against. An example of its use would be:

```
LD      HL,TEXT
RST     8
DEFB    28H
; execution resumes here
```

In this case, we want to make sure that the HL points to a 28H. If the comparison is not valid (HL) <> (SP), a syntax error is generated. If the comparison is valid, the contents of the (SP) are incremented (to the correct return address), and the RST 16 is executed.

RST 16 (10H) is used to advance the "PC" (actually the HL register) to the next non-blank text character. In other words, it skips spaces in the text.

RST 24 (18H) is used to logically compare two 16-bit unsigned numbers. The HL is compared to the DE, and the condition bits are set accordingly.

RST 32 (20H) sets the condition bits according to which type of variable is currently being worked upon by the interpreter. The contents of a special "expression type flag" located at address 40AFH are compared against 03H. The flag contains a one-byte designator which describes the variable type, by representing the number of bytes required to hold the variable. These values, and the results of a RST 32 are:

		Condition Bits (F Register)		
Integer	**2 bytes**	**M**	**C**	**NZ**
String	**3 bytes**	**P**	**C**	**Z**
Single Precision	**4 bytes**	**P**	**C**	**NZ**
Double Precision	**8 bytes**	**P**	**NC**	**NZ**

The manner in which the RST's get to their final destination is via a rather roundabout tour of the ROM and RAM. Since the RST destinations are placed 8 bytes apart, it is difficult to get any meaningful code between them. Therefore, Level II uses the RST's as vectors, which in turn lead to other vectors in RAM which can be changed by the user. The RST vectors and their normal values are:

RST 8	**4000/1C96**
RST 10	**4003/1D78**
RST 18	**4006/1C90**
RST 20	**4009/25D9**

The remaining RST vectors normally contain RET instructions. Their Disk Basic values are:

RST 28	**400C/4BA2**
RST 30	**400F/44B4**
RST 38	**4012/4518**

By looking at the appropriate addresses in ROM and RAM, we can see that at address 8, there is a JUMP to address 4000H, and address 4000H contains a JUMP to address 1C96H. It is worth knowing that the RST 28 is used quite heavily by DOS in the normal carrying on of its business. It is also handy to know that the RST 38 is the forced instruction during a MODE 1 maskable interrupt, which is what DOS uses for its heartbeat etc.

Armed with this new found knowledge, one can use one of the available Z-80 disassemblers (such as Small System Software's RSM, or Apparat's Disassem) to decode the logic of the individual ROM routines. Figure 1 is a disassembled listing of the ROM resident portion of the USR command. We will explore its logic to gain an understanding of how the execution routine works.

The first instruction is an example of how existing language features are expanded when Disk Basic (or Level III, etc.) are present. It is a CALL to the reserved RAM area set aside for future expansion by Disk Basic, and is initialized to a RET instruction by the Level II power-on sequence.

The second instruction is the RST 16, which finds the first parameter of the USR argument list (this is the value that will be passed to the USR routine if asked for). The third instruction invokes one of the special subroutines. In this case, it is the parenthesised expression evaluator. This routine will take a list of arguments within parenthesis and whittle away at them until what's left is a pure number. That routine will save the results of its computations in a special area of RAM reserved for this reason, 411DH-4123H.

After the argument is evaluated, a PUSH is executed to preserve the current "PC". Then 0890H is PUSHed onto the top of the stack to serve as the return address for the user's machine language routine. Then the evaluated argument's type flag is compared with that for a string, which requires special handling (CALL Z,29DAH). Finally, the stack is restored (POP AF), the DE and HL registers are exchanged, the USR address is retrieved from address 408EH (where, presumably, it was POKED before), and a JP (HL) is executed, passing control to your routine. (At power-up, this address is loaded with the address of the Illegal Function Call routine to prevent the whole world from going ape should a USR be called without first POKEing (or defining a USR) the address.

When the USR program wishes to get the result of the argument, a CALL to address 0A7FH is performed, which is the entry point of the CINT executor! The CINT routine tests the contents of the area 411D-4123 to see if it is an integer. If it is not, the number is converted to an integer. In either case, the result is stored in address 4121H, and in the HL register as a 16-bit signed integer.

When the USR wishes to give a value back to BASIC, a jump to address 0A9AH is executed, which stores the value in the HL in address 4121H and executes a RET instruction. If the user was careful in his stack usage, the RET will end up at address 0890H, which takes the value stored at 4121 and places it in the variable requested.

For my test program, I wished to recieve a string valued parameter and to decode my own syntax. For this purpose I chose to use the '&' vector, 4194H. After a few test programs, I observed that when a string expression was evaluated by the evaluator, location 4121H pointed to a string describer whose first byte told me the length of the string, and the second and third bytes (in standard Z-80 reverse format) told me the starting address of the actual string.

Figure 3 is a listing of my test program called AEXIT, which implements, in 232 bytes, the following functions:

- **Hex Constants**
- **Convert Hex String Value**
- **Load Word From Memory**
- **Rotate Word Left by Constant or Variable Count**
- **Exclusive or Word With Constant or Variable Mask**
- **Exchange Word Value at Constant or Variable Address**

The advantages of the machine language approach became evident after writing a few of these functions as BASIC subroutines. There can be a hundredfold increase in speed and a significant reduction in memory requirements by using this technique in a large software project without losing the advantages of a high-level language interpreter. The key is to choose a personal set of functions or commands which fit your particular needs and create an OBJECT tape which will be loaded as part of your normal power-up sequence.

There are times when it is desirable to have the ability to decode a decimal number into its hexadecimal equivelant. The common subroutine at 1E4AH will take an ASCII representation of a decimal number and will change it to its HEX value, and return that value as an unsigned 16-bit number in the DE. The limits are a positive number from 0 to 65529.

Here is a partial list of the uses of the RESERVED areas of RAM. It should by no means be considered a complete listing, but should be used as a starting point in a search of the ROM software.

LOCATION (HEX)	PROBABLE FUNCTION
408E	Entry address called by USR function.
4096	Used to build an "OUT (n),A" instruction.
4093	Used to build an "IN (n),A" instruction.
409A	Error code save area
409C	Output control switch, 0 means output to video, pos outputs to printer, neg outputs to cassette.
40A0	Pointer to bottom of string space.
40A2	Line number currently being executed.
40A4	Pointer to beginning of BASIC text.
40A7	Pointer to I/O buffer.
40AF	Variable type flag.
40B1	Pointer to highest usable memory address (protected memory follows)
40D0	Pointer to top of string space.
40E6	Points to statement currently being executed.
40F9	Pointer to end of BASIC text and start of simple variables.
40FB	Pointer to end of simple variables and start of array variables.
40FD	Pointer to end of array variables and start of free memory.
40DF	SYSTEM command autostart address.

THE FOLLOWING ARE JUMP VECTORS FOR THE LIII COMMANDS

COMMAND	ADDRESS
CVI	4152
FN	4155
CVS	4158
DEF	415B
CVD	415E
EOF	4161
LOC	4164
LOF	4167
MKI$	416A
MKS$	416D
MKD$	4170
CMD	4173
TIME$	4176
OPEN	4179
FIELD	417C
GET	417F
PUT	4182
CLOSE	4185
LOAD	4188
MERGE	418B
NAME	418E
KILL	4191
&	4194
LSET	4197
RSET	419A
INSTR	419D
SAVE	41A0
LINE	41A3
(used by INPUT)	41A6
(used by USR)	41A9

THERE ARE ALSO THE FOLLOWING KNOWN RAM CALLS

41AC	Called just before 'READY' is issued.
41AF	Operator keyboard input.
41B2	Called after compression of BASIC text.
41B5	MID$, if on left side of =
41E2	Called by SYSTEM command.
41C7	Called by RUN command.
41DF	Called by LIST and LLIST.
41CF	Called by PRINT.

During my search through the ROM, I have discovered some other interesting information. Perhaps the most frequently asked question is: "What is the format of an OBJECT tape?". The answer lies in the SYSTEM executor located at address 02B2H. An object file is actually one large continuous record on the cassette tape which consists of a header block, from one to n text blocks, and a trailer block which contains the transfer (autostart) address:

HEADER BLOCK	**126 Bytes of zeros, and an A5 sync byte + one byte character "U" + 6 byte name field, padded with blanks.**
TEXT BLOCK	**one byte character "X" + one byte length of block + 2 byte storage address of block + one to 256 bytes of text + one byte checksum**
TRAILER BLOCK	**one byte character "X" + two byte transfer address**

The checksum character is computed by taking the low order storage address byte and successively adding each text character to that value, ignoring any overflow. Each time that a new text block is found, the byte at 3C3FH is EXCLUSIVE OR'd with 0AH, which causes the '*' to flash. This may be accomplished by CALLing 022CH. When the trailer block is found, the autostart address is stored at address 40FDH, and execution may be started at that address by entering the "/(ENTER)" at the second "*?".

The SYSTEM command, like the USR function, can be used as a vector by appending the exit routine. The first instruction of the command execution routine is a CALL to location 41E2H, which is initialized on power-up to a RET instruction. If one of the OBJECT blocks changes this address to a jump to a starting routine, the program will execute as soon as it's finished loading, with no further operator input.

An examination of the PRINT command at location 206FH and its associated subroutines yield some interesting information. There is theoretically no limitation to the length or content of the cassette tape record which this command can create as is implied in the reference manual. This means that one can write a SYSTEM tape file, or in fact any format file with a single BASIC program PRINT statement.

Many of you have experienced the problem of having to retype the same subroutine or block of text into BASIC programs. Some of you have also wished for a means of automating the generation of certain sequences of BASIC statements. To provide a solution to these problems we will use the CLOAD command as a model for a new command which provides the cassette tape merge function.

The ROM version of the CLOAD command calls the NEW command as a subroutine which effectively destroys the text in memory. It next loads the address of the program buffer and enters the tape load sequence which loads RAM until 3 successive zeros are encountered, which signals the end of text. Having loaded the program buffer, it then changes the storage pointers within the text so that programs are fully transportable from Disk BASIC to Level II and vice versa. Studying this section also explains how such programs as TSHORT and Level III can reside in RAM BEFORE the BASIC text (In fact, you can put your routines there too, but that's another story).

The program text buffer is located by the pointer at 40A4H and ends at a location described by the pointer at 40F9H. The buffer is filled by a number of line blocks organized in a forward-chained list in ascending order of line numbers. The last block in the list contains only a pair of zeros, which is used at RUN time to signal the end of the program (the END command is a forced ending, but isn't required). Each line of BASIC text has the following format:

- A two byte pointer to the first byte of the next line of text, in standard Zilog (low then high) format.
- A two byte line number in HEX (also in Zilog format).
- 1 to 240 bytes of text (although the LII manual alludes to 256 as the max text line length, anything over 240 characters long causes some truly wierd things to happen. Also, the keyboard input routine won't let you input more than 240 characters.)
- A zero byte indicating the end of the line.

The format of a CSAVE'd tape is a simple type identifier (3 consecutive D3H characters), a one-byte filename, and the contents of the text buffer.

Figure 4 is a listing of the MERGE command. Notice that it does not call the NEW processor and loads the location of the last line block into the current start of text pointer before going to the tape load routine. This action means that the text on the tape will be appended onto the end of the text currently in RAM. One thing to remember, though. Since the line number search routine which starts at address 1B2CH only looks until it finds either the line number it is searching for or one with a higher line number, you must be sure that the line numbers of the program you are MERGEing are higher than any others already in RAM. Otherwise, they probably won't get found when you try to do your GOSUB.

If you don't have a copy of EDTASM or T-BUG you will want to copy the BASIC program shown in Figure 5 which POKEs the values of the merge program in RAM.

By using the techniques mentioned above, you could write a SYSTEM format tape and/or create any other sequence of BASIC program text statements using the compressed codes explained earlier in this article (see pages E/1-E/2 of the new Level II manual, or page 40 of the Sept-Oct 1979 issue of the Journal for a complete listing of the codes). One interesting possible application of this technique would be to generate a macro language for Level II that would incorporate a modifiable subroutine library capability. ●

```
27FE:  CD A9 41        CALL 41A9
2801:  D7              RST  16
2802:  CD 2C 25        CALL 252C
2805:  E5              PUSH HL
2806:  21 90 08        LD   HL,0890
2809:  E5              PUSH HL
280A:  3A AF 40        LD   A,(40AF)
280D:  F5              PUSH AF
280E:  FE 03           CP   03
2810:  CC DA 29        CALL Z,29DA
2813:  F1              POP  AF
2814:  EB              EX   DE,HL
2815:  2A 8E 40        LD   HL,(408E)
2818:  E9              JP   (HL)
```

Figure 1

```
CINT FUNCTION
(GET USR PARM ENTRY)

0A7F:  E7              RST  32
0A80:  2A 21 41        LD   HL,(4121)
0A83:  F8              RET  M
0A84:  CA F6 0A        JP   Z,0AF6
0A87:  D4 B9 0A        CALL NC,0AB9
0A8A:  21 B2 07        LD   HL,07B2
0A8D:  E5              PUSH HL
0A8E:  3A 24 41        LD   A,(4124)
0A91:  FE 90           CP   90
0A93:  30 0E           JR   NC,0AA3
0A95:  CD FB 0A        CALL 0AFB
0A98:  EB              EX   DE,HL
0A99:  D1              POP  DE

(RETURN USR VALUE
  ENTRY POINT)

0A9A:  22 21 41        LD   (4121),HL
0A9D:  3E 02           LD   A,02
0A9F:  32 AF 40        LD   (40AF),A
0AA2:  C9              RET
```

Figure 2

```
1Ø ' PROGRAM:  STMERGE    "STORE MERGE COMMAND"
2Ø ' BY     :    JOHN R. HIND
3Ø '             41ØØ PICKWICK DR.
4Ø '             RALEIGH , N.C. 27612
5Ø ' FUNCTION:
6Ø '        THIS BASIC PROGRAM WILL STORE THE OBJECT CODE
7Ø '        OF THE MERGE COMMAND INTO THE TRS-8Ø LEVEL II
8Ø '        MEMORY
```

Figure 5

```
90 ' NOTES:
100 '       THE DATA STATEMENTS OF THIS PROGRAM WHERE
110 '       GENERATED BY ANOTHER BASIC PROGRAM AND THEN
120 '       COMBINED BY USE OF THE MERGE COMMAND
130 '
140 DEFINT I-Z
150 PRINT " INSTALLING THE MERGE COMMAND"
160 READ I,J
170 IF I=0 THEN PRINT "FINISHED":END
180 POKE I,J
190 GOTO 160
200 '
210 DATA 32470, 205, 32471, 147, 32472, 2, 32473, 43,
     32474, 215, 32475, 62, 32476, 0, 32477, 40, 32478, 7,
     32479, 205, 32480, 55, 32481, 35, 32482, 205, 32483, 19,
     32484, 42, 32485, 26
220 DATA 32486, 95, 32487, 175, 32488, 87, 32489, 6, 32490, 3,
     32491, 205, 32492, 53, 32493, 2, 32494, 214, 32495, 211,
     32496, 32, 32497, 247, 32498, 16, 32499, 247, 32500, 205,
     32501, 53
230 DATA 32502, 2, 32503, 28, 32504, 29, 32505, 40, 32506, 3,
     32507, 187, 32508, 32, 32509, 8, 32510, 42, 32511, 249,
     32512, 64, 32513, 43, 32514, 43, 32515, 195, 32516, 95,
     32517, 44
240 DATA 32518, 50, 32519, 62, 32520, 60, 32521, 6, 32522, 3,
     32523, 205, 32524, 53, 32525, 2, 32526, 183, 32527, 32,
     32528, 248, 32529, 16, 32530, 248, 32531, 205, 32532, 150,
     32533, 2
250 DATA 32534, 24, 32535, 209
260 DATA 16779, 195, 16780, 214, 16781, 126
270 ' THE NEXT DATA STATEMENT SIGNALS THE END OF STORE BY USE
      OF AN ADDRESS VALUE OF ZERO
280 DATA 0,0
```

Figure 4

```
                00100 ;AUTHOR :         JOHN R. HIND
                00110 ;                 4100 PICKWICK DR.
                00120 ;                 RALEIGH, N.C.    27612
                00130 ;
                00140 ;NAME :           MERGE
                00150 ;FUNCTION :       PROVIDES A LEVEL II BASIC COMMAND WHICH
                00160 ;                 WILL MERGE CSAVED CASSETTE TAPE FILES
                00170 ;OPERATION :
                00180 ;         THE MERGE COMMAND APPENDS A FILE TO THE END OF
                00190 ;         THE CURRENT LEVEL II BASIC PROGRAM TEXT BUFFER.
                00200 ;         THIS MEANS THAT LINE NUMBERS IN THE APPENDED
                00210 ;         FILE MUST BE HIGHER THAN ANY APPEARING IN
                00220 ;         THE CURRENT PROGRAM SEGMENT IN THE BUFFER. WHEN
                00230 ;         CREATING A NEW PROGRAM FROM A GROUP OF CSAVE'D
                00240 ;         FILES, FIRST CLOAD THE FILE WITH THE LOWEST LINE
                00250 ;         NUMBERS , THEN SUCCESIVELY MERGE EACH ADDITIONAL
                00260 ;         FILE IN ORDER OF LINE NUMBERS.
                00270 ;SYNTAX :
                00280 ;         MERGE HAS SAME SYNTAX AS CLOAD EXCEPT FOR SUPPORT
                00290 ;         OF " ? " COMPARE OPTION
                00300 ;
                00310 ;NOTE:
                00320 ;         THIS PROGRAM IS 'ORG'ED TO NOT INTERFERE WITH THE
                00330 ;         AEXIT PROGRAM.
7ED6            00340           ORG     7ED6H
7ED6 CD9302     00350 MERGE     CALL    293H        ; EVALUATE THE DRIVE NUMBER, START
                00360                               ; IT, AND FIND SYNCH BYTE
```

```
7ED9 2B         00370           DEC     HL
7EDA D7         00380           RST     10H     ; GET FIRST EXPRESSION CHARACTER
7EDB 3E00       00390           LD      A,0     ; NO NAME FLAG
7EDD 2807       00400           JR      Z,NONAME ; IF END OF STATEMENT
7EDF CD3723     00410           CALL    2337H   ; EVALUATE EXPRESSION
7EE2 CD132A     00420           CALL    2A13H   ; GET POINTER TO STRING VALUE
7EE5 1A         00430           LD      A,(DE)  ; GET FIRST CHARACTER = FILENAME
7EE6 5F         00440 NONAME    LD      E,A     ; SAVE
7EE7 AF         00450           XOR     A       ; ZERO
7EE8 57         00460           LD      D,A     ; CLOAD FLAG = STORE
                00470 ;         SEARCH FOR FILE NAME
7EE9 0603       00480 S1        LD      B,3     ; COUNT
7EEB CD3502     00490 S2        CALL    235H    ; GET CHAR
7EEE D6D3       00500           SUB     0D3H    ; FILE NAME FLAG
7EF0 20F7       00510           JR      NZ,S1   ;     NO
7EF2 10F7       00520           DJNZ    S2      ; YES, THREE IN A ROW ?
7EF4 CD3502     00530           CALL    235H    ; YES, GET FILE NAME
                00540 ;         COMPARE FILE NAME IF REQUESTED
7EF7 1C         00550           INC     E
7EF8 1D         00560           DEC     E       ; TEST GIVEN
7EF9 2803       00570           JR      Z,NONE  ; NO
7EFB BB         00580           CP      E
7EFC 2008       00590           JR      NZ,NXTFIL ; NO MATCH , SKIP FILE
7EFE 2AF940     00600 NONE      LD      HL,(40F9H)  ; PTR TO END OF PGM TEXT
7F01 2B         00610           DEC     HL
7F02 2B         00620           DEC     HL      ;  MINUS 2 IS END OF LAST STATMENT
7F03 C35F2C     00630           JP      2C5FH   ; ENTER CLOAD LOGIC TO STORE
7F06 323E3C     00640 NXTFIL    LD      (3C3EH),A ; FILE NAME ON CRT
7F09 0603       00650 E1        LD      B,3
7F0B CD3502     00660 E2        CALL    235H
7F0E B7         00670           OR      A
7F0F 20F8       00680           JR      NZ,E1
7F11 10F8       00690           DJNZ    E2
7F13 CD9602     00700           CALL    296H    ; EOF, FIND NEXT SYNCH BYTE
7F16 18D1       00710           JR      S1      ; NEXT FILE NAME
                00720 ;         INSTALL COMMAND
418B            00730           ORG     418BH
418B C3D67E     00740           JP      MERGE
06CC            00750           END     06CCH
00000 TOTAL ERRORS
E1      7F09 00650   00680
E2      7F0B 00660   00690
MERGE   7ED6 00350   00740
NONAME  7EE6 00440   00400
NONE    7EFE 00600   00570
NXTFIL  7F06 00640   00590
S1      7EE9 00480   00510 00710
S2      7EEB 00490   00520
```

Figure 3

```
00100 ; EXTENSION OF LEVEL II BASIC USING DISK BASIC & FUNCTION
00110 ; EXIT AT 4194H IN RAM
00120 ;       BY:     JOHN R. HIND
00130 ;               4100 PICKWICK DR.
00140 ;               RALEIGH, NC  27612
00150 ;
00160 ;       FUNCTIONS IMPLEMENTED :
00170 ;               HEX CONSTANT    -> &H 7DF
00180 ;               HEX CONVERT     -> &H("7DC")
00190 ;               LOAD WORD       -> &L( ADDR )
00200 ;               ROTATE LEFT HEX CONSTANT COUNT
00210 ;                               -> &R4(VALUE)
```

```
                  00220 ;                ROTATE LEFT VARIABLE COUNT
                  00230 ;                         -) &R(COUNT)(VALUE)
                  00240 ;                EXCHANGE WORD AT HEX CONSTANT ADDR
                  00250 ;                         -) &E 40F1(NEW-VALUE)
                  00260 ;                EXCHANGE WORD VARIABLE ADDR
                  00270 ;                         -) &E(ADDR)(NEW-VALUE)
                  00280 ;                XOR HEX CONSTANT WITH VALUE
                  00290 ;                         -) &X 1DC (VALUE)
                  00300 ;                XOR VARIABLE WITH VALUE
                  00310 ;                         -) &X(MASK)(VALUE)
                  00320 ;

7F18              00330          ORG     7F18H     ;MEM SIZE = 32536
7F18 D7           00340 AEXIT    RST 10H           ;GET FUNCTION TYPE
7F19 CABC7F       00350          JP      Z,BADEX
7F1C FE48         00360          CP      48H       ;HEX CONVERSION
7F1E 201F         00370          JR      NZ,NOTH
7F20 D7           00380          RST     10H
7F21 FE28         00390          CP      28H       ;PARM IE &H("FF")
7F23 2812         00400          JR      Z,HPAR
7F25 1E08         00410          LD      E,8       ;MAX SCAN LENGTH
7F27 CDDC7F       00420          CALL    CONHS
7F2A E5           00430          PUSH    HL
7F2B ED432141     00440 EXIT     LD      (4121H),BC ;SAVE VALUE
7F2F 3E02         00450          LD      A,2
7F31 32AF40       00460          LD      (40AFH),A ;SET TYPE INTEGER
7F34 C39008       00470          JP      890H      ;EXIT
7F37 CDA47F       00480 HPAR     CALL    GETSTR
7F3A CDDC7F       00490          CALL    CONHS
7F3D 18EC         00500          JR      EXIT
7F3F FE4C         00510 NOTH     CP      4CH       ;LOAD?
7F41 2009         00520          JR      NZ,NOTL
7F43 23           00530          INC     HL
7F44 CDBF7F       00540          CALL    GETINT
7F47 4E           00550          LD      C,(HL)
7F48 23           00560          INC     HL
7F49 46           00570          LD      B,(HL)
7F4A 18DF         00580          JR      EXIT
7F4C FE52         00590 NOTL     CP      52H       ;ROTATE LEFT?
7F4E 202F         00600          JR      NZ,NOTR
7F50 CD597F       00610          CALL    GPAIR
7F53 53           00620          LD      D,E
7F54 CDCE7F       00630          CALL    RBCLD
7F57 18D2         00640          JR      EXIT
                  00650 ;  GET PAIR OF INTEGER VALUES , PLACE FIRST VALUE IN
                  00660 ;        DE AND SECOND IN BC. (SP) CONTAINS THE
                  00670 ;        NEXT PROGRAM TEXT ADDR.
                  00680 ;        FIRST VALUE IS EITHER A HEX CONSTANT IN
                  00690 ;        TEXT OR AN  " ( EXPRESSION )"; SECOND VALUE
                  00700 ;        IS ALWAYS  " ( EXPRESSION )"
7F59 D7           00710 GPAIR    RST     10H       ;OPTION
7F5A 2002         00720          JR      NZ,GOON
7F5C C1           00730          POP     BC
7F5D C9           00740          RET
7F5E FE28         00750 GOON     CP      28H
7F60 2808         00760          JR      Z,RPAR    ; FIRST IS VARIABLE
7F62 1E08         00770          LD      E,8       ; HEX CONSTANT
7F64 CDDC7F       00780          CALL    CONHS
7F67 C5           00790          PUSH    BC
7F68 1804         00800          JR      RGETV
7F6A CDBF7F       00810 RPAR     CALL    GETINT
7F6D E3           00820          EX      (SP),HL
7F6E 2B           00830 RGETV    DEC     HL
```

```
7F6F D7       00840             RST     10H
7F70 2003     00850             JR      NZ,RVAL
7F72 C1       00860             POP     BC
7F73 C1       00870             POP     BC
7F74 C9       00880             RET
7F75 CDBF7F   00890 RVAL        CALL    GETINT
7F78 E5       00900             PUSH    HL
7F79 C1       00910             POP     BC
7F7A E1       00920             POP     HL
7F7B D1       00930             POP     DE
7F7C E3       00940             EX      (SP),HL
7F7D E5       00950             PUSH    HL
7F7E C9       00960             RET
7F7F FE45     00970 NOTR        CP      45H
7F81 200F     00980             JR      NZ,NOTE
7F83 CD597F   00990             CALL    GPAIR
7F86 D5       01000             PUSH    DE
7F87 E1       01010             POP     HL
7F88 5E       01020             LD      E,(HL)
7F89 71       01030             LD      (HL),C
7F8A 23       01040             INC     HL
7F8B 56       01050             LD      D,(HL)
7F8C 70       01060             LD      (HL),B
7F8D D5       01070             PUSH    DE
7F8E C1       01080             POP     BC
7F8F C32B7F   01090             JP      EXIT
7F92 FE58     01100 NOTE        CP      58H
7F94 200C     01110             JR      NZ,NOTX
7F96 CD597F   01120             CALL    GPAIR
7F99 78       01130             LD      A,B
7F9A AA       01140             XOR     D
7F9B 47       01150             LD      B,A
7F9C 79       01160             LD      A,C
7F9D AB       01170             XOR     E
7F9E 4F       01180             LD      C,A
7F9F C32B7F   01190             JP      EXIT
7FA2 1818     01200 NOTX        JR      BADEX
              01210 ;GET STRING PARAMETER: HL=>ADDR, E=LEN,
              01220 ;                     , (SP)=>STATEMENT PTR
7FA4 CD2C25   01230 GETSTR      CALL    252CH
7FA7 E3       01240             EX      (SP),HL
7FA8 E5       01250             PUSH    HL
7FA9 3AAF40   01260             LD      A,(40AFH)
7FAC FE03     01270             CP      3
7FAE 200A     01280             JR      NZ,BAD
7FB0 2A2141   01290             LD      HL,(4121H)
7FB3 5E       01300             LD      E,(HL)
7FB4 23       01310             INC     HL
7FB5 7E       01320             LD      A,(HL)
7FB6 23       01330             INC     HL
7FB7 66       01340             LD      H,(HL)
7FB8 6F       01350             LD      L,A
7FB9 C9       01360             RET
7FBA E1       01370 BAD         POP     HL
7FBB E1       01380             POP     HL
7FBC C32D01   01390 BADEX       JP      012DH
              01400 ; GET INTEGER VALUE PARAMETER, HL = INT
7FBF CD2C25   01410 GETINT      CALL    252CH
7FC2 E3       01420             EX      (SP),HL
7FC3 E5       01430             PUSH    HL
7FC4 3AAF40   01440             LD      A,(40AFH)
7FC7 FE03     01450             CP      3
7FC9 28EF     01460             JR      Z,BAD
```

```
7FCB C37F0A    01470          JP       0A7FH
               01480 ; ROTATE BC LEFT D TIMES
7FCE 14        01490 RBCLD    INC      D
7FCF 15        01500 RBCL1    DEC      D
7FD0 C8        01510          RET      Z
7FD1 37        01520          SCF
7FD2 CB11      01530          RL       C
7FD4 CB10      01540          RL       B
7FD6 38F7      01550          JR       C,RBCL1
7FD8 CB81      01560          RES      0,C
7FDA 18F3      01570          JR       RBCL1
               01580 ; COVERT HEX STRING AT HL TO BC, STRING MUST END IN
               01590 ; A NONBLANK NONHEX CHARACTER
7FDC 1C        01600 CONHS    INC      E
7FDD 2B        01610          DEC      HL
7FDE 010000    01620          LD       BC,0
7FE1 23        01630 CONHS1   INC      HL
7FE2 1D        01640          DEC      E
7FE3 C8        01650          RET      Z
7FE4 7E        01660          LD       A,(HL)
7FE5 FE20      01670          CP       20H
7FE7 28F8      01680          JR       Z,CONHS1
7FE9 D630      01690          SUB      30H
7FEB F8        01700          RET      M
7FEC FE0A      01710          CP       0AH
7FEE 3808      01720          JR       C,CONHS2
7FF0 D611      01730          SUB      11H
7FF2 F8        01740          RET      M
7FF3 D606      01750          SUB      6H
7FF5 F0        01760          RET      P
7FF6 C610      01770          ADD      A,10H
7FF8 1604      01780 CONHS2   LD       D,4
7FFA CDCE7F    01790          CALL     RBCLD
7FFD B1        01800          OR       C
7FFE 4F        01810          LD       C,A
7FFF 18E0      01820          JR       CONHS1
               01830 ; INSTALL EXIT AT BRANCH VECTOR ADDR
4194           01840          ORG      4194H
4194 C3187F    01850          JP       AEXIT
06CC           01860          END      06CCH
00000 TOTAL ERRORS
AEXIT   7F18 00340   01850
BAD     7FBA 01370   01280 01460
BADEX   7FBC 01390   00350 01200
CONHS   7FDC 01600   00420 00490 00780
CONHS1  7FE1 01630   01680 01820
CONHS2  7FF8 01780   01720
EXIT    7F2B 00440   00500 00580 00640 01090 01190
GETINT  7FBF 01410   00540 00810 00890
GETSTR  7FA4 01230   00480
GOON    7F5E 00750   00720
GPAIR   7F59 00710   00610 00990 01120
HPAR    7F37 00480   00400
NOTE    7F92 01100   00980
NOTH    7F3F 00510   00370
NOTL    7F4C 00590   00520
NOTR    7F7F 00970   00600
NOTX    7FA2 01200   01110
RBCL1   7FCF 01500   01550 01570
RBCLD   7FCE 01490   00630 01790
RGETV   7F6E 00830   00800
RPAR    7F6A 00810   00760
RVAL    7F75 00890   00850
```

END

SPECIAL DELIVERY with EXTRACT

A 100% Machine Language Word Processor

TRS-80® + Electric Pencil® or Radio Shack's Scripsit®

We can't stop improving and expanding the capabilities of your TRS-80®! By using SPECIAL DELIVERY with EXTRACT and either Electric Pencil® or Radio Shack's Scripsit® you can get even more out of your computer. From just one package, you will get all this:

MAILFORM: Create MAILFILE: The ONLY complete name and address list entry/editor program written in machine language. Instant search on any field, complete cursor control, just fill in the form!

MAILRITE: Print letters written with either the Electric Pencil® or Radio Shack's Scripsit® inserting information from a MAILFILE into the letter for personalizing and addressing. You can send a personalized letter to one person then a different personalized letter to a second person with true typist quality from your fine printer. Features: Indents, Underscore, Bold Type, End of Page Stop, Address Envelopes, unlimited insertion from address list and More!

EXTRACT: Take out information from MAILFORM, the machine language mailing list. Find the names you need by Zip Code, Street Address, Gender, Age or any other way!

SORT: In-Memory sort on an entire address list using any field as the key. This program can sort an entire list in a matter of seconds!

LABEL: Prints labels from MAILFILE.

CONVERT: Make MAILFILE from RS mail list.

MOD II version available soon

SPECIAL DELIVERY will run on your TRS-80® with TRSDOS,® NEWDOS® or any other TRS®-like DOS.

If you bought SPECIAL DELIVERY from us, send us the disk and we will update it to include EXTRACT for $25. If you haven't already taken advantage of our SPECIAL DELIVERY program, we'll send you the complete program including EXTRACT for just $125 (Disk). We can also send you the Electric Pencil® for $150.

Software Etc . . .
1839 Chamberlain Drive
Carrollton, Texas 75007.
Phone Orders: (214) 492-0515

Demand a Demonstration from your local Dealer or write for a brochure of our complete line of fine software.

Board Games-1, CS-3001 (16K)

- **Mugwump** $7.95

Mugwump is a board game which uses a 10x10 grid on which four friendly Mugwumps are hiding. Your mission is to locate these mysterious animals and capture them.

- **Flip Disc**

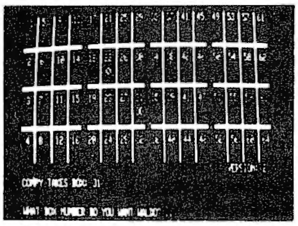
Are you an Othello freak? Flip Disc is a program which will turn your computer into an excellent opponent. Three different skill levels, (good, expert, and genius), provide an introduction for the novice and continuing interest for the experienced player.

- **Wumpus**

In game 1, you scour a network of underground caves in search of the prized Wumpus. Bagging a Wumpus wins the game, but if you accidentally stumble into his cave, the Wumpus will enjoy a tasty dinner of sauteed computer freak.

- **Wumpus 2**

If you master the dodecahedron cave network in Wumpus 1, you may proceed to Wumpus 2 which allows you to choose from five different caves, or you can design your own.



- **Qubic**

Qubic is a three dimensional Tic Tac Toe game. The game is played in a 3 dimensional cube (4x4x4). The object is to outwit the computer and place four pieces in any straight line.

- **Backgammon**

This is the TRS-80 adaptation of the popular board game. Backgammon uses graphics and all the standard backgammon rules, not a strange computer variation. The computer is your opponent in this version, written by Scott Adams of "Adventure" fame

How To Order

Send order and payment to **Creative Computing**, P.O. Box 789-M, Morristown, NJ 07960. Add $1.00 shipping and handling per order (foreign, $2.50) N.J. residents add 5% sales tax. Visa, MasterCharge and American Express orders welcome. For faster service, call in your bank card order toll free to: **800-631-8112.**(In NJ, call (201) 540-0445.)

For the SERIOUS Game Player

• • •

sensational software

Who Is Creative Computing?

Creative Computing consists of five divisions serving you. **Creative Computing magazine** is the number 1 magazine of software and applications. **Creative Computing Press** publishes a wide variety of books, art prints, posters and T-shirts for the computer enthusiast. And **Creative Computing Software** produces and markets software on cassette and floppy disk for a wide variety of computers for home, school, and small business.

If your dealer does not carry the full line of Creative Computing products, please send three first-class stamps for a free catalog of products.

Strategy Games, CS-3005 (16K)

- **Tunnel Vision** $7.95

You are transported into a massive labyrinth and must find the exit or be lost forever. This is an excellent example of three dimensional perspective using TRS-80 graphics.

- **Evasion**

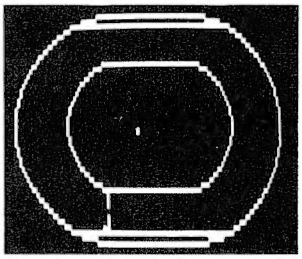
In this real time game, you are pursued around the game board by an evil-looking snake. Variations of play include two different speeds and hyper-jumps which randomly relocate you on the board. Looking for an escape? Try Evasion.

- **Jigsaw**

Jigsaw is a computer-age puzzle game making extensive use of TRS-80 graphics. The computer generates a random puzzle and puzzle board. Using a combination of deductive reasoning and luck you must fit the graphically represented puzzle piece into place.

- **The Masters**

Are you a wandering pro or just a Sunday golfer who would like to keep in practice? Once you're on the green, a worm's-eye view is displayed for putting.



- **Motor Racing**

Motor Racing combines real time racing action with advanced graphics functions. The graphics and animation make Motor Racing fun to watch as well as play.

WRITE FOR...

FREE

SOFTWARE CATALOG

Space Games-3, CS-3002 (16K)

- **Ultra-Trek** $7.95

Ultra-Trek is a fast-paced version of Star Trek, complete with "real time" action graphics, lasers, Nilon space mines, high energy photon torpedoes enemy ships that move, and an experimental ray which does something different each time you use it. You must act quickly to save yourself and the Federation.

- **Star Lanes**

Imagine yourself the president of an intergalactic shipping company. If you're successful, you may be named Imperial Advisor on Economic Affairs. Entrepreneurs: to your ships.

- **Star Wars**

If you hate Darth Vader, you'll love Star Wars. This real time game is fun for aliens of all ages. May the Force be with you!

- **Romulan**

Your mission is to destroy an invading Romulan space craft. Maneuver through space and around stars looking for the deadly enemy, but be careful! The nasty Romulans fire back.

Air Traffic Controller, CS-3006 (16K) $7.95

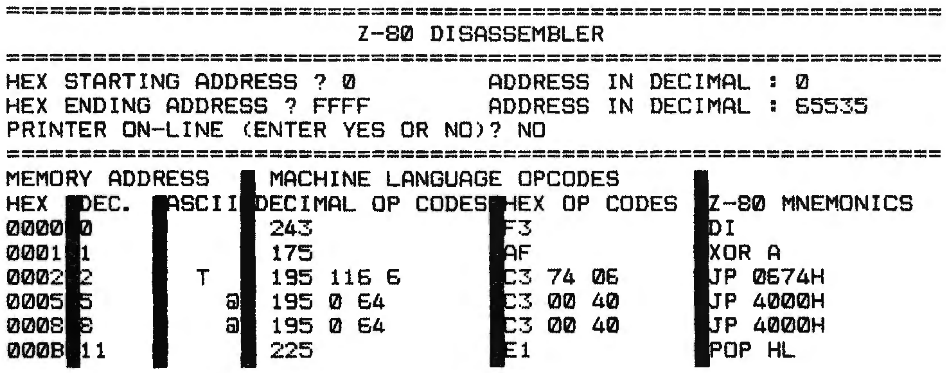
This real time machine language program puts you in the chair of an air traffic controller. There are 27 airplanes — jets and prop planes — which must be controlled as they land, take off and fly over your air space. You give the orders to change altitude, turn, maintain a holding pattern, clear for approach, and land at your two airports. This realistic simulation includes navigational beacons, and requires planes to take off and land into the wind. Air Traffic Controller was written by an air traffic controller and is a favorite of the Creative Computing staff!

creative computing

A Basic Z-80 DISASSEMBLER

Bill Atchison
Elwood, KS

Here is a Basic Disassembler for 16K Level II and up. It occupies just 6K of memory, and will disassemble from 0000 to the top of memory. Formatted hardcopy is provided via the lineprinter - or you can view it on the video. Figure 1, below, shows how your screen should look. Note that it displays Hex, decimal and ASCII, as well as the Z-80 nmemonics.



```
==============================================================
                         Z-80 DISASSEMBLER
==============================================================
HEX STARTING ADDRESS ? 0          ADDRESS IN DECIMAL : 0
HEX ENDING ADDRESS ? FFFF         ADDRESS IN DECIMAL : 65535
PRINTER ON-LINE (ENTER YES OR NO)? NO
==============================================================
MEMORY ADDRESS      MACHINE LANGUAGE OPCODES
HEX   DEC.    ASCII DECIMAL OP CODES  HEX OP CODES   Z-80 MNEMONICS
0000  0              243               F3             DI
0001  1              175               AF             XOR A
0002  2         T    195 116 6         C3 74 06       JP 0674H
0005  5            @ 195 0 64          C3 00 40       JP 4000H
0008  8            @ 195 0 64          C3 00 40       JP 4000H
000B  11             225               E1             POP HL
```

Figure 1

```
NOTE: 132-CHARACTER LINE PRINTER OWNERS MAY MAKE THE
        FOLLOWING CHANGES TO ALLOW THE INCLUSION OF A LINE
        FOR COMMENTS ON THE HARDCOPY:
135 ZQ$=STRING$(35,95)
160 IFLP$="YES"THENLPRINT"MEMORY ADDRESS :",,"MACHINE LANGUAGE
    OPCODES :":LPRINT"HEX DECIMAL","ASCII","DECIMAL OP CODE","HE
    X OP CODE","Z-80 MNEMONICS"
300 LPRINTZ$;"  ";V,W1$,W$,A$,M,ZQ$:GOTO350
```

```
10 '***Z-80 DISASSEMBLER BY WILLIAM C. ATCHISON
20 '***REQUIRES A 16K LEVELII (AND UP) BASIC TRS-80
30 '***THIS PROGRAM LEAVES 10K FREE SPACE  TO LOAD PROGRAMS
40 '***(MORE SPACE IN LARGER SYSTEMS!!) TO DISASSEMBLE.
50 '***THE DISASSEMBLER WILL WORK FROM X'0000' TO X'FFFF'
60 '***IN MEMORY (0-65535 IN DECIMAL).
70 '***FORMATTED HARDCOPY IS PROVIDED VIA LINEPRINTER .
80 '***NOTE : DON'T FORGET TO REMOVE REMARKS (THEY USE MEMORY)!
90 CLS
```

```
100 CLEAR350:PRINT@960,STRING$(64,61)+STRING$(24,32)+"Z-80 DISA
    SSEMBLER"+CHR$(13)+STRING$(64,61);
110 CLEAR500:DEFINTA-J,U,Z:DEFSTRL-T:L1=CHR$(191):L="0123456789
    ABCDEF":R="BCDEHL:A":D1$="-++"
120 INPUT"HEX STARTING ADDRESS ";J$:GOSUB1830 :X=K:INPUT"HEX EN
    DING ADDRESS ";J$:GOSUB1830 :Y=K:IFX>YTHENPRINT:PRINT"START
    GREATER THAN END?":PRINT:GOTO120
130 INPUT"PRINTER ON-LINE (ENTER YES OR NO)";LP$
140 PRINTSTRING$(64,61);"MEMORY ADDRESS  ";CHR$(191);" MACHINE
    LANGUAGE OPCODES      ";CHR$(191)
150 PRINT"HEX ";CHR$(191);"DEC. ";CHR$(191);"ASCII";CHR$(191);"
    DECIMAL OP CODES";CHR$(191);"HEX OP CODES ";CHR$(191);"Z-80
    MNEMONICS"
160 IFLP$="YES"THENLPRINT"MEMORY ADDRESS :",,"MACHINE LANGUAGE
    OPCODES :":LPRINT"HEX","DECIMAL","ASCII","DECIMAL OP CODE","
    HEX OP CODE","Z-80 MNEMONICS"
170 Z=90:A=VARPTR(Z)
180 '***START DISASSEMBLY LOOP
190 FORV=XTOY:IFV>32767THENV1=-INT(65535-V+1)ELSEV1=V
200 IN$=INKEY$:IFIN$="↑"THEN100
210 Z=VAL(STR$(V1)):A1=PEEK(A+1):A2=PEEK(A):A3=PEEK(Z)
220 '***GOTO OCTAL SUBROUTINE AND RETURN (B,C,D)
230 I=Z:G=A3:GOSUB480  :A$=Q1:M=""
240 G1=A1:G2=A2:GOSUB500  :Z$=NN
250 '***BRANCH ON VALUE OF FIRST OCTAL DIGIT (B)
260 GOSUB380  :ONB+1GOSUB610  ,960  ,980  ,1080
270 '***PRINT ON LINEPRINTER
280 IFLP$<>"YES"THEN320
290 W1$="":W$="":FORW=ITOZ:W1=PEEK(W):W$=W$+STR$(W1):IFW1>31AND
    W1<128THENW1$=W1$+CHR$(W1)+" ":NEXTELSEW1$=W1$+"  ":NEXT
300 LPRINTZ$;TAB(6);V;TAB(10);W1$;TAB(20);W$;TAB(38);A$;STRING$
    (12-LEN(A$)," ");M:GOTO350
310 '***PRINT ON SCREEN
320 PRINT@960,Z$;L1;RIGHT$(STR$(V),LEN(STR$(V))-1);
330 PRINT@970,L1;:W$="":FORW=ITOZ:W1=PEEK(W):W$=W$+STR$(W1):IFW
    1>31ANDW1<128THENPRINTCHR$(W1)+" ";:NEXTELSEPRINT"  ";:NEXT
340 PRINT@976,L1;W$;:PRINT@993,L1;A$;:PRINT@1007,L1;M
350 IFZ>=0THENV=ZELSEV=V+ABS(I-Z)
360 NEXTV:GOTO100
370 '***OCTAL CONVERSION SUBROUTINE AND REGISTER COMPUTATION
380 B=(A3AND192)/64:C=(A3AND56)/8:D=A3AND7:C$=MID$(R,C+1,1):D$=
    MID$(R,D+1,1):IFC$=":"THENC$="(HL)"
390 IFD$=":"THEND$="(HL)":RETURNELSERETURN
400 '***REGISTER PAIR COMPUTATION SUBROUTINES
410 Q=MID$("BCDEHLAF",C+1,2):RETURN
420 S=MID$("BCDEHLSP",C,2):RETURN
430 '***CONDITION COMPUTATION SUBROUTINE
440 T=MID$("NZ ZNC CPOPE P M",C*2+1,2):RETURN
450 Z=Z+1:G=PEEK(Z):GOSUB480  :A$=A$+" "+Q1:N1=Q1+"H":RETURN
460 GOSUB450  :A3=G:GOSUB380  :RETURN
470 '***CONVERT DECIMAL TO ONE-BYTE HEX (0-255)
480 Q1=MID$(L,((GAND240)/16)+1,1)+MID$(L,(GAND15)+1,1):RETURN
490 '***CONVERT DECIMAL TO TWO-BYTE HEX (0-65535)
500 G=G1:GOSUB480  :NN=Q1:G=G2:GOSUB480  :NN=NN+Q1:RETURN
510 G1=PEEK(Z+2):G2=PEEK(Z+1):GOSUB500  :Q1=NN:Z=Z+2:A$=A$+" "+
    RIGHT$(Q1,2)+" "+LEFT$(Q1,2):N1=Q1+"H":RETURN
```

```
990 IFB=3THENRETURNELSEM=M+D$:RETURN
1000 M="ADD A,":RETURN
1010 M="ADC A,":RETURN
1020 M="SUB ":RETURN
1030 M="SBC A,":RETURN
1040 M="AND ":RETURN
1050 M="XOR ":RETURN
1060 M="OR ":RETURN
1070 M="CP ":RETURN
1080 ONDGOTO1100 ,1140 ,1150 ,1220 ,1230 ,1260 ,1270
1090 GOSUB440  :M="RET "+T:RETURN
1100 IF(CAND1)=0THENGOSUB410  :M="POP "+Q:RETURN
1110 IFC=1THENM="RET"ELSEIFC=3THENM="EXX"
1120 IFC=5THENM="JP  (HL)"ELSEIFC=7THENM="LD SP,HL"
1130 RETURN
1140 GOSUB510  :GOSUB440  :M="JP "+T+","+N1:RETURN
1150 ONCGOTO1290 ,1160 ,1170 ,1180 ,1190 ,1200 ,1210 :GOSUB510
     :M="JP "+N1:RETURN
1160 GOSUB450  :M="OUT ("+N1+"),A":RETURN
1170 GOSUB450  :M="IN A,("+N1+")":RETURN
1180 M="EX (SP),HL":RETURN
1190 M="EX DE,HL":RETURN
1200 M="DI":RETURN
1210 M="EI":RETURN
1220 GOSUB510  :GOSUB440  :M="CALL "+T+","+N1:RETURN
1230 IF(CAND1)=0THENGOSUB410  :M="PUSH "+Q:RETURN
1240 IFC=1THENGOSUB510  :M="CALL "+N1:RETURN
1250 IFC=5THEN1410 ELSEIFC=3THEN1640 ELSEIFC=7THEN1660
1260 GOSUB980  :GOSUB450  :M=M+N1:RETURN
1270 G=C*8:GOSUB480  :M="RST "+Q1+"H":RETURN
1280 '***SUBROUTINE TO COMPUTE TWO-BYTE "CB" INSTRUCTIONS
1290 GOSUB460:IFB<>0THEN1380
1300 ONCGOTO1310 ,1320 ,1330 ,1340 ,1350 ,1360 ,1370 :M="RLC "+
    D$:RETURN
1310 M="RRC "+D$:RETURN
1320 M="RL  "+D$:RETURN
1330 M="RR  "+D$:RETURN
1340 M="SLA "+D$:RETURN
1350 M="SRA "+D$:RETURN
1360 M="* * ERROR * *":RETURN
1370 M="SRL "+D$:RETURN
1380 IFB=1THENM="BIT"ELSEIFB=2THENM="RES"ELSEIFB=3THENM="SET"
1390 M=M+STR$(C)+","+D$:RETURN
1400 '***COMPUTE "ED" TWO-BYTE INSTRUCTION GROUP
1410 GOSUB460  :ONBGOTO1420 ,1600 :GOTO1360
1420 ONDGOTO1440 ,1450 ,1470 ,1490 ,1500 ,1510 ,1530
1430 M="IN "+C$+",(C)":RETURN
1440 M="OUT (C),"+C$:RETURN
1450 IF(CAND1)=0THENC=C+1:GOSUB420  :M="SBC HL,"+S:RETURN
1460 GOSUB420  :M="ADC HL,"+S:RETURN
1470 GOSUB510  :IF(CAND1)=0THENC=C+1:GOSUB420  :M="LD ("+N1+"),
    "+S:RETURN
1480 GOSUB420  :M="LD "+S+",("+N1+")":RETURN
1490 IFC=0THENM="NEG":RETURNELSE1360
1500 IFC=0THENM="RETN":RETURNELSEIFC=1THENM="RETI":RETURNELSE13
    60
```

```
520 '***INSERT (IX,IY) OR (IX+00H,IY+00H) IN
530 '***THE (DD,FD) TWO-BYTE INSTUCTION GROUP
540 K$=I$:M=M+" ":I$="(HL)":GOSUB1900 :IFJ=0THEN580
550 M1=MID$(M,1,J-1):M2=MID$(M,J+4):M=M1+K$+M2
560 I$="HL":GOSUB1900 :IFJTHENM=M2+K$
570 I$=K$:RETURN
580 I$="HL":GOSUB1900 :IFJ<>0THEN590  ELSERETURN
590 P1=LEFT$(M,J-1):IFJ>=LEN(J$)-2THEN600  ELSEP2=RIGHT$(M,LEN(
    J$)-J+2)
600 M=P1+J$+P2:RETURN
610 ONDGOTO740  ,760  ,840  ,850  ,860  ,870  ,880
620 IFC<2THEN660ELSEGOSUB450:N1%=PEEK(Z)+1:IFN1%<128THENN1!=Z+N
    1%ELSEN1!=Z-256+N1%
630 IFN1!>32767THENN1%=-1*(65536-N1!)ELSEN1%=N1!
640 G1=PEEK(VARPTR(N1%)+1):G2=PEEK(VARPTR(N1%)):GOSUB490
650 NN=NN+"H"
660 ONCGOTO670  ,680  ,690  ,700  ,710  ,720  ,730  :M="NOP":RE
    TURN
670 M="EX AF,AF'":RETURN
680 M="DJNZ "+NN:RETURN
690 M="JR "+NN:RETURN
700 M="JR NZ,"+NN:RETURN
710 M="JR Z,"+NN:RETURN
720 M="JR NC,"+NN:RETURN
730 M="JR C,"+NN:RETURN
740 IF(CAND1)=0THENGOSUB510  :C=C+1:GOSUB420  :M="LD "+S+","+N1
    :RETURN
750 GOSUB420  :M="ADD HL,"+S:RETURN
760 ONCGOTO770  ,780  ,790  ,800  ,810  ,820  ,830  :M="LD (BC)
    ,A":RETURN
770 M="LD A,(BC)":RETURN
780 M="LD (DE),A":RETURN
790 M="LD A,(DE)":RETURN
800 GOSUB510  :M="LD ("+N1+"),HL":RETURN
810 GOSUB510  :M="LD HL,("+N1+")":RETURN
820 GOSUB510  :M="LD ("+N1+"),A":RETURN
830 GOSUB510  :M="LD A,("+N1+")":RETURN
840 IF(CAND1)=0THENC=C+1:GOSUB420  :M="INC "+S:RETURNELSEGOSUB4
    20  :M="DEC "+S:RETURN
850 M="INC "+C$:RETURN
860 M="DEC "+C$:RETURN
870 GOSUB450  :M="LD "+C$+" ,"+N1:RETURN
880 ONCGOTO890  ,900  ,910  ,920  ,930  ,940  ,950  :M="RLCA":R
    ETURN
890 M="RRCA":RETURN
900 M="RLA":RETURN
910 M="RRA":RETURN
920 M="DAA":RETURN
930 M="CPL":RETURN
940 M="SCF":RETURN
950 M="CCF":RETURN
960 IFC=6ANDD=6THENM="HALT":RETURN
970 M="LD "+C$+","+D$:RETURN
980 ONC+1GOSUB1000  ,1010  ,1020  ,1030  ,1040  ,1050  ,1060 ,1
    070
```

```
1510 IFC=0THENM="IM 0"ELSEIFC=2THENM="IM 1"ELSEIFC=3THENM="IM 2
     "
1520 IFM=""THEN1360 ELSERETURN
1530 ONC+1GOTO1540 ,1550 ,1560 ,1570 ,1580 ,1590 :GOTO1360
1540 M="LD I,A":RETURN
1550 M="LD R,A":RETURN
1560 M="LD A,I":RETURN
1570 M="LD A,R":RETURN
1580 M="RRD":RETURN
1590 M="RLD":RETURN
1600 IFD=0THENM="LD"ELSEIFD=1THENM="CP"
1610 IFD=2THENM="IN"ELSEIFD=3THENM="OUT"
1620 IFM=""THEN1360 ELSEM=M+MID$("I D IRDR",(C-4)*2+1,2):RETURN
1630 '***SET VARIABLES FOR "DD" INSTRUCTIONS
1640 I$="(IX+:::)":J$="IX":GOTO1680
1650 '***SET VARIABLES FOR "FD" INSTRUCTIONS
1660 I$="(IY+:::)":J$="IY"
1670 '***COMPUTE "DD" AND "FD" TWO BYTE INSTRUCTIONS
1680 GOSUB460:IFG=203THEN1930ELSEIFG=54THEN2000ELSEONB+1GOSUB61
     0,960,980,1080:GOSUB540
1690 GOSUB1900 :IFJ=0ANDC<>7THENRETURN
1700 IFB<>0THEN1740
1710 ONASC(RIGHT$(J$,1))-87GOTO1720,1730
1720 M=M+","+MID$("BCDEIXSP",INT(C/2)*2+1,2):RETURN
1730 M=M+","+MID$("BCDEIYSP",INT(C/2)*2+1,2):RETURN
1740 Z=I+1:A3=PEEK(Z):GOSUB380
1750 GOSUB450
1760 M="":ONB+1GOSUB610  ,960  ,980  ,1080 :GOSUB540
1770 I$=":::":GOSUB1900:IFJ=0THENRETURN
1780 IFG<128THEND=GELSED=-256+G:G=ABS(D)
1790 GOSUB480
1800 SD%=SGN(D):M=LEFT$(M,J-2)+MID$(D1$,SD%+2,1)+Q1+"H"+MID$(M,
     J+3)
1810 RETURN
1820 '***HEX TO DECIMAL CONVERSION
1830 A=LEN(J$):IFA>4ORJ$=""THENK=-1:GOTO1870
1840 IFA=4THENH$=J$ELSEH$=STRING$(4-A,32)+J$
1850 K=0:FORI=(5-A)TO4:J$=MID$(H$,I,1):IFASC(J$)>=65ANDASC(J$)<
     =70THENJ=ASC(J$)-55ELSEJ=VAL(J$):IFJ=0ANDJ$<>"0"THENK=-1
1860 IFJ<0THENK=-1ELSEK=K+J*2↑((4-I)*4):NEXT
1870 IFK<0THENK=0:PRINT"ONLY ENTER HEX":INPUT"HEX :";J$:GOTO183
     0
1880 PRINT@929,"ADDRESS IN DECIMAL :"K:RETURN
1890 '***INSTRING SUBROUTINE
1900 FORJ=1TOLEN(M)-LEN(I$)+1:IFI$=MID$(M,J,LEN(I$))RETURN
1910 NEXT:J=0:RETURN
1920 REM** TO HANDLE 4-BYTE FDCB AND DDCB INSTRUCTIONS
1930 GOSUB450
1940 IFG<128THEND=GELSED=-256+G
1950 G=ABS(D):GOSUB480
1960 SD%=SGN(D):DD$="("+J$+MID$(D1$,SD%+2,1)+Q1+"H)"
1970 GOSUB1290:I$=",":GOSUB1900:IFJ=0THENI$=" ":GOSUB1900
1980 M=LEFT$(M,J)+DD$:RETURN
1990 '***SPECIAL "LD  (IX+D),N" AND "LD (IY+D),N" INSTRUCTION
2000 GOSUB1930:GOSUB480:M="LD "+DD$+","+Q1+"H":RETURN
```

CAPTAIN 80

Bob Liddil

Now what do you suppose Captain Eighty is doing in the 80-U.S. Journal? For those TRS-80 enthusiasts who subscribe to both 80-U.S. and 80 Microcomputing, an explanation is in order. Allow me to introduce myself.

I am Captain Eighty. Software Super Hero and man of disguises. I also act as Woodfern and Boldword, investigative reporters, and the Software Secret Agent. For the first six months of 1980 I authored a combination software review and general commentary on the TRS-80 support industry through a slick and well oiled magazine called 80-Microcomputing.

Now, through the kindness of Mike and the staff at 80-U.S., I will be allowed to express my opinions and observations in a new environment. Naturally, the opinions expressed will be mine, and not necessarily those of the 80-U.S. Journal.

And now, The Software Secret Agent digs into a huge sack of programs submitted for review.

Without a doubt the world of ADVENTURE has been rocked by the new Scott Adams offering, Ghost Town. Scott has really gotten down to business with this, his ninth in a series of microizations of the popular mainframe cult rage. GHOST TOWN deals, as the name suggests, with a western theme. No history is implied or needed.

The location descriptions are vivid. The puzzle scenerio is tough but not impossible, and this one upstages all but one or two of Al's earlier ADVENTURES. Style, Wit, and ease of delivery, Scott's trademarks, are more pronounced here than ever before. ADVENTURE cultists will recieve a much anticipated fix, and newcomers to ADVENTURE will delight in the treat that Adams, the acknowledged Grandmaster of Compu-novels, has produced.

MYSTERY MANSION, an ADVENTURE WORLD production, is Greg Hassett's newest entry into the highly competitive ADVENTURE market. Mansion is a high speed, machine code program that incorporates all the features ADVENTURE lovers expect, blinking cursor, 'HELP', 'SAVE GAME', plus some plot twists that can leave the player chagrined with their simplicity. Though a bit thin in the storyline department, MANSION leaves the player with the feeling of having been entertained whether he wins or loses the game. The ADVENTURE WORLD programs are not so widely distributed as ADVENTURE INTERNATIONAL so you might have to ask for MYSTERY MANSION.

From a tiny little company in WARREN MICHIGAN called SIMULATION SOFTWARE, comes an entry into the densly populated space gaming market. Called COSMIC TRADER, it incorporates some features I really like. Usability is one. Nothing gripes me off any more than plunking out fifteen dollars for a program only to have to spend hours decoding unintelligable charts and number clusters. COSMIC TRADER is simple without being moronic. It allows the buying and selling of commodities on an intersteller basis with price dickering taking place on each planetfall. The bargaining routine brings back memories of pleasant hours of arguing with TEMPLE OF APSHAI'S innkeeper, though COSMIC's not quite so penny conscious. Fuel economy and pricing is a consideration, as well as pirates, and a Galactic patrol which sometimes pops up unannounced to be met with hot laser blasts from trigger happy skippers.

Up to four people can play COSMIC TRADER, a situation I put to test with three small boy baseball players who'd never seen a computer before. They understood the game within minutes and after an hour of play had thrashed me so badly I had to withdraw to save face. I could see enjoyment written all over their faces. There is no higher recommendation available for software as far as I'm concerned.

DUNGEON EXPLORER 2.0, a second entry from the fledgling SIMULATION SOFTWARE, is aimed at the virgin DUNGEONS AND DRAGONS market. This market is dominated by the beautifully documented AUTOMATED SIMULATIONS line whose DATESTONES OF RYN and MORLOCK'S TOWER so far seem pretty much unchallenged.

DUNGEON EXPLORER 2.0 does not go out of its way to provide loads of plotline or complexities. It is aimed at the Fantasy Role Player who incidently happens to be a computerist. The program will roll a combatant for you (handy) in D&D style or will accept your low level character when you input him. Once inside the dungeon, single letter commands, INKEYed, provide directions, combat, and object manipulation. All the elements of Fantasy Role Playing are provided. Magic swords, enchanted armor, chests unopened, flagons of elixer, are just some of the possibilities. Combat features a hair trigger fight or run command that can be deadly. Resurrection is never assured.

Personally, I never tire of slaying Orcs and Bugbears. One character created in 2.0 went on to a short but glorious career in other Dungeons and Dragons worlds. For its $14.95 price tag it returned a hundred dollars in playing pleasure.

The above fine examples of imaginative and well written software represent a mere drop in the bucket compared to the locally produced, locally marketed programs that never seem to find their way into the national market. It seems that Joe average programmer is content to sell fifteen copies of his work to his local STOP and SWAP computer center, or give it away free of charge in the one for one exchange market.

At a recent user's group meeting in Boston, I sat in on a discussion on computer cryptography, led by a high school youngster, that was awesome in its intellectual scope. If this prodigy is an example of America's computer youth, then stand by for the explosion. His presentation was brilliant.

I wonder how many well written and marketable programs are sitting on their author's shelf for lack of publisher, or worse, how many are sitting in envelopes in the large software houses, unreviewed, or bogged down in red tape, while the market, ever in motion, passes them by.

Getting author, publisher, and buyer together has been a problem ever since the cave man ULGH first submitted a manuscript on dinosaur broiling to the Cro-Magnon Press. But I wonder if a self publishing effort isn't sometimes more appropriate than eleven to fourteen months of waiting in limbo because some publisher didn't have the manpower to review and produce it. Or worse, be broke because some fly-by-night ripped it off.

I think the market has unlimited possibilities for someone who writes well, be he 13 or eighty nine. And with advertising prices as low as $98. for a sixth of a page ad in a major TRS-80 oriented publication, the excuse of no national market demand is no longer valid.

So come on, major producers! Get off your duffs! There are people out here that want new programs! There are stores who want to stock them! Don't let your submissions die on the vine. Let's go authors! If you produce and advertise a few on your own, you'll find your name pulled out of the pile of incoming mail. And after all, isn't meeting the public's demand for quality software what it's all about? ●

BUSINESS SOFTWARE

FROM COMPUTER SYSTEMS DESIGN INC
FOR MODEL I AND MODEL II

OPEN ITEM ACCOUNTS RECEIVABLE $240.00
Edit - flags customer number errors, gives correction option. Sales Distribution - Distributes each department total. Customer Journal - Lists each invoice for each customer. Statements - Pre-printed or blank statements may be used.

ACCOUNTS PAYABLE $240.00
Edit - Flags vendor number errors, gives correction option. Cash Requirements (Journal) - Lists all payable vendors. Suspended Journal (Unpaid Journal) - Lists unpaid vendors. General Ledger Distribution - Distributes General Ledger totals. Check Register - You just key the first check number. Checks - Check stub includes information on invoice date, gross and discount amounts. Complete with check protection and aged payables.

PAYROLL $240.00
Edit - Flags employee number errors, gives correction option. Journal - Year-to-date totals for gross, W/H, FICA and vacation. Deduction Summary - Totals of individual deductions. Deduction Register - By employee deduction type. Check Register - Check stub includes year-to-date information for employee W-2 forms. 941 Quarterly Reports - Federal $, State $, FICA $, Quarter $, Hours.

GENERAL LEDGER (interactive with A/P) $240.00
Edit - Flags account number errors, gives correction option. Summary Ledger - Month's debits and credits. General Ledger - Complete detail list. Income statement - Up to four levels of totals. Balance Sheet - 1 or 2 page control, print control. Department Statements - Maximum of 99 departments. Schedules - Supporting lists of any detail accounts. Input Sheet - Working trial balance.

FROM GALACTIC SOFTWARE LTD

MAIL/FILE SYSTEM for Model I $99.00
Under constant sort, both by name and zip. Retrieve by any combination of up to 19 user codes. Format your own labels or use standard format. Supports an 11 digit alphanumeric zip code. Provision for a "message" line. Very complete user oriented documentation. 600 records per non-system disk, 300 records with system disk. Simple editing throughout. 32K and one or more disk drives required. Name rotate, tab listing and more!

MAIL/FILE SYSTEM for Model II $199.00
Has all the features of the Model I system above plus a machine language sort, 2500 names per disk, "word processor" type input editing. 64K single drive required.

NEW!

EDITOR-ASSEMBLER (EDAS 4.2) $229.00
Complete Zilog Editor-Assembler featuring Global search and exchange, assemble to disk or direct to memory, direct linkage with DEBUG or RSM II. Jump to memory with automatic return, and text block move. Executes SYSTEM commands and even remembers filespecs.

GAMES SOFTWARE

NEW!

@119 REPEAT AFTER ME by James Talley $9.95
Challenge your memory with this unique new game! Based on the games "Simon" and "Einstein", this new game has surperb animation and SOUND Five levels of difficutly make this game for all ages.

#118 CAT2/XFR by Don Fielding $24.95
A disk directory program that will read and organize your directories on an index disk. Available on disk only, with doucmentation, requires 32K 2 drives and NEWDOS

#107 Owl Tree by James Talley $9.95
Can you fill the Owl tree with Owls by shooting out the Bats? Easy? Careful, when you shoot a bat it scares away owls! With animated graphics and Sound.

#109 The Great Race by Scott Carpenter $9.95
Try to finish this 600 mile race before your opponents, or before they stop you with flats, wrecks etc. The computer plays too With sound.

#111 Lying Chimps by Roy Groth $9.95
The old game of "I doubt it" or "Liar", only you play with four cheating chimps. Animated graphics and sound

#113 Concentration by Richard Taylor $9.95
The game of concentration. Prizes change places every game. With excellent sound effects!

#110 Scramble by Richard Taylor $9.95
A word game for two players Use your words or the computers words. With sound and an excellent scoring routine.

#103 Snake Eggs by Leo Christopherson $14.95
This version of 21 has talking snakes who argue with each other. Try to avoid scrambled eggs, they lose!

#108 TRS-80 Opera by Richard Taylor $9.95
A sound extravaganza! Hear the William Tell overture in intricate detail and clear sound Contains four other operatic selections.

#112 Challenge by Richard Taylor $9.95
Guess the hidden phrase, but if you guess vowels wrong you lose 10 points With sound, for 2 players, use your own phrases or the comuters.

#106 Beewary by Leo Christopherson $14.95
Brilliant graphics and fantastic sound enhance this challenging game matching a persistent bee with a cunning spider in a duel to the death!

#104 Lifetwo by Leo Christopherson $14.95
Conway's game of Life at an astounding 100 generations per minute! Plus Leo's "talking" animated creatures playing the Battle of Life in one 16K L2 program

#105 Cubes by Leo Christopherson $9.95
Gives the solution to "Instant Insanity"® or numbered blocks. Watch the computer try all the possible combinations.

#102 Android Nim by Leo Christopherson $14.95
The TRS-80's first animated, and most popular, graphics game with sound.

All game software except CAT1/XFR is on 16K L2 cassette. Orders filled within week of receipt and sent postpaid, first class. Any malfunctioning program will be replaced free - no cash refunds. BEEWARY will not function properly when placed on disk using DOS 2.2 or 2.3. It will work with 2.0, 2.1 or NEWDOS.

Dealer Inquires Invited

Use the order card in this issue, VISA and MasterCard orders welcome. Call (206) 475-2219.

80-U.S.SOFTWARE

3838 South Warner Street Tacoma, WA 98409

Toll-free order no.:
1-800-527-4196†

LOOK!!

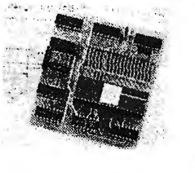
TRS-80*Owners Save on Equipment & Software!

Percom Mini-Disk Systems

These Percom mini-disk systems store more data, are more reliable. Access times are fastest possible with your Expansion Interface. Heavy duty power supplies run cooler, last longer. Low noise three-wire ac power cord is safer. Enclosures are finished in compatible silver enamel. Prices:

TFD-100™ (40-track, 102 Kbytes/side)	
One-Drive Add-On	$ 349.95
Two-Drive Add-On	699.45
Three-Drive Add-On.............	1049.95
TFD-200™ (77-track, 197 Kbytes)	
One-Drive Add-On	634.95
Two-Drive Add-On	1268.95
Three-Drive Add-On.............	1903.95

Price includes Percom upgrade PATCHPAK™ program.



DATA SEPARATOR™

This PC board plug-in adapter for the TRS-80* virtually eliminates data read errors (CRC error — Track locked out!) which occur on high density inner disk tracks, a problem that has plagued TRS-80* systems. The Percom Data Separator™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty: $29.95.

Percom OS-80™

An advanced easy-to-use disk operating system that works with Level II BASIC commands. Resides in only 7-Kbyte of memory. May be extended indefinitely with disk-resident utilities. Supplied on 5" disk with example programs: $29.95 with instructions.

CIRCLE J Software

Two extremely useful utilities for Percom's OS-80™ DOS:

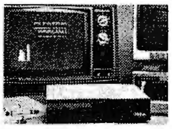
1. Machine Language Save/Load Utility. On 5" disk with bonus patch program that allows RS Renumber Utility to run under OS-80™. $14.95, with instructions.
2. VARKEEP — Adds NAME SAVE and NAME KEEP commands to OS-80™ Use one set of common data with two or more BASIC programs. Also runs under Radio Shack DOS. On 5" disk, with instructions. $14 95.

Z80ZAP

Super fast machine language disk modification utility. Read, Write, Display, and Modify sectors; remove passwords; apply patches, fixes; make backups and much more. On 5" disk with instructions: $29.95.

100% machine language word processor... SPECIAL DELIVERY (by Software Etc.)

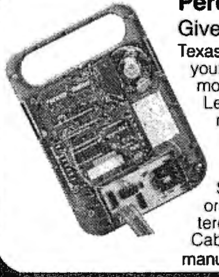
Use MAILFORM to create name and address lists; EXTRACT to find names by ZIP, address, gender, age, etc ; SORT to sort an entire list on any field in seconds Print personalized letters written with either the Electric Pencil‡ or Scripsit* using MAILRITE Prints labels from Mailfile created under MAILFORM Runs under Percom's OS-80™, Radio Shack's TRSDOS* $125 (disk)



Inexpensive Color Graphics: the Percom Electric Crayon™

Spectacular multicolor graphics, sharp 2-color alphanumerics with your TRS-80*, a color tv and the Percom Electric Crayon™. Up to eight colors. Resolution with full display memory (6 Kbytes) is 256 X 192 picture elements Microprocessor controlled, the Electric Crayon™ is not only a full color graphics system but also a complete, self-contained control computer with a dual bidirectional parallel I/O port — provision for second dual port. Interface the TRS-80* via your Expansion Interface or Printer Cable Adapter Supplied with 1 Kbyte display RAM, EGOS™ operating system and comprehensive users manual with example programs $249 95

Optional TRS-80* interconnecting cable $24 95.



Percom's Speak-2-Me-2™

Give your TRS-80* the gift of speech

Texas Instruments' Speak & Spell˜ is the voice of your TRS-80* computer with this clever interface module manufactured by Percom Your own Level II BASIC programs announce, command, implore with sentences and expressions formed from Speak & Spell's˜ vocabulary The Speak-2-Me-2™ PC module installs in the battery compartment of your Speak & Spell˜ Power is supplied from an ordinary calculator power pak. Comes with interconnecting cable (for TRS-80* EI or Printer Cable Adapter), operating software and users manual: $69 95 (Speak & Spell˜ not included)

Games People Play

Far out !

Scott Adams' Adventures (disks):	
Adventure Sample	$ 6 95
Adventure 1-9	14.95 ea
Adventure 1 & 2 .	24.95
Adventure 3, 4 & 5 . .	39 95
Adventure 6 & 7	24 95
Adventure 6, 7 & 8 .	39.95
The Galactic Trilogy (disks):	
Galactic Empire	$19.95
Galactic Trader	19.95
Galactic Revolution . .	39 95
Space is this hunter's domain (disks):	
Starfleet Orion	$24 95
Invasion Orion . .	24 95

Space Battles — from Adventure (disk):
Super fast, real-time war game with exploding missiles, full space ship control: $19.95

Space Invaders — by Level IV (disk).
TRS-80* machine language version of Apple 'Super Invaders'. Sound effects: $19.95

Far in!

Datestones of Ryn (disk)	$24 95
Temple of Apshai (disk)	29 95
Morloc's Tower (disk)	24 95

Disk System Interconnecting Cables

Improvement over RS cable design places drive 0, which includes the cable termination, at the end of the cable to eliminate the reflected noise of an unterminated cable Better data integrity Prices

Two-Drive Cable	$24 95
Four-Drive Cable	34.95

Power Line Filter

115/250 V, 50-400 Hz. Instructions included for easy installation in standard mini-box chassis: $19.95

Minidiskettes

10 Disks in a convenient plastic organizer box .	$34.90
Single Disk	3.49

Disk Drive ID Tabs

1" x 1-¼" self-adhering plastic drive identification tabs Compatible silver with engraved black drive number. Two tabs (Nos 0, 1): $2 50; three tabs (Nos 0, 1, 2) $3 25, four tabs (0, 1, 2, 3) $4 50

How to Order

Order by calling Access Unlimitod toll-free on 1-800-527-4196†.

Mail orders also accepted Orders may be charged to a VISA or Master Charge account or paid by a cashier's check, certified check or money order We accept COD orders with 25% deposit Sorry, we cannot accept personal checks. We pay shipping and insurance charges on orders over $1 000 00 Add approximate insurance and shipping charges for under $1,000 00 If in doubt about these charges, ask when you call in your order Texas residents include 5% sales tax Minimum order $20 00 Allow 2 to 4 weeks for delivery

†Texas residents call (214) 494-0206

ACCESS UNLIMITED

315 N. Shiloh · Ste. D1 · Garland, TX 75042
(214) 494-0206

Prices and specifications subject to change without notice

™ trademark of Percom Data Company, Inc
* RADIO SHACK and TRS-80 are trademarks of Tandy Corporation
˜ trademark of Texas Instruments Corporation
‡ trademark of Michael Shrayer Software, Inc.

NOTES

DISABLE PASSWORDS ON MODEL II

There comes a time now and then when it is necessary or at least desirable to get around the passwords on a disk. NEWDOS had done this for Model I, but nothing has been done for Model II. Now, here is a way to do it for your Model II.

Since this fix consists of four patches, we recommend you build a "DO" file (we chose the name FIX) which consists of the following:

```
PATCH  SYS1/SYS  A=262D,  F=28,  C=18
PATCH  SYS4/SYS  A=221F,  F=20,  C=18
PATCH  SYS4/SYS  A=22B0,  F=28,  C=18
PATCH  SYS5/SYS  A=24C1,  F=28,  C=18
```

When you DO this file, these four patches will (as far as we can determine) completely disable all password checks in TRSDOS 1.2. This is true for OPEN, KILL and for any level of password protection.

PEEK AND POKE FOR MODEL II

For those of you who haven't noticed yet, Basic on the Model II is almost the same as that on the Model I. But why no PEEK or POKE? Here is a way to fully implement both PEEK and POKE.

First, a couple of warnings: There are a total of eight patches required for this mod. All eight *must* be entered, or the results can be disasterous. Use a backup disk, and neatness counts. This modification replaces the OCT$ and NAME commands in Basic, and if for some reason you need either of these commands, you should not perform these patches.

1. Make a fresh backup copy of TRSDOS 1.2
2. Execute the "Disable Password on Mod II" (see the preceding note).
3. BUILD (see pages 2/13 - 2/15 of your TRSDOS manual) a file which reads as follows:

```
PATCH BASIC A=6771 F=C5CD2061 C=CD903CD5
PATCH BASIC A=6775 F=C5CD0C66 C=E72CCD9D
PATCH BASIC A=6779 F=E741E753 C=3CD112C9
PATCH BASIC A=28FB F=CE414D   C=D04F4B
PATCH BASIC A=5A77 F=CD3D4E   C=C37D67
PATCH BASIC A=677D F=E3011E00 C=CD0A447E
PATCH BASIC A=6781 F=09444D   C=C3A93A
PATCH BASIC A=2A05 F=CF435424 C=D045454B
```

4. When you are satisfied that there are no errors in your file, "DO" that file (see pages 2/38 - 2/39 of your manual). You should see "PATCH IS DONE" eight times. If for any reason you see any kind of error message, make note of which one of the patches caused it (i.e., was it the first one, the fifth, etc.). Any error will cause that patch not to be executed, and that one will have to be entered by hand from TRSDOS READY.

Now, PEEK and POKE should be a permanent part of your Basic. The syntax for the PEEK command is:

*operator*PEEK*(argument)*

Where *operator* may be a variable = statement, PRINT statement, etc. *Argument* is the address you want to PEEK at, and it may be a numeric constant, hex constant, numeric variable or formula and must be enclosed in parenthesis.

If the address you want to PEEK at is greater than 32767 (7FFF hex), the address must be computed by the formula:

-1*(65536-*address*)

Thus, address 7FFF hex may be looked at by using PRINT PEEK(-32768). Note that address FFFF hex translates to -1 using this method. Using a hex constant such as &H8000 bypasses all this.

EXAMPLES OF PEEK

```
A=PEEK(J):     PRINT PEEK(&H5000):     B=A+PEEK(-45)
```

The syntax for POKE is:

POKE*argument,operator*

Where *argument* is the address desired, and is developed exactly the same as the for PEEK, except that parenthesis are not required. *Operator* may be either a numeric constant, variable or expression, so long as it is in the range 0-255 (0-FF Hex); and must be separated from *argument* by a comma.

EXAMPLES OF POKE

```
POKE A,B     POKE 14938,22     POKE &H900,A%
POKE PEEK(A+1)*256+PEEK(A),ASC(MID$(TR$,G%,1))
```

Using PEEK and POKE carefully allows the use of such techniques as string packing, program chaining without losing variables and many other useful functions.

FIELDing Made easy for Mod I or II

Random file processing is often a real problem since each variable in a field must be individually identified and set up. Most programs do this with a long list of FIELD statements that allocate space to them one at a time. Have you ever wished you could do this with a FOR-NEXT loop? You can, consider this: We have 30 identical variables in an array, each of which needs to be fielded with four spaces in the data file. If the array is A$(1) through A$(30), you could use:

FIELD #1,4ASA$(1),4ASA$(2),4ASA$(3),....and so on until all 30 items were listed. Wouldn't it be much simpler to use:

FORI%=1TO30:FIELD#1,(I%-1)*4AS DUMMY$, 4ASA$(I%):NEXTI% Not only is it easier, it works.

SCREEN DISPLAYS for Model I or II

Screen displays are what people see most of when using a program. If the display is amateurish, the user will think the program is also amateurish. To dress up a request for user input it is handy to have the screen indicate the size of the field (how many characters are allowed). One nice way is to define two strings: **DT$=STRING$(X,160):BL$=STRING$(X,28)** The first string will put a string of X graphics blocks on the screen (160 is the lower left hand block on the Mod I and a blank on the Mod II). The second string backspaces the cursor X spaces without erasing.

To set up for entry, simply position the cursor where you want it with a PRINT@, PRINTTAB or whatever, then use: **PRINT DT$;BL$;** (the semicolons are important!). This statement will print the string of graphics blocks and then backspace over them without erasing them. You can use some routine to input the desired information assured that your allowed data field is well marked.

Some notes on this technique. First, it is not necessary to define several strings of different lengths for different length fields. Just use the MID$ command to print whatever length is desired. Also if you use the INPUT command to get the data in, the computer will print a question mark and a space prior to inputting the data. This will have to be allowed for. Also, 160 is not the only character you can use, any valid character will work as well.

TRS-80 TRS-80

SIMUTEK COMPUTER PRODUCTS

THE COMPANY THAT SELLS IT'S FANTASTIC SOFTWARE ALL OVER THE WORLD, NOW BRINGS YOU GREAT BARGAINS ON TRS-80 COMPUTERS AND PERIPHERALS

Disk Drives ready to plug in and fully compatable with TRSDOS or NEWDOS. Full 90 day warranty!!!!!!

All Drives fully tested and ready to run

MPI Single Drive 365.00
MPI's can operate 8 times faster than Radio Shack's and will handle 35 or 40 tracks!!!

Percom Single Drive 379.95
Percoms handle 35 or 40 tracks plus you may use both sides of a diskette!! Best buy!! Save $$ on expensive Diskettes!!!

MPI Dual Drive 565.00
This is like having 2 drives yet uses only one Diskette!! Use one side as drive O, the other side as drive one!! (Save $435 compared to 2 RS Drives!!) Save $$ on expensive Diskettes too!!

4 Drive Cable	39.95
2 Drive Cable	24.95
10 PK Diskettes	39.95
NEWDOS+	99.95
TRSDOS 2.3 + Manual	19.95
TRSDOS Manual Only	4.95

Microsofts Basic Compiler — Special — 179.95

Requires 48K, at least 1 Disk Drive. 2 Disks preferred. (Regular price 199.95 — Save $20.00!!)

ALL TRS-80 COMPUTER PRODUCTS 10 — 15% DISCOUNTS!!!

FULL RADIO SHACK WARRANTY ON ALL TRS-80 PRODUCTS!!!!!

Includes 4K LV 1's, 16K LV II's, Expansion INT'S, Model II's, Disk DR's, Line Filters, RS Software, Printers, Paper, Computer Books, and any other Radio Shack products!

TRS-80 16K MEMORY CHIPS — ONLY 69.95!!

Instruction & Jumpers for Keyboard 2.50 extra. Free Shipping.

SIMUTEK'S MICRO SPEED

Upgrade your slow TRS-80 (1.77 MZ) to a 2.66 MZ machine!!!! Easy to install! Shuts off automatically during disk or cassette I/O. By far the best on the market! Only $29.95. Free Shipping.

SIMUTEK FANTASTIC GAMES PACKAGES!!!

Five (5) Great Programs in Each Package!!!

Package 1	Space/Wars	**Package 6**	Finance/Amort. Loan+
Package 2	Board/Artfi	**Package 7**	Backgammon/Speed Read/Yahtzee
Package 3	Educ. Artist		
Package 4	Life/Lander	**Package 8**	Pinball W/Sound!
Package 5	Horse Race/Ameba Kill!		

- **Packages come with complete instruction books!** •
- **Packages on cassette 14.95 packages on disk 19.25** •
- **Free Shipping on Simutek Software Packages!!!** •

WE ACCEPT VISA, MASTER CHARGE, MONEY ORDERS, PERSONAL CHECKS, (C.O.D. $5.00 Extra)

Send Orders To: SIMUTEK
P.O. Box 13687
Tucson, AZ 85732

Name ______________________
Address ______________________
City ____________ State ______ Zip ______
Order ______________________ C.C. # ____________ Exp. ________

24 HR. HOTLINE (CREDIT CARD/C.O.D.)
(602) 886-5880

Technical Questions or Daytime Orders
(602) 885-7026

Shipping and Handling $15 extra on all drives, printers, computers, expansion int's. All other shipping $5 extra.

TRS-80 is a trademark of Radio Shack, a Tandy Corporation. No charge for shipping Simutek Software!!!!

Programming Languages

T R Dettmann

Pascal

another look

About 12 years ago, Professor Nicklaus Wirth at ETH in Switzerland (*That's the Eidgenossische Technishe Hochschule in Zurich, Switzerland*) became dissatisfied with programming languages for teaching.

FORTRAN was the major language being taught at the time. FORTRAN was, and still is, a versatile and useful language. But, it is easy to develop bad habits that make programs hard to write and debug. Also, because of it's flexibility, FORTRAN could be written very haphazardly. This created a whole generation of programmers who had to muddle through without really knowing how to program well. BASIC has the same disadvantages.

In order to teach logical program design and construction, Professor Wirth used his experience with the language ALGOL, then very popular in Europe, to write a new language aimed specifically at teaching. That language was PASCAL.

Practicing programmers have avoided PASCAL for years since they were already familiar with other languages and did not like some of it's features. But PASCAL found believers in the universities where it was used primarily for teaching.

Experience showed that students who learned PASCAL first, had little trouble learning to program in FORTRAN, BASIC or other languages, since they had learned correct logical techniques. Students who learned another language first had some trouble learning PASCAL since they had to "unlearn" bad habits. True or not, I once heard that Professor Wirth, who is still teaching, refuses to take graduate students who have learned to program in FORTRAN or BASIC since they have been damaged by their backgrounds.

As knowledge of PASCAL spread slowly out of the university community, interest began to increase. A PASCAL user group was started and now has its headquarters at the University of Minnesota Computing Center. The University of California developed what has come to be known as "USCD PASCAL", a special one user form of PASCAL. Special interest groups have also started to foster the use of PASCAL. No accurate figure is available on just how many systems have PASCAL available, but at least 60 different systems can run it and that number grows daily. One magazine article called it the "BASIC of the future" and "PL/I, done right". Let's see how PASCAL works, and see if we agree.

PROGRAMMING IN PASCAL

Programming in PASCAL is different from what you have learned in BASIC. The language forces good habits and gives you capabilities that BASIC just does not have. Let's look at how we can write programs in BASIC and PASCAL to do the same thing in the same way.

Consider this simple Level II program in BASIC:

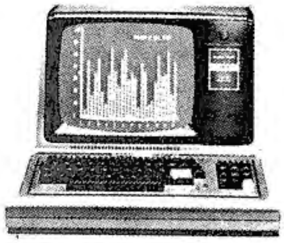
```
10 REM IS YOUR NAME TERRY?
20 PRINT "WHAT IS YOUR FIRST NAME";:INPUT
   A$
30 IFA$="TERRY" THEN 40 ELSE 60
40 PRINT "HOORAY TERRY, YOU HAVE A NICE
   NAME"
50 STOP
60 PRINT "I'M SORRY ";A$
70 PRINT "YOUR PARENTS GAVE YOU THE
   WRONG NAME"
80 STOP
```

This simple program uses string handling, input and output, and decision making in it's eight lines. How would this be written in PASCAL?

```
  PROGRAM WHATSYOURNAME;
    (* IS YOUR NAME TERRY? *)
    VAR     NAME: STRING;
BEGIN (* PROGRAM *)
    WRITE ('WHAT IS YOUR FIRST NAME?');
    READLN(NAME)
    IF NAME = 'TERRY' THEN
      BEGIN(* THEN CLAUSE *);
```

(Continued on page 42)

AUTHORIZED TRS-80® DEALER A301

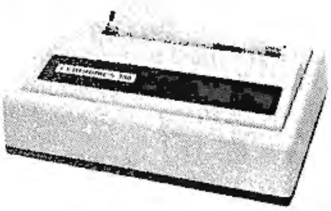


COMPUTER SPECIALISTS

10% DISCOUNT Off List
26-4002
64K 1 Drive
$3499.00

26-1056 16K Level II System with Keypad	$ 700.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	254.00
26-1141 "16" K Interface	381.00
26-1142 "32" K Interface	508.00
26-1160 Mini Disk - Drive 0	424.00
26-1161 Mini Disk - Additional	424.00
26-1154 Lineprinter II	720.00
26-1156 Lineprinter III	1799.00
26-1157 WP50 Daisy Printer	2670.00
26-1180 Voice Synthesiser	339.00
26-1181 VOXBOX	145.00
26-1104 Factory Upper/Lower Case Modifaction Installed	90.00
26-1605 Scripsit - Tape	60.00
26-1563 Scripsit - Disk	85.00

15% DISCOUNT Off List
26-1054
4K Level II
$527.00



ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES DEDUCT 10% FROM CATALOG PRICE

LOBO drives INTERNATIONAL

Plug Compatible Lobo 5¼ Drives - $375.00

Versatile Lobo Interface, 8" Drives and Hard Drive.
Call For Prices

CENTRONICS

Fast 100 CPS Centronics 730 Printer - $675.00
Text Quality Centronics 737 Printer - $850.00

Novation Cat Modem	$159.00
CCA Data Management System	72.00
Adventure Games Games 1-9	14.00

MICROSOFT

Model I Basic Compiler	$195.00
Model II Basic Compiler	395.00

BASF

10-5¼ Diskettes	$45.00
10-8 Diskettes	47.00

Acorn Software Products, Inc.

GAMES:	
Alien Invasion	$9.00
UTILITIES:	
System Savers	14.00
EDUCATION:	
Language Teacher	18.00

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock.

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

Full Factory Warranty on All Items Sold.

Largest Inventory In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

```
      WRITE ('HOORAY TERRY,');
      WRITELN('YOU HAVE A NICE NAME')
    END (* THEN CLAUSE *)
  ELSE
    BEGIN (* ELSE CLAUSE *)
      WRITE('I'M SORRY');WRITE(NAME);
      WRITE('YOUR PARENTS GAVE YOU');
      WRITELN('THE WRONG NAME')
      END (* ELSE CLAUSE *)
END. (* PROGRAM *)
```

Several things stand out in the two programs. First, the PASCAL program is longer. This is somewhat artifical because of the way the program is generated. Two, the logical structure of the PASCAL program is laid out by the way the program is indented. BASIC could be indented, but look at most programs and you will find they are not.

From the programs above, we can see some natural similarities. First, it is easy to see that 'PRINT' in BASIC is the same as 'WRITE' in PASCAL. Next, 'INPUT' in BASIC is the same as 'READ' in PASCAL. There are some special features to notice. "WRITE(A)" in PASCAL is the same as "PRINT A;" in BASIC (the semicolons after each PASCAL statement are separators like the ENTER at the end of a line in BASIC). To get the same effect as "PRINT A" (prints the value of A and goes to the beginning of the next line), PASCAL uses the extra characters "LN", for example, WRITELN(A), which I always think of as "write line". Read works the same way.

IF-THEN-ELSE statements are more interesting. In BASIC, we usually skip to another line to handle our choice because we can't get any more than 255 characters on a single line. In PASCAL, we are not restricted to a single line for anything. PASCAL treats everything between a "BEGIN" and and "END" as one statement. Think of the "BEGIN" and "END" as left and right parentheses that work for program statements.

With BEGIN-END pairs, we can put blocks of statements together and place them anywhere in a program where a single statement is legal. This means we do not have to jump around the program to do several steps. You can also see that since a PASCAL statement's end is marked by a semicolon, it is not necessary to keep statements on a single line. You can put a PASCAL statement on several lines to make it more readable.

Two statements which are probably a bit different from any you have seen are "PROGRAM" and "VAR". The "PROGRAM" statement is used to identify the program by name. In this case, I have called my program by a long name just to show that such a name is legal.

The "VAR" statement is for variable definition. Unlike BASIC, there are no predefined variable types. You have to identify the names of any variables used in the program and define their type such as integer, string, or real with a "VAR" statement. This may seem to be a bother at first, since in BASIC you do not even have to worry about the variables until you actually use them. But this is one of the best features of PASCAL. By insisting that you identify your variables, you have to decide what variables and types you will use and list them.

Have you ever tried to figure out what variables were being used in a complex program so you could add something to it? PASCAL make it easy. PASCAL also checks the program to make sure you do not do stupid things like setting apples equal to oranges.

Speaking of apples and oranges, the "VAR" command lets you define apples and oranges in your program! The "VAR" statement:

```
VAR     FRUIT (APPLE,ORANGE,PEAR);
```

defines the variable FRUIT and lets it take on only three possible values, namely; apple, orange or pear. You can then use this definition in your program, for example, you can test:

```
IF FRUIT ='APPLE' THEN ....
```

The ability to set up variables directly in terms of the problem you are working on makes PASCAL more useful than any other language now available.

PASCAL has some very unique advantages that make you wonder why you ever took up BASIC to start with. More important, you can develop PASCAL programs as independent units.

In BASIC, you have to keep track of your variables throughout your program and be careful to keep the first two letters of every variable different. In a very complex program, you could generate a hundred or more variables this way. You might also be forced to use other than the best variable name for the job because something too similar was already used.

This shows up even more when you try to modify someone's program. First you have to find out what their variables are and then use them only in ways that do not conflict with the rest of the program. This can be nearly impossible at times.

In PASCAL, when you develop a subroutine (called either a function or a procedure), it is totally independent of the rest of the program to whatever extent you like. If you define a variable called "X" in the main program, it is available to all procedures called by the main program, unless that procedure defines it's own variable "X". So you can put together procedures now and not worry about whether the variables conflict, they will not if all of the variables are defined within a procedure.

To pass values to a procedure, you are allowed to put them as arguments in the statement which calls the procedure. For example:

```
INVERT(MATRIX)
```

Invert is a procedure which has been defined for the program and MATRIX is a variable array of values that has been defined in the program which calls INVERT.

SUMMARY

This only scratches the surface of what PASCAL can do. It has every capability that BASIC has and more. Procedures, functions, structured programming, long variable names, and so many other features that a full manual would have to written to describe them all.

If the idea of PASCAL wets your appetite, you could use someone else's. In particular, if you have access to a larger computer system, many of them now feature PASCAL. You could buy either the full system, or a "Mini-PASCAL" system.

Would I recommend PASCAL as a programming language? Definitely! It is not hard to learn and it teaches you good habits. I find that since I started programming in it, I now do much more of my program design in it, no matter what language the system will be written in. I do it because my ideas are more easily expressed in PASCAL than in BASIC or FORTRAN. I even do my initial design for Assembly routines in my own pseudo-PASCAL.

Learning PASCAL is an advantage for today and it may be essential in the future, as systems get larger and larger. PASCAL or languages like it will probably dominate the programming field in the years to come.

System/Command

The 11th in a series on Machine Language Applications

Phil Pilgrim

A Split Personality for TRS-80

Picture this: You're deeply involved in a game of Flaming Starcruisers. You're about to blast another Thingon Buzz-bomber to Kingdom Come when the phone rings. "Hey, kid, this's your Uncle Ralph. Whatcha doin'?"

"Well, I. . ."

"Good! I'm glad you're not busy. Listen, I just got this great line on a new house. 'Ya still got that mortgage loan program?"

"Uh-huh.. "

"Great! 'Just wanna make a few calculations. Aunt Edith and I will be over in a few minutes. Bye!"

"(Expletive deleted.)", you mutter, walk over to the machine, and hit (BREAK). If only you could afford TWO computers, things like this wouldn't be a problem, huh? Well, there is another solution: The No-Hardware Second Computer. The principle is this: In BASIC there is an area in RAM containing pointers and sundry other information defining BASIC's entire computational environment. Combined with the screen contents and stack pointer, they are all you need to know to resume execution of a program where you left off, assuming the program and its variables are left intact. By saving this information and setting up another environment for using BASIC in a different part of memory, you can have two programs resident at once and switch between them just by swapping environments between BASIC RAM and a save area.

The program shown here, SPLIT, does just that. Once activated, it gives you entirely independent program areas in memory and the means to move between them. If you're involved in a long computation and need to run a short program, hold down the SHIFT key, hit (BREAK), and the "other BASIC" is ready. Do what you have to do, hit (SHIFT-BREAK) again, and the first program resumes execution where it left off, with the original screen contents restored. You can do anything in either BASIC you wish -- including CLOAD's and CSAVE's -- without affecting the other (be careful with POKE's, of course). The only catch is that available memory for each is reduced considerably.

What's happening here? To understand this we need to look at a typical BASIC memory map with a large chunk of memory reserved with MEMORY SIZE. (see Figure 1) Location 40A4 points to the beginning of the BASIC program area, and location 40B1 points one byte below protected memory. By changing these two pointers to point into protected memory and reinitializing BASIC, we can define the alternate environment therein. Figure 2a shows the situation with SPLIT activated and BASIC in environment one; Figure 2b, environment two. Whenever (SHIFT-BREAK) is keyed, the environments are swapped.

To use SPLIT, key it into the EDTASM* as shown. 32 or 48K users should use MEM32K or MEM48K, respectively, in the ORG pseudoinstruction. The comments give the decimal equivalents of each ORG. Disk users, of course, should change the END to END 402DH. Make an object tape and power up BASIC. Disk users should have the program on disk as a /CMD file. Load the program, and then load BASIC. In either case, set MEMORY SIZE to whatever size you need for the "first BASIC". Anything left over that isn't actually used by the SPLIT program itself will be allocated to the "second BASIC". Be sure to use a number at least 300 bytes less than the decimal equivalent of your ORG. A few thou might be better, but it depends on how much room you will need in the second environment or how much you can spare from the first. For 48K NEWDOS users, a MEMORY SIZE of 45263 (assuming default of 3 disk files) nets a free space of 18386 for both "BASIC1" and "BASIC2". Level II users should now load their SYSTEM tape and hit "/(ENTER)" to activate it (Disk Users hit "/" and the starting address of their particular memory size). The screen will flicker, and BASIC will again be READY. -- BASIC number one that is.

During that flicker quite a lot happens. First, beginning with SPLIT, the screen, BASIC RAM, and the stack pointer are saved in the save area. Next, the screen is cleared and location 40A4H is changed to point into the beginning of reserved memory. Meanwhile, IKBD is temporarily inserted into the keyboard calling sequence. Finally, BASIC number two is initialized with a new memory size by the sequence of lines 350-400 of the source. First 40B1H is loaded with IKBD-1 (this actually sets the memory size), next, the DE is loaded with the hexidecimal equivalent of negative 50, and this value is added to the HL. This new value is loaded into address 40A0H, which in effect CLEARs the customary 50 bytes of string space. The CALL to 1B4DH sets up other pointers such as stack pointer, end of BASIC text, etc. Then, BASIC, having initialized itself, calls the keyboard routine, looking for user input. But IKBD responds. First it plugs SKBD, the switching routine, into the keyboard calling sequence, then swaps environments by a CALL to CHANGE. Finally it returns to BASIC number one by a JP into BASIC's reset sequence, 06CCH, thus completing the setup.

From this point, SKBD will examine everything keyed in and pass it on -- except (SHIFT-BREAK). Encountering this, it CALLs CHANGE and returns with a zero. But because CHANGE swaps the stack pointer, SKBD is actually returning from the previous call to IKBD -- that is, back to BASIC number two. A subsequent SHIFT-BREAK will swap environments again and return from the call to SKBD back to BASIC number one. And so it goes, back and forth. One additional note on CHANGE: Because one or the other BASIC might be using the display in the 32-character mode, it is necessary to restore this, too. Luckily BASIC keeps track of what mode it's in with the byte at 403DH. Outputting this byte to port FF does the trick.

There you have it; a "second computer" in your garage for the price of a magazine. Perhaps some enterprising person will figure out a way to use the interrupts in DOS to actually make BASIC run two or even three programs at the same time. But for now, get busy and help Uncle Ralph with those mortgage calculations.

*Special note to T-BUG users. Note the use of the pseudo-op "DEFS". This can be thought of to mean "DEFine Space". Nothing actually gets put there, but it does set aside the memory space for later storage. In this case, the 3 areas that are set aside by the "DEFS" command are VSAVE, which stores a copy of the VIDEO RAM; BSAVE, which stores the BASIC pointers for memory limits, etc.; and SPSAVE, which stores the stack pointer. In most cases (this one included), it won't make too much difference to the T-BUG'er, but it is something to be aware of.

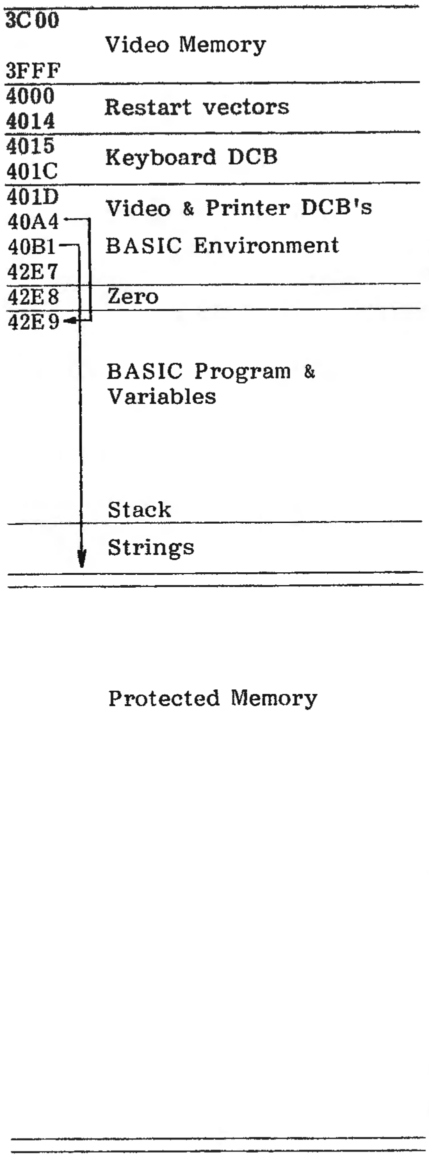


Figure 1: Regular BASIC Memory Map

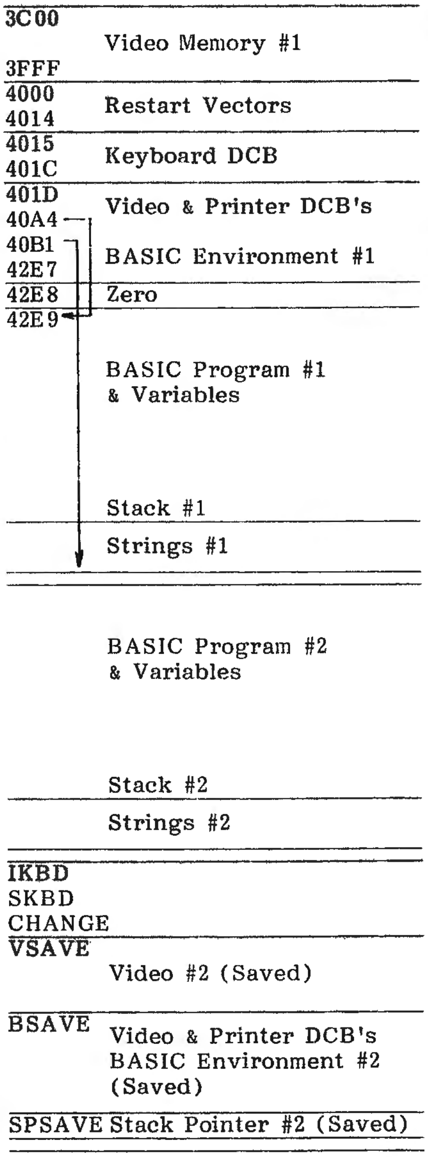


Figure 2a: BASIC Memory Map for Program #1

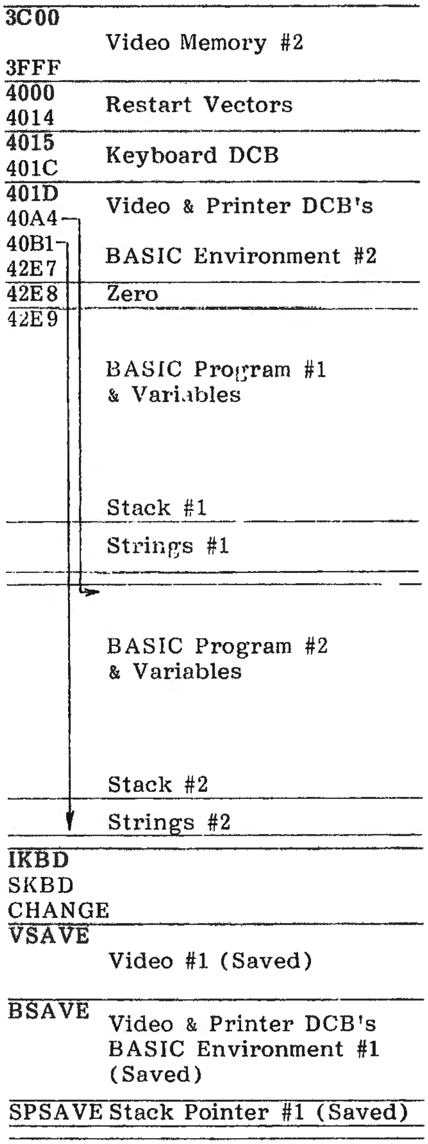
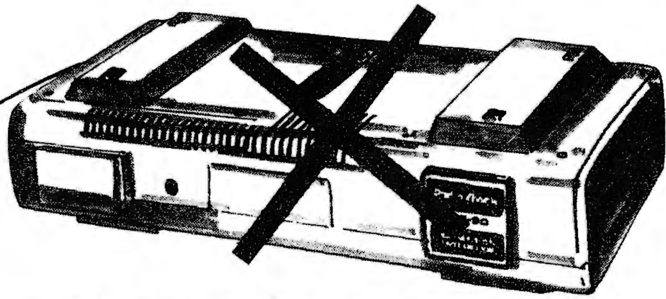


Figure 2b: BASIC Memory Map for Program #2

```
78A4            00100 MEM16K  EQU    78A4H           ;30884
B8A4            00110 MEM32K  EQU    0B8A4H          ;47268
F8A4            00120 MEM48K  EQU    0F8A4H          ;63652
4016            00130 KBD     EQU    4016H           ;KEYBOARD DEVICE ADDR.
78A4            00140         ORG    MEM16K          ;ORIGIN FOR 16K MEMORY
78A4 21003C     00150 SPLIT   LD     HL,3C00H        ;GET START OF SCREEN ADDR.
78A7 113379     00160         LD     DE,VSAVE        ;GET ADDR. OF SAVE AREA
78AA 010004     00170         LD     BC,1024         ;AND NO. OF BYTES
78AD EDB0       00180         LDIR                   ;SAVE SCREEN
78AF 211D40     00190         LD     HL,401DH        ;1ST ADDR. OF BASIC RAM
78B2 01CB02     00200         LD     BC,42E8H-401DH  ;NO. OF BYTES
78B5 EDB0       00210         LDIR                   ;SAVE IT, TOO
78B7 ED73FE7F   00220         LD     (SPSAVE),SP     ;SAVE STACK POINTER
78BB CDC901     00250         CALL   01C9H           ;CLEAR SCREEN
78BE 2AB140     00260         LD     HL,(40B1H)      ;GET MEMORY SIZE
78C1 23         00270         INC    HL              ;+1
78C2 3600       00280         LD     (HL),0          ;MUST BE ZERO
78C4 23         00290         INC    HL              ;+1
78C5 22A440     00300         LD     (40A4H),HL      ;SAVE AS PROGRAM POINTER
78C8 2A1640     00310         LD     HL,(KBD)        ;GET OLD KBD DEVICE ADDR.
78CB 22F278     00320         LD     (SKBD+1),HL     ;PLUG IT INTO $-$
```

```
78CE 21E578   00330        LD     HL,IKBD        ;ADDR. OF INITIAL KBD
78D1 221640   00340        LD     (KBD),HL       ;SAVE AS NEW KBD DEVICE ADDR
78D4 2B       00350        DEC    HL             ;-1
78D5 22B140   00360        LD     (40B1H),HL     ;ESTABLISH NEW MEM SIZE
78D8 11CEFF   00370        LD     DE,0FFCEH      ;-50 DECIMAL
78DB 19       00380        ADD    HL,DE          ;ADD (SAME AS SUB 50)
78DC 22A040   00390        LD     (40A0H),HL     ;EST. 50 BYTES STRING SPACE
78DF CD4D1B   00400        CALL   1B4DH          ;SET UP OTHER POINTERS
78E2 C3CC06   00410        JP     06CCH          ;RE-ENTER BASIC
78E5 21F178   00420 IKBD   LD     HL,SKBD        ;CALLED JUST ONCE
78E8 221640   00430        LD     (KBD),HL       ;FINAL NEW KBD DEVICE ADDR.
78EB CD0479   00440        CALL   CHANGE         ;SWITCH BACK TO OLD BASIC
78EE C3CC06   00450        JP     06CCH          ;"RESET" OLD BASIC
78F1 CD0000   00460 SKBD   CALL   $-$            ;CALL OLD KBD ROUTINE
78F4 FE01     00470        CP     01             ;BREAK KEY HIT?
78F6 C0       00480        RET    NZ             ;  NO: JUST RETURN
78F7 3A8038   00490        LD     A,(3880H)      ;SHIFT KEY DOWN?
78FA E601     00500        AND    01             ;(IF USER HAS CONTROL KEY)
78FC EE01     00510        XOR    01             ;(REVERSE BIT)
78FE C0       00520        RET    NZ             ;  NO: RETURN A 1 FOR BREAK
78FF CD0479   00530        CALL   CHANGE         ;  YES: CHANGE ENVIRONMENT
7902 AF       00540        XOR    A              ;AND RETURN A ZERO
7903 C9       00550        RET                   ;
7904 21003C   00560 CHANGE LD     HL,3C00H       ;GET VIDEO ADDR.
7907 113379   00570        LD     DE,VSAVE       ;AND SAVE AREA ADDR.
790A 010004   00580        LD     BC,1024        ;AND BYTE COUNT
790D CD2979   00590        CALL   SWAP           ;SWITCH 'EM
7910 211D40   00600        LD     HL,401DH       ;GET BASIC RAM ADDR.
7913 01CB02   00610        LD     BC,42E8H-401DH ;AND BYTE COUNT
7916 CD2979   00620        CALL   SWAP           ;SWITCH 'EM
7919 C1       00630        POP    BC             ;POP RETURN ADDR.
791A 2AFE7F   00640        LD     HL,(SPSAVE)    ;GET OLD STACK POINTER
791D ED73FE7F 00650        LD     (SPSAVE),SP    ;SAVE NEW SP
7921 F9       00660        LD     SP,HL          ;RESTORE
7922 3A3D40   00700        LD     A,(403DH)      ;GET 64/32 CHAR. FLAG
7925 D3FF     00710        OUT    (0FFH),A       ;FIX UP SCREEN
7927 C5       00720        PUSH   BC             ;RETURN ADDRESS ONTO STACK
7928 C9       00730        RET                   ;AND RETURN
7929 1A       00740 SWAP   LD     A,(DE)         ;A=(DE)
792A EDA0     00750        LDI               ;(DE)=(HL):HL=HL+1:DE=DE+1:BC=BC-1
792C 2B       00760        DEC    HL             ;GET OLD HL BACK
792D 77       00770        LD     (HL),A         ;(HL)=A
792E 23       00780        INC    HL             ;NOW OK TO INCREMENT
792F EA2979   00790        JP     PE,SWAP        ;LOOP BACK IF BC NOT EQUAL 0
7932 C9       00800        RET                   ;ELSE RETURN
0400          00810 VSAVE  DEFS   1024           ;SCREEN SAVE AREA
02CB          00820 BSAVE  DEFS   42E8H-401DH    ;BASIC RAM SAVE AREA
0002          00830 SPSAVE DEFS   2              ;STACK POINTER SAVE AREA
7FFF          00840 MKEND  EQU    $-1      ;JUST TO CHECK END OF PROGRAM ADDR.
78A4          00850        END    SPLIT          ;AUTOSTARTS AT SPLIT
00000 TOTAL ERRORS
BSAVE   7D33 00820
CHANGE  7904 00560  00440 00530
IKBD    78E5 00420  00330
KBD     4016 00130  00310 00340 00430
MEM16K  78A4 00100  00140
MEM32K  B8A4 00110
MEM48K  F8A4 00120
MKEND   7FFF 00840
SKBD    78F1 00460  00320 00420
SPLIT   78A4 00150  00850
SPSAVE  7FFE 00830  00220 00640 00650
SWAP    7929 00740  00590 00620 00790
VSAVE   7933 00810  00160 00570
```



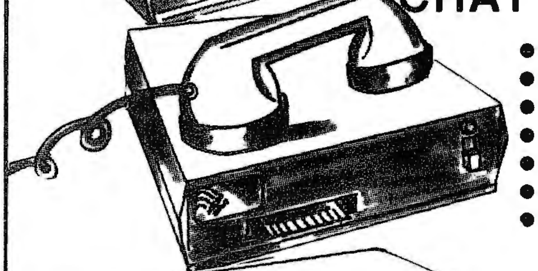
TRS-80* interfacing alternatives...

Featuring the Quality Engineering and Manufacturing of THE MICRO MINT

COMM-80

- FULL 8-BIT PARALLEL PORT
- RS-232-C SERIAL PORT (up to 19,200 baud)
- UP TO 16 UNITS CAN BE CHAINED TOGETHER
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- INCLUDES TERMINAL SOFTWARE

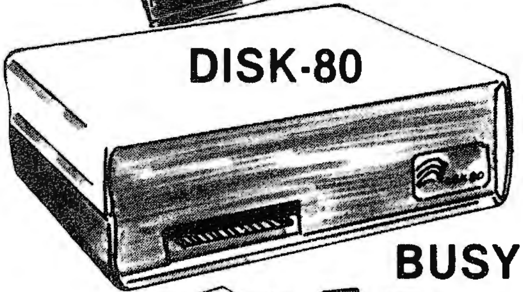
ONLY $179.95



CHATTER BOX

- FULL 8-BIT PARALLEL PORT
- RS-232-C SERIAL PORT (up to 19,200 baud)
- ACOUSTIC MODEM — 300 BAUD
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION
- CONNECTS TO KEYBOARD OR E.I.
- CAN BE CHAINED TOGETHER WITH COMM-80
- INCLUDES TERMINAL SOFTWARE

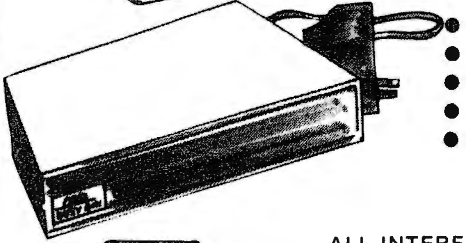
ONLY $259.95



DISK-80

- DISK CONTROLLER (up to 4 drives)
- DATA SEPARATOR INCLUDED
- INCLUDES 16K OF RAM
- PROVISION FOR AN ADDITIONAL 16K RAM
- TRS-BUS CONNECTOR FOR FUTURE EXPANSION

ONLY $299.95



BUSY BOX

- ALLOWS TRS-80 TO CONTROL AC APPLIANCES
- CABLE AND CONNECTOR FOR TRS-80 KEYBOARD OR E.I.
- WORKS WITH THE POPULAR BSR X-10 SYSTEM
- INCLUDES BASIC LISTING FOR CONTROL ROUTINE
- LEVEL II MINIMUM SYSTEM

ONLY $109.95

ALL INTERFACES ARE RADIO SHACK HARDWARE AND SOFTWARE COMPATIBLE.

HARDSIDE 6 SOUTH ST., MILFORD, NH 03055

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

VISA

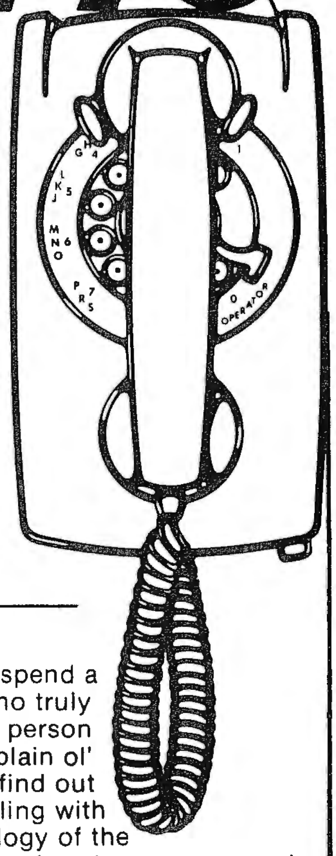
The Software Exchange & HardSide (Div. of Robitaille & Sons Enterprises, Inc.) SoftSide Publications

*TRS-80 is a trademark of Tandy Corp.

Where can you dial one TOLL FREE number and be able to select the finest software from the world's foremost suppliers?

1-800-258-1790

The Software Exchange

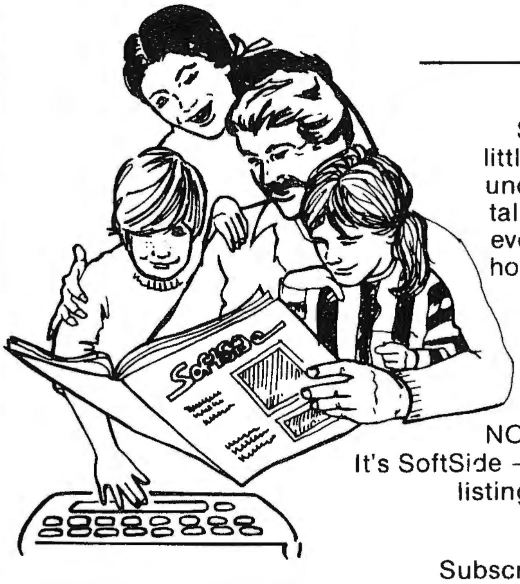


Our stockrooms are overflowing with the finest software available in today's marketplace. We carry software from all these companies:

- ●Racet
- ●Adventure International
- ●Personal Software
- ●Web Associates
- ●Small Systems Software
- ●Sensational Software
- ●Synergistic Software
- ●Strategic Simulations
- ●Lance Micklus
- ●Softape
- ●Quality Software
- ●Image Products
- ●Acorn Software
- ●Hayden
- ●Muse
- ●Microsoft
- ●Apparat

We could not possibly list all of the software on this page, however, if you send us $1.00, you'll receive our catalog plus a $2.00 credit toward your next purchase!

TSE



Suppose you and your family could spend a little time each month with someone who truly understands microcomputers. And this person talked to you about your computer in plain ol' everyday English. Suppose you could find out how these people are successfully dealing with the ever-increasing technology of the microcomputer. Suppose too, that each article was geared to the person with a beginning or intermediate interest in personal computing.

NOW you can get all this, and more, at a price you can afford. It's SoftSide — the magazine that provides its readers with tried and true listings, along with structured "walk-thrus" of some of the most stimulating software pieces, month after enjoyable month.

Subscribe today! Until November 1, 1980, the price is only $15.00. After November 1, the price will be $18.00, so why not save $3.00 and start your family on the road to microcomputing?

SS A SoftSide Publication

The Software Exchange 6 SOUTH ST., MILFORD, NH 03055

VISA

To order: Call Toll-Free 1-800-258-1790 (in NH call (603) 673-5144)

master charge

The Software Exchange & HardSide (Div. of Robitaille & Sons, Enterprises, Inc.) SoftSide Publications

JKL for Level II

Brad Hoza, Puyallup, WA

If you have been around a disk system using NEWDOS, you have probably become familiar with it's "JKL" feature, which copies whatever is on the screen to the line printer when the J, K and L keys are pressed at the same time. This is a handy feature of the system, and comes into use often. But what about the people who are still using Level II?

This article will discuss implementing the "JKL" feature in Level II. In addition, there is a second routine, activated by pressing "234", which will make a hardcopy of a graphics screen.

Because most printers do not support graphics, the JKL routine changes a graphics character to a period. If your printer does support graphics, this means you do not need the 234 routine and you can simply omit lines 1210-1230, 2600-2800 and lines 4200-11700.

The program is 239 bytes long, so change the number after the ORG and END statements to correctly place the routine at the end of your RAM (32767 less 239, or 32528 for a 16K machine). Don't forget to answer MEMORY SIZE with this number less one.

If you own one of Radio Shack's printers, all you have to do is type in the listing using the Editor/Assembler and assemble it and make a tape of the object code. Then, whenever you power up, load the routine in and execute it. A "READY" will immediately be displayed; however, the first command you enter may sometimes return an "OM ERROR". This does not affect anything else, and you can proceed as normal.

If you own a serial printer and/or have to load a printer routine into high memory before your printer will work, there are some modifications you must make.

The easiest of these is to locate the program in memory so that it will not overlap your printer routine. To do this, simply take the address you reserve memory size with, subtract 239, and change lines 100 and 1370 with this number. Now, when you power up, reserve memory with this number less one. Then change the value of LPRINT (line 1360) with the location of your printer routine (one more than you reserved memory size with).

Type in the listing using the Editor/Assembler. First make a temporary source code file of the program using the Write command. Then assemble the listing, making the object code file. Now power up, reserve memory with the new number, load your printer routine, and finally the JKL program. Press the J, K and L keys all at once. If the printer starts printing whatever happens to be on the screen, you are all set. If the printer locks up, or prints a bunch of garbage, it means that the printer routine looks at either the B or C registers for the ASCII coded characters, not the A register as Radio Shack's printers do.

First, you will need to figure out which register it is. By assembling and executing the program in firure 1 (with your printer routine loaded) the printer will print either a "B" or a "C", depending on which register the printer looks at. We will call this register "r".

Now load the source code file of the program back into the Editor/Assembler. Then all you have to do is insert the following two statements before *every* line that reads "CALL LPRINT"

```
PUSH    BC
  LD     r,A
```

and put the statement "POP BC" directly after the "CALL LPRINT" statement.

Now assemble and make the object code file and proceed as before.

```
0   ;no need to worry about the orgin!
10 ORG    4000H
20 LD     B,42H
30 LD     C,43H
40 CALL   NNNN   ;put the address of your printer
                 ;routine here.
50 END 4000H
```

Figure 1

```
BF1Ø         ØØ1ØØ          ORG    ØBF1ØH         ;OR 7F1ØH OR ØFF1ØH
BF1Ø 2A1E4Ø  ØØ11Ø START    LD     HL,(4Ø1EH)
BF13 2236BF  ØØ12Ø          LD     (DISPLY+1),HL
BF16 211FBF  ØØ13Ø          LD     HL,CHECK
BF19 221E4Ø  ØØ14Ø          LD     (4Ø1EH),HL
BF1C C3CCØ6  ØØ15Ø          JP     Ø6CCH
BF1F F5      ØØ16Ø CHECK    PUSH   AF
BF2Ø E5      ØØ17Ø          PUSH   HL
BF21 D5      ØØ18Ø          PUSH   DE
BF22 C5      ØØ19Ø          PUSH   BC
BF23 3AØ238  ØØ2ØØ          LD     A,(38Ø2H)
BF26 FE1C    ØØ21Ø          CP     28
BF28 28ØE    ØØ22Ø          JR     Z,JKL
BF2A 3A1Ø38  ØØ23Ø          LD     A,(381ØH)
BF2D FE1C    ØØ24Ø          CP     28
BF2F 2832    ØØ25Ø          JR     Z,TWO34
BF31 C1      ØØ26Ø OUT      POP    BC
BF32 D1      ØØ27Ø          POP    DE
BF33 E1      ØØ28Ø          POP    HL
BF34 F1      ØØ29Ø          POP    AF
BF35 C3ØØØØ  ØØ3ØØ DISPLY   JP     $-$
BF38 21ØØ3C  ØØ31Ø JKL      LD     HL,3CØØH
BF3B Ø64Ø    ØØ32Ø          LD     B,64
BF3D ØE1Ø    ØØ33Ø          LD     C,16
BF3F 3A4Ø38  ØØ34Ø LOOP1    LD     A,(384ØH)
BF42 FEØ4    ØØ35Ø          CP     4
BF44 28EB    ØØ36Ø          JR     Z,OUT
BF46 7E      ØØ37Ø          LD     A,(HL)
BF47 FE8Ø    ØØ38Ø          CP     128
BF49 FA4EBF  ØØ39Ø          JP     M,CONT1
BF4C 3E2E    ØØ4ØØ          LD     A,46
BF4E CD3BØØ  ØØ41Ø CONT1    CALL   LPRINT
BF51 23      ØØ42Ø          INC    HL
BF52 1ØEB    ØØ43Ø          DJNZ   LOOP1
BF54 3EØD    ØØ44Ø          LD     A,13
BF56 CD3BØØ  ØØ45Ø          CALL   LPRINT
BF59 Ø64Ø    ØØ46Ø          LD     B,64
BF5B ØD      ØØ47Ø          DEC    C
BF5C 79      ØØ48Ø          LD     A,C
BF5D FEØØ    ØØ49Ø          CP     Ø
BF5F 2ØDE    ØØ5ØØ          JR     NZ,LOOP1
BF61 18CE    ØØ51Ø          JR     OUT
```

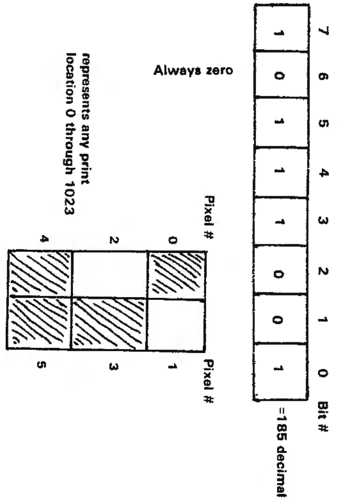


Figure 2

```
BF63 21ØØ3C  ØØ52Ø TWO34    LD     HL,3CØØH
BF66 161Ø    ØØ53Ø          LD     D,16
BF68 1EØØ    ØØ54Ø LOOP6    LD     E,Ø
BF6A Ø64Ø    ØØ55Ø LOOP2    LD     B,64
BF6C E5      ØØ56Ø          PUSH   HL
BF6D 3AØØØF  ØØ57Ø LOOP5    LD     A,(384Ø)
BF7Ø FEØ4    ØØ58Ø          CP     4
BF72 2ØØ3    ØØ59Ø          JR     NZ,CONTA
BF74 E1      ØØ6ØØ          POP    HL
BF75 18BA    ØØ61Ø          JR     OUT
BF77 3E7F    ØØ62Ø CONTA    LD     A,127
BF79 4E      ØØ63Ø          LD     C,(HL)
BF7A B9      ØØ64Ø          CP     C
BF7B FA82BF  ØØ65Ø          JP     M,LOOP3
BF7E ØE8Ø    ØØ66Ø          LD     C,128
BF8Ø CBF1    ØØ67Ø          SET    6,C
BF82 7B      ØØ68Ø LOOP3    LD     A,E
BF83 FEØ2    ØØ69Ø LOOP4    CP     2
BF85 2ØØ8    ØØ7ØØ          JR     NZ,LOOP4A
BF87 CB71    ØØ71Ø          BIT    6,C
BF89 28Ø4    ØØ72Ø          JR     Z,LOOP4A
BF8B 7E      ØØ73Ø          LD     A,(HL)
BF8C F5      ØØ74Ø          PUSH   AF
BF8D 182E    ØØ75Ø          JR     CONT4
BF8F B7      ØØ76Ø LOOP4A   OR     A
BF9Ø 17      ØØ77Ø          RLA
BF91 17      ØØ78Ø          RLA
BF92 E5      ØØ79Ø          PUSH   HL
BF93 D5      ØØ8ØØ          PUSH   DE
BF94 219CBF  ØØ81Ø          LD     HL,INDEX
BF97 16ØØ    ØØ82Ø          LD     D,Ø
BF99 5F      ØØ83Ø          LD     E,A
BF9A 19      ØØ84Ø          ADD    HL,DE
BF9B E9      ØØ85Ø          JP     (HL)
BF9C CB41    ØØ86Ø INDEX    BIT    Ø,C
BF9E 1812    ØØ87Ø          JR     CONTB
BFAØ CB49    ØØ88Ø          BIT    1,C
BFA2 18ØE    ØØ89Ø          JR     CONTB
BFA4 CB51    ØØ9ØØ          BIT    2,C
BFA6 18ØA    ØØ91Ø          JR     CONTB
BFA8 CB59    ØØ92Ø          BIT    3,C
BFAA 18Ø6    ØØ93Ø          JR     CONTB
BFAC CB61    ØØ94Ø          BIT    4,C
BFAE 18Ø2    ØØ95Ø          JR     CONTB
BFBØ CB69    ØØ96Ø          BIT    5,C
BFB2 D1      ØØ97Ø CONTB    POP    DE
BFB3 E1      ØØ98Ø          POP    HL
BFB4 F5      ØØ99Ø          PUSH   AF
BFB5 28Ø4    Ø1ØØØ          JR     Z,CONT3
BFB7 3E23    Ø1Ø1Ø          LD     A,35
BFB9 18Ø2    Ø1Ø2Ø          JR     CONT4
BFBB 3E2Ø    Ø1Ø3Ø CONT3    LD     A,32
BFBD D5      Ø1Ø4Ø CONT4    PUSH   DE
BFBE CD3BØØ  Ø1Ø5Ø          CALL   LPRINT
```

```
BFC1 D1        01060           POP    DE
BFC2 F1        01070           POP    AF
BFC3 FE04      01080           CP     4
BFC5 280C      01090           JR     Z,CONT5
BFC7 FE0C      01100           CP     12
BFC9 2808      01110           JR     Z,CONT5
BFCB FE14      01120           CP     20
BFCD 2804      01130           JR     Z,CONT5
BFCF 7B        01140           LD     A,E
BFD0 3C        01150           INC    A
BFD1 18B0      01160           JR     LOOP4
BFD3 23        01170 CONT5     INC    HL
BFD4 1097      01180           DJNZ   LOOP5
BFD6 3E0D      01190           LD     A,13
BFD8 D5        01200           PUSH   DE
BFD9 CD3B00    01210           CALL   LPRINT
BFDC D1        01220           POP    DE
BFDD E1        01230           POP    HL
BFDE 1C        01240           INC    E
BFDF 1C        01250           INC    E
BFE0 3E06      01260           LD     A,6
BFE2 BB        01270           CP     E
BFE3 2085      01280           JR     NZ,LOOP2
BFE5 014000    01290           LD     BC,64
BFE8 09        01300           ADD    HL,BC
BFE9 15        01310           DEC    D
BFEA 7A        01320           LD     A,D
BFEB FE00      01330           CP     0
BFED C268BF    01340           JP     NZ,LOOP6
BFF0 C331BF    01350           JP     OUT
003B           01360 LPRINT    EQU    3BH
BF10           01370           END    0BF10H
00000 TOTAL ERRORS
```

PROGRAM STORE (202) 337-4691

4200 Wisconsin Ave N W PO Box 9609 Washington D C 20016

All programs for TRS-80
16k, Level II computer.

SYSTEM SAVERS

by Tom Stibolt from Acorn

If you ever use the SYSTEM command, you should buy this two program package. These programs allow you to save any system format tape on tape or disk, plus offer several features for machine language programmers. Many two part, protected system tapes like Sargon II are not system format.

With FLEXL, which is one of the two programs, you can make back-up copies of any system format tape. Most often a cassette that you make will load easier than an original. Plus you can find the filename on any system tape because it is displayed on the screen. And at any time you can stop the reading of the tape by pressing <BREAK>.

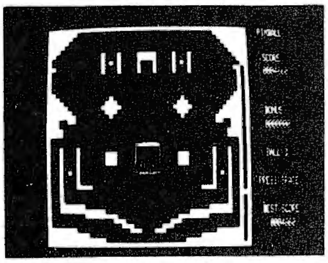
For any machine language programmer, FLEXL offers the advantage of producing more efficient tapes than the assembler. Also, it is written to interface directly with monitor programs. And you can merge machine language tapes into one file.

Disk drive owners can use TDISK to save any system format tape onto disk. Adventure, Airaid, Ting-Tong, Editor Assembler and other programs cannot normally be loaded to disk using TRSDOS. Now, TDISK allows you to save these programs onto disk. After DOS READY you will be able to simply type the filename and be up and running. It even loads non-contiguous tapes. TDISK will greatly increase the benefit of owning a disk drive.

And as a FREE BONUS, Acorn provides instructions on how to load MicroChess 1.5 onto disk.

Complete your system with the routines not found in either Level II or DOS for only $14.95. Order your System Savers, today!

*TRS-80 is a trademark of Tandy Corp



PINBALL

by John Allen from Acorn

Get your flipper fingers ready for action in this real-time, machine language game. Lots of sound and flashing graphics. There are five speeds so anyone can play. This version features the dreaded "Bermuda Square"!

Protected tape $14.95
Protected disk $20.95

FLIGHT SIMULATION

from Sub-Logic

The wait is over. 3D graphics would seem impossible given the resolution of the TRS-80. But wait till you see this package. Even if you don't order it, try to see this one.

First you must learn to fly using instruments, radar and the cockpit window view. While it is easy to take off, it is even easier to crash.

Once you have some air time, bomb the enemy fuel depot. They'll send five war planes after you. Each one maneuvers differently in dogfights. Good luck.

Level I and II $25.00

DEATH DREADNAUGT

from Programmer's Guild

This machine language adventure is twisted! Around every turn death awaits. Be careful of what you touch. Disgusting room descriptions. For those who would like a ghoulish adventure. $14.95.

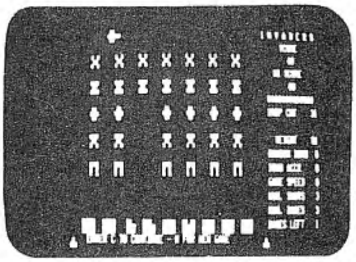
Also available from the Programmer's Guild are two adventures written in BASIC: Spider Mountain and Lost Dutchman's Gold. $14.95 each.
All three on disk for $35.00

RESCUE at RIGEL

from Automated Simulations

In this first Starquest you must save the prisoners of the evil High Tollah. There are both a different style of play and features than Dunjonquest, but similar floor graphics.

	Tape	Disk
Rescue at Rigel	$14.95	$19.95
Temple of Apshai	24.95	29.95
Datestones of Ryn	14.95	19.95
Morloc's Tower	14.95	19.95



INVADERS FROM SPACE

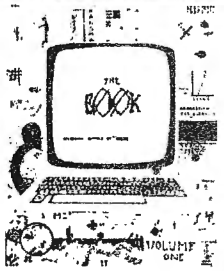
by Carl Miller from Acorn

"Maybe it's too fast!" Perferred by all the local arcade addicts, this machine language game has great sound. Alien Invasion, Invaders, and Invaders Plus just can't compete. You can adjust six parameters including the speed, if you think it's too fast. Only a few heros will be able to save earth.

Protected tape $14.95
Protected disk $[illegible]

THE BØØK

ACCESSING THE TRS-80* ROM



from Insiders Software Consultants

For the learner or expert! This first of three volumes gives you access to over firy arithmetic, mathematical, and data conversion routines. Includes from Insiders Software Consultants

For the beginner or expert! This first of three volumes gives you access to over fifty arithmetic, mathematical, and data conversion routines. Includes sample programs, a disassembler program, and a commented listing of these ROM routines.

A memory map noting over 500 addresses is included. 136 pages. $14.95

INSIDE LEVEL II

by John Blattner, PhD from Mumford

Much to our surprise, this book and THE BØØK are complimentary. The style is easy but packed with information, though there is no commented listing of the ROM. For both Disk and Level II, complete entry and exit information is included for conversion, arithmetic, mathematical, keyboard, cassette and video routines. Part two describes how to link assembly language with BASIC. 60 pages. $15.95

DISASSEMBLER

by Roy Soltoff from Misosys & Acorn

This two pass, machine language, Z-80 disassembler produces symbolic labels with output to either the video monitor, printer or tape. Radio Shack's Editor Assembler can load the tapes. If you own the Editor/Assembler, complete the package with this program. Program on tape for two different memory locations. Cassette version NOW only $14.95

THE PROGRAM STORE
4200 Wisconsin Ave NW
P.O. Box 9609
Washington DC 20007

VISA AND MASTERCARD

TO ORDER TOLL FREE 800-424-2738

for program information call
(202) 337-4691

Mail orders. Send check or money order for total purchase price, plus $1.00 for shipping & handling. Charge card customers, please include card number, expiration date, and your name as it appears on the card.

48K with Disks - For Free!

Cameron C Brown

Bellarmine Prep is a 900 student private school located in Tacoma, WA. As with all schools, sources of funds are scarce. Our Math Department operates on a very limited budget, but over the past two years we have been able to obtain four TRS-80 Level II computers; three of which are 4K and one is a 48K system, complete with two disk drives. Each computer is equiped with a printer and a serial interface

Our key to success has been to have the computers pay for themselves. The computers are used in six different fund raising activities. Some of the events are easy to run and others are major projects. The object of this report is to help other institutions obtain computer hardware for a minimal outlay of funds.

MINOR FUND RAISING

These projects require little or no work on the instructor's part, and can be run by students.

Each fall the school sponsors a carnival and fund raiser called an "Oktoberfest". During Oktoberfest weekend, we open up the computer room and sell game-time and specialized printouts for 50¢ to $2.00 each. Students have written programs that will print out Biorhythms, posters, calendars, etc. A customer can play anything from Tic-Tac-Toe to Chess for 25¢ to $1.00 depending on the length of time to play. During just one weekend we were able to raise about $200.00. There is almost no overhead and students are more than willing to man the "booth". In fact, for many students in programming there is nothing more appealing than to have access to the computers all day and night.

During the regular school day, the computers are used for instructional purposes. But we have found that after school many students want to play games on the machines. For 25¢, any student may play a game if the computer is free of programming students. During the course of one school year, we have been able to raise enough to pay for subscriptions to three computer magazines. The money also goes to buy other, more expensive games such as Sargon. Twenty-five cents is not much, but $2.50 per week does add up.

After discussing possible computer uses with other faculty, we found out that the school newspaper needed mailing labels printed for their subscribers. We have set up a subscriber file and now run off mailing labels for 3¢ each. This nets the department about $5.00 every two weeks. All data entry and processing is done by the students, it only took one hour to train students to run the program. We are now investigating setting up data files for other in-school needs. Even though the administration has access to outside computer services, smaller organizations within the school do not. Some potential areas of service are mailings for the Senior class, accounts receivable for the annual, budget management for the student body, etc.

MAJOR FUND RAISING

These projects involve much more time and careful planning. Much more can be raised if you involve the whole school in a project.

Last year, our school held a "Jogathon" in which the students gained pledges for donations depending upon the number of laps they would run in an hour. All of the billing, reporting and promotional work was supplied by a professional company. The event was a large success and raised over $38,000. gross. We investigated the data processing involved and decided that it was feasible for us to take on this project. In return for a complete 48K system with two drives, and printers for all the computers and necessary supplies, the school's development office agreed to have us run the "Jogathon" this year.

Working with an exceptional programming student for about three weeks over the summer break, we were able to develop the programs necessary. We expect to be able to keep track of 900 runners, their number of laps, who sponsored them, what was the pledge amount (total or per lap), and which homeroom is to be credited with the income. At the same time, we are keeping track of around 5000 sponsors, their names, addresses, who they sponsored, amount due and amount received. We have written a billing routine using specialized mailers that will notify each sponsor of his amount due for all runners he pledged. The development office will be given reports listing all sponsors, joggers, amount raised per homeroom (and by whom), totals received and amount outstanding.

The cost to the school is expected to be about one-half that of last year. For each succeeding year, we are charging the school $2000 plus 10% of the gross over $30,000.

We are concerned about the amount of keypunching needed and are spending time now training reliable students on how to operate all programs. The reliability of data retrieval and processing time is also a concern, but we won't know for sure until the Jogathon has been run. If worse comes to worse, we plan on having all freshman math students address 20 mailers each before they pass Algebra!

Most students love to go to dances, and what would be better than having five dates at once? Students fill out a questionaire and are then paired with the five most compatible matches. Each participant gets a number and a list of five other numbers. At the dance, each person wears his or her number for identification. The last time we had this event, the response from the students was excellent. A few people objected to being paired up with their brother or sister, and we have yet to figure out that situation. For many students it is a dance with a real "ice-breaker". The questionaire includes many areas of interest and students spend as much time finding out what they have in common as they do dancing. Also, it isn't every day that a freshman boy gets to dance with a senior girl!

Plan on about one week for promotion and data entry. Income varies greatly depending on the time of the year, what band is hired, location, etc., but you should net a minimum of $2.00 per participant.

We are not unique in our fund raising activities. The computer dance and Jogathon are also being used at Sehome High School in Bellingham, WA. Each year, schools are sent proposals from professional promotional companies. With a little work and imagination most activities can be done with in-house equipment.

Besides the obvious benefits in being able to obtain hardware, there are many other intangible gains. We find that the administration is more than willing to support activities which result in direct income without having to pay a percentage. At the same time, many more parents and students are made aware of our needs and facilities. In fact, last year an alumni donated a 4K Level II computer to our department. He chose to remain anonymous, a smart move, else we would have been back for more. We have also found that our programming students gain from our efforts. Many more students understand what is really involved in data processing and what is needed to do a professional, real-life job. Students considering a career with computers are able to get a better feel for the profession while still in high school, and those involved with the projects gain real pride in their accomplishments.

There are many other sources for funds. Federal monies are available under Title IV funding and your local Radio Shack dealer should have information available on it. Plan on finding a good grant writer to help you. We have noted that many companies donate appliances to home economics departments at or below cost, since the students trained on their equipment tend to become future customers. We have been trying to convince Radio Shack of the wisdom of such a policy. At the same time we have offered to become a demonstration school.

Many private foundations are willing to make one-time contributions for hardware. Check your local library for information on foundations and those which are applicable to your situation. Again, plan on writing many letters, rewriting proposals to fit unique requirements, and waiting for someone to nibble (or is it byte?).

In keeping with the spirit of this report, we are more than willing to discuss our methods with anyone interested. Contact Cameron Brown, Math Dept Chairman, Bellarmine Prep, 2300 South Washington, Tacoma WA 98405 ●

PROGRAM STORE (202) 337-4691

4200 Wisconsin Ave.NW P.O.Box 9609 Washington D.C. 20016

Save time & space!

PACKER

by Cottage Software

This is the ultimate editing tool for BASIC program lines. There are five commands which allow easier reading of BASIC programs and more efficient execution by the computer.

The 'unpack' command breaks multiple statement lines into single statement lines with extra spaces for easier reading and editing. The 'short' command deletes any unnecessary words like LET and all REMarks.

The 'pack' commmand compresses lines into multiple statement lines up to the maximum length you specify while maintaining complete program logic. This can easily reduce the memory requirement by more than 33%! As you can imagine this also speeds up execution of a program, saves time in loading a program from either tape or disk and saves disk space.

And the 'move' command allows you to move any section of your program to a new location. With the 'renumb' command you can renumber your BASIC lines.

So if your programs need more memory, or you need more time, order your 'packer'! 16k, 32k and 48k versions supplied on two cassettes for $29.95.

Morloc's Tower

from Automated Simulations

As Brian Hammerhand, you race to save the comely wench Imelda from the fireballs of Morloc the Mad. An exciting fantansy role playing game in the Dunjonquest tradition that started with "The Temple of Apshai" and continued with "Datestones of Ryn." This is the deadliest Dunjonquest yet! $14.95

"Datestones of Ryn" $14.95

"Temple of Apshai" $24.95

Supplied on disk for an additional $5 per program.

Dungeon

from Chameleon

A full fantasy & adventure where you create your character, choose from among 26 types of armor and 80 weapons, and search for fame and fortune. In each dungeon there are random events which occur, but skill most often determines the outcome. The program has many prompts and the twenty page, illustrated manual explains the game. Requires 32k and two disk drives. $34.95

Galactic Trader

by Douglas Carlston

Trade your way to a fortune in a three dimensional universe. You must barter, deal with the energy cartel and watch for spies. This is truly an outstanding computer game by the author of "Galactic Empire." $14.95

Also available "Galactic Empire" for $14.95

STRUCTURED BASIC TRANSLATOR

by Gene Bellinger

Try structured programming. You can write programs using PROCEDURES, CALLS, CASE-CALLS, IF-THEN-ELSE, WHILE and UNTIL. Once written, SBT will quickly translate the structured code into an effecient BASIC program. Speeds up program development and documentation. The program is both fast (a 20k BASIC program in less than 4 minutes) and compact. Requires 32k and one disk drive. Supplied on disk for $29.95.

Air Flight

from Instant Software

Transform your TRS-80 into a cockpit full of elaborate controls. Take off, land and do acrobatics once you learn how to fly. $7.95

Air Raid

from Small Systems

Real-time, "shooting gallery" game which is better than the arcades. In machine language for excellent graphics and endless hours of fun. ~~$14.95~~ NOW only $9.95

Space War

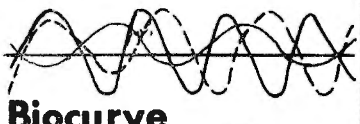
by Device Oriented Games from Acorn

Two-player, real-time space battle lets each player control a spaceship with rotate, thrust, fire and hyperspace. Five game options (including gravity) and three speeds included. In machine language. $9.95

Calculator Plus

from Manhattan Software, Inc.

Turn your TRS-80 into a calculator. This program has optional printer features, performs all calculations in double precision and will chain and mix calculations. Optional dollar format. $9.95.



Biocurve

by Microflair from Hayden

Graph your biorhythms and calculate your critical days. Compare yours with a friend. $9.95

KEYEDIT

by Phil Pilgrim from Discovery Bay

This machine language utility is for all users of either Level II or Disk BASIC. The functions give you auto-repeat on every key, screen-oriented editing, and debounce. In the editing mode, you just move the cursor to any position on the screen to instantly insert or delete. Plus, whole BASIC statements can be copied to other parts of the program.

You also get single keystroke entry. For example, just press <SHIFT> R to enter the word READ. And you can redefine the keys to any BASIC keyword.

In addition, the keyboard macro facility allows you to define any frequently typed pattern and then execute it with a single keystroke. This is especially great for DATA entry and other repetitive tasks.

Order the greatest keyboard utility yet! $18.95

DEBUG

by Bob Pierce from Quality

Single step your programs through RAM and ROM, with this relocatable machine language utility. Includes multiple breakpoints, option to execute CALLs in a single step and several display options. $14.95

DISASSEMBLER 1.2

by Roy Soltoff from Misosys & Acorn

This two pass Z-80 disassembler produces symbolic labels with output to either the video monitor, printer or tape. Radio Shack's Editor Assembler can load the tapes. If you own the Editor Assembler, complete the package with this program. Program on tape for two different memory locations. TRS-80 Level II ~~$19.95~~ NOW only $14.95

DDT

Disk Drive Timer

by Disco-Tech

Analyze and adjust your disc drive motor speed with real-time graphic display. Manual details use for Radio Shack, Shugart, MPI, Pertec, and Vista drives, but will work with any drive. All you need is DDT, two screwdrivers and five minutes. On disk for $19.95

TO ORDER TOLL FREE

800-424-2738

CREDIT CARD CALLERS MAY PHONE US 24-HOURS A DAY. OR CLIP THIS COUPON AND MAIL YOUR ORDER TODAY.

THE PROGRAM STORE™
4200 Wisconsin Ave NW
PO Box 9609 Dept KR8
Washington, D.C. 20016

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:

BUSINESS COMPUTING

Check Write

A personal finance system

R A Shmina
Detroit, MI

Part 1 of a 2 Part Series

Having begun, like so many others, my computer interest with a TRS-80 Level I 4K System, I quickly advanced to Level II 16K with Expansion Interface with two cassette recorders and the tractor feed line-printer. The printer quickly proved itself invaluable, because an actual printout served far better than a video display for analyzing and understanding programs. From my own experience, it was the use of the printer that provided the best learning tool and the most enjoyment and satisfaction that I derived from my TRS-80. It was only exceeded when I added two disk units to my 48K system. However, the true appreciation of the disk units can only come after using the tape cassette system.

The Level I User's Manual provided me with an introduction to personal computing and a solid basis for understanding and using the TRS-80, and it was the example regarding temperature and humidity in the Appendix which really demonstrated the great potential that a computer possesses. Every new TRS-80 owner should study this manual. The jump to Level II was quite a shock because the Level II Manual only describes the Level II Statements and Commands and doesn't give the instruction and illustrative problems to the extent that the Level I Manual did.

The ability of Level II in handling two dimensional arrays and its greater string handling capability than Level I was an area of programming that I wanted to discover and use. As well as a learning experience, I desired the tangible results of my TRS-80 and my efforts, so a plan of action to develop a program to handle all checkbook chores, using a two dimensional string array, was charted. The programs that resulted from this effort will hopefully transfer to the reader a better understanding of your TRS-80 and its operation in a format with which we are all familiar -- your own checking account.

The plan of action visualized a person sitting at his TRS-80, opening a bill, typing in data, and letting the computer write the check and do the rest of the bookkeeping. the plan also visualized the creation of a data base that could ultimately be used for the necessary end-of-year summarization at tax time. The action plan was quickly summarized as follows:

1. Fill a two dimensional string array with the data for a check (date, check number, amount, payee, budget code).
2. Use this data to print out, on preprinted check forms, the actual check.
3. Use this data to create a summary of outstanding checks and a constantly growing summary of cancelled checks, as well as keeping track of all deposits, bank charges, and bank balance.
4. Use the cancelled check summary to be able to obtain a summary of any budget code for tax purposes, or any purpose you desire.

Let us now get a brief overview of the entire system.

CKWRTDAT/ONE accepts a line of input consisting of data, check number, amount, payee, and assigned budget code. After accepting this input, the program saves the data on disk or tape (in a file named NEWCHKS), and it is used immediately to write the checks. This same data file is later used in CKWRTDAT/TWO for balancing the books.

One of the features of this program is that the assigned budget code can be any alphanumeric mnemonic you desire to create. You are not limited to a numerical code. This is because of the manner in which all data is stored, that being in a two dimensional string array. The fields are not limited to any predetermined length, but, for screen display only, they are limited to a very practical size. The programs can be used with disk or tape and require at least a 16K Level II System with a printer. The programs can be "edited" to provide video display only, but a printer is necessary to really appreciate these programs.

Another feature of the program is the ease of correcting typing or entry errors. We will demonstrate how to stop a program, reassign values, and then continue the program to completion. The method used is not designed to be "idiot proof" but rather to hopefully give a better understanding of exactly what the Basic program is doing. Realizing that not all bills are paid by the computer process, CKWRTDAT allows for the manually written checks. These are the times the wife goes to the store and they won't wait until your computer run for their check. We will demonstrate many techniques that are easier to understand by an actual program than by long-winded explanations.

CKWRTDAT/TWO does the actual bookkeeping. It works from a file of outstanding checks, "OUTSTNDG/CXX" and a file of new checks, "NEWCKS/XXX". Then you can cancel the same checks the Bank cancels and enter any deposits. The new balance is calculated and an updated file of "OUTSTND/CKS" is saved as well as a file of "CANCECKS/XXX". The balances are then calculated. The last item "Correction to Program Balance" is the invisible "cushion" I always keep in my checking account. If you do not use the "Invisible Cushion Principle", this item will always be zero, unless you have made an entry error. This is complete "hardcopy output" from every session to provide an "audit trail" through the entire process. CKWRTDAT/TWO works with three files: "NEWCHKS/XXX", "OUTSTNDG/XXX", and "CANCECKS/XXX". With "NEWCHKS" and a lot of work, you can reconstruct any session, so take especially good care of it. Even though the program creates a new "CANCECKS/XXX" every session, these files are "appended" to form the master file of "DATAFILE" for the year. For those of you on a tape I/O system, we will explain how to perform this same function.

Once we have a master file of "DATAFILE" the fun really starts. Using the program "SUMMARY/ONE" we can obtain a quick summary (totals only) of all our assigned budget codes, while using "SUMMARY/TWO" enables us to obtain a complete listing of all checks in any budget code. An option exists in SUMMARY/TWO to obtain a listing by payee, if desired.

The following is a line by line explanation of the program "CKWRTDAT/ONE". If you faithfully follow the procedures, you will discover much of the operation of your TRS-80 and have an understandable and extremely useful package which you can alter to your own needs. You will also see that CKWRTDAT/ONE is the basis for a Data Base Generator system which can be extremely efficient. By using the technique CKWRTDAT/ONE uses, it is possible to store all your data in small, concise, numerically coded packages and convert the codes to their predetermined meaning by the process demonstrated here.

Line 100-210 **Are readily self-explanatory.**

Line 210 **Clears string space and dimensions variables and arrays. "Clear" must precede other dimension statements as it wipes the "clear" also. From this point on, variables A, C, D, and P are strings. Also A and P are two dimensional arrays. Array P will be explained later.**

Line 220-230 Enter the date in the form of MMDDYY. Variable D represents date input as six digits. D is a string. Refer to pages 5/1 and 5/2 of L II Manual for background for use of quotes in certain instances. I is initialized and is used as a counter of the rows of (input items) array A, which is the input data array Next, enter the complete Filespec. I use "NEWCHKS/XXX", starting with A and incrementing each session.

Line 240-270 Prompting message and accepts a type line of input. *Each item must be separated by commas.* DO NOT USE ANY COMMAS OR SEMICOLONS OTHER THAN TO SEPARATE DATA ITEMS!

Line 280 Array column element five is used as a "cancel" flag later in the program. Now it is set to zero. It's just a readily visible indicator of the current status of the check.

Line 290 Test to see if done entering checks.

Line 300 Increment counter (array row number) - go back for more.

Line 310 Self-explanatory with remark. Even though this array element is filled with zeros, we will only use the meaningful elements below it.

Line 320 Initializes SUM = O. If we do not do this, SUM will be double if we make another list later in the program. Lprints heading and calls the subroutine for printing the list.

Line 330-410 **Asks if corrections are needed. This is a unique part of the program, so be alert. If you have detected a spelling error, etc., respond "YES" to "Any Corrections necessary". The program will ask you to "identify" the row and column of the element to be corrected. Enter them and also the correct entry for that element. Enter 0, 0, 0, to exit this mode. If you forgot what row number (I), don't panic; press "Break". Everything is in memory and will not be lost, unless you panic.** You are now in Command Mode. DO NOT ALTER THE PROGRAM. You can now type *(use no line number)* FOR X = 1 to I: PRINT A(X,0); A(X,1); A(X,2); A(X,3); A(X,4); A(X,5); " ";X: NEXT X and the entire list will appear on the screen (use SHIFT @ to stop scrolling): *Remember number* (I) of entry with the error and now type "CONT" and the program resumes. Now make the necessary reassignment of values by entering I, J, and T$ as directed by the program. Also, you can press "BREAK" anywhere in the program and reassign values like A(N,3) = "(NEW)": CONT and then press "ENTER". You have merely stopped the program run and reassigned the values to variables and then continued the program run. You should do this once just to get the feel of things. Nothing to it, and it beats starting the program anew. However, if you started this command (accidentally, of course) with a digit (0-9), BASIC would think you are editing the program and set all variables to zero. When the correction process is complete, a corrected list is printed to provide a good "audit trail".

Line 420 Makes a quick pass and adds the amount of all checks written. Note that S has been defined as double precision in Line number 210.

Line 430-450 Do the calculations for the new checkbook balance in case you are not going to run CKWRTDAT/TWO immediately.

Line 460-470 Gives an opportunity to get another copy for your use, if you desire. Don't forget to advance the paper.

Line 480-500 Self-explanatory prompting message for tape or disk storage.

Line 510-550 Saves data on tape.

Line 560-610 Saves data on disk.

Line 620 In my early days, tape input/output was not to be trusted, so we wanted to see if we had a good "dump". If we did not, we would stop the program, type "GOTO480" and try again while data was still in memory. This process is a carry-over from the early days.

Line 630 Jump over subroutine.

Line 640 Protective STOP.

Line 650-720 Subroutine for listing entered data - uses TAB, PRINT USING, VAL (convert from string to numerical).

Line 730-750 Asks if file is on tape or disk.

Line 760-780 Input routines for tape and disk. Notice use of array P as separate and distinct from array A so there is a true check on new input. Note use of Print Using when printing "SUM"; otherwise all sorts of digits show up. You also get a screen listing as data is input. I like the screen to show something is happening and that the computer is not "hung-up".

Line 870 Redefines variables within the same program to show it can be done. This segment of the program uses different variables so there is no carry-over from memory. All data comes from input of tape or disk! This is the check of "dump" referred to earlier. Sum is still Double Precision from earlier definition.

Line 880 Stops program until you have the preprinted checks inserted and aligned. When this statement is at this part of the program, all checks are written without stopping. If you insert the checks one at a time, place this line between Line 970 and 980 and call it Line 975. Also change "GOTO" in Line 950 to Line 975.

Line 890-970 Since this program can actually print your checks, this is the manner in which manually written checks (between computer sessions) are handled. If a check number has been used, data has been entered in the first part of the program, but no check will be written. This segment strips off manually written checks, and when the current check number is reached, you get to Line 930. This line assigns values to variables used in the actual printing of the checks.

Line 980 to end Subroutine for printing checks. Data statements are self-explanatory.

Line 980 Breaks D (a six digit number) in day, month, and year using the technique described in the Level I Manual.

Line 1000 Reads month from data. Notice procedure to skip over unwanted part of data; this is a very useful technique.

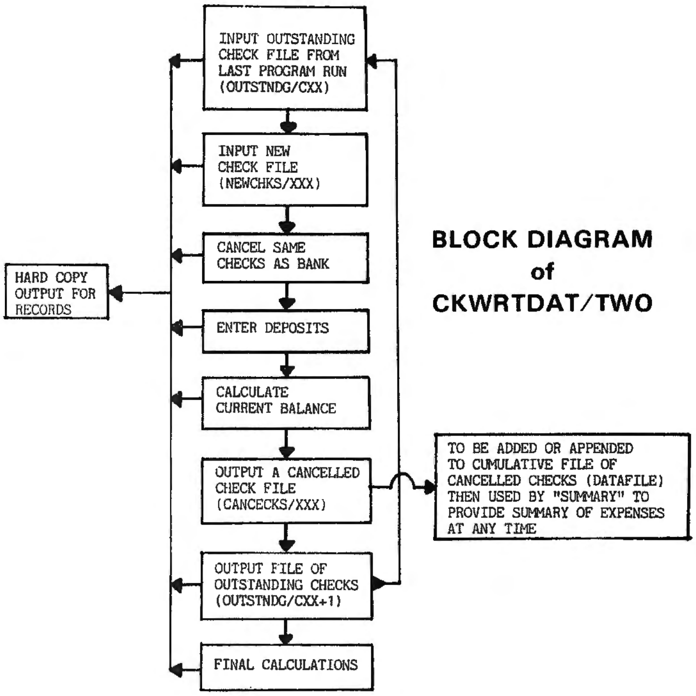
Line 1010-1070 Actually prints check up to the amount line. Note use of "PRINT USING". String (63, 42) gives a row of asterisks while String $(X,138) is a line feed.

Line 1080-1540 More exercise in digit stripping (Level I Manual) and Read-Data Statements. This segment converts the amount number to the written word. More use of an interesting routine. It does a lot, but is easy to understand. Using this method, it is possible to store data in compact numerical form and convert it back to large strings.

At this point (if you have had your own custom checks printed) you have written a batch of checks, ready for signature, and all data entered into the program is stored on tape or disk and is used for the next program. Included is a sample run of only three data items to show the output of CKWRTDAT/ONE. See Figure 1.

CKWRTDAT/TWO now uses the file 'NEWCHKS/XXX' created by CKWRTDAT/ONE in conjunction with a previously existing file "OUTSTNDG/CXX" (assuming we have run the program previously) and creates a file of cancelled checks "CANCECKS/XXX". It allows us to enter our deposits and does all the calculations necessary to maintain the balance and check out the Bank on their accuracy.

CKWRTDAT/TWO is the type of program that is easiest to describe by means of a block diagram. The program begins by the input of a file of outstanding checks which was created the last time we ran the program. For now let's assume it exists. Later we will explain how to use CKWRTDAT/ONE to create a file of outstanding checks for that time when we first begin using the program. The block diagram illustrates CSWRTDAT/TWO.


INPUT OUTSTANDING CHECK FILE FROM LAST PROGRAM RUN (OUTSTNDG/CXX)
INPUT NEW CHECK FILE (NEWCHKS/XXX)
CANCEL SAME CHECKS AS BANK
HARD COPY OUTPUT FOR RECORDS
ENTER DEPOSITS
CALCULATE CURRENT BALANCE
OUTPUT A CANCELLED CHECK FILE (CANCECKS/XXX)
TO BE ADDED OR APPENDED TO CUMULATIVE FILE OF CANCELLED CHECKS (DATAFILE) THEN USED BY "SUMMARY" TO PROVIDE SUMMARY OF EXPENSES AT ANY TIME
OUTPUT FILE OF OUTSTANDING CHECKS (OUTSTNDG/CXX+1)
FINAL CALCULATIONS

BLOCK DIAGRAM of CKWRTDAT/TWO

We will now explain CKWRTDAT/TWO.

Line 100-200 Self-evident.

Line 210-220 Reserves 5000 bytes of memory for string storage and defines the variables beginning with A, D, G, P, and W as string variables. Dimensions array A. You may have to someday adjust these depending on the number of transactions in any one session.

Line 230-250 A reminder to update filespecs if you are using a disk system. The procedure is to press "BREAK", then edit Lines 290, 460, 970, and 1080. After editing, start the program again (RUN).

Line 260 Input data in string form (i.e. March 1 1979).

Line 270-290 Self-explanatory.

Line 300-350 Disk input routine for "OUTSTNDG/CXX" file. If you are on a tape system, delete lines 300 and 350. By editing, insert a minus (-) sign between the # and the 1 (change "INPUT#1," to "INPUT #-1,") in lines 310 and 340. See how similar disk sequential files are to cassette files. Naturally, change references (in text) from "disk" to "cassette".

Line 360-400 Lprints the first four lines of the hardcopy output.

Line 410 Goes to heading printing subroutine.

Line 420 Loop to print data in array A from N=1 to I (number of items).

Line 430 Goes to subroutine to print "End of List".

Line 440 Self-explanatory.

Line 450-540 Input routine for "NEWCHKS" file. As before, we can modify these lines for cassette input by deleting lines 470 and 540 and by editing lines 480 and 520. Notice the technique of increasing I (number of items in array A) by the new number of items (K). T is set equal to I before we add K to I. So when we "loop" to fill the array, we increase "N" from T+1 to I (where it now equals I=K).

Line 550-590 As above, this part prints data on the screen and line printer.

Line 600 A carry-over from the days before the "Buffered Cable" when mysterious bit changes occured. This was an effort to flag attention if any changes occurred that could cause some real damage.

Line 610-660 Routine to cancel checks. When a check number is input, Line 620-630 loop through all the checks to see if we have a match. If we have a match, A(N,5) is set equal to one (1). This is the "cancel flag" I spoke of earlier. There are many ways to accomplish this end, but I like this one. If no match is found, error line comes into play.

Line 670-730 As before, routine for printing video and the line printer.

Line 740-830 Deposits are now "input" to the program. By now, if you have really been reading the BASIC program, this is easy stuff.

Line 840-880 Routine for printing to video and line printer.

Line 890-900 Calculates new balance (BN) from old balance (BO) plus deposits (E) less sum of "NEWCHKS" (S2). BO was input from the file of outstanding checks in Line 300. S2 was the calculation in Line 510 while E came from Line 760.

Line 910-970 Routine for printing the "OUTSTNDG" checks file. Notice how it looks at the cancel flag in Line 880 looking for uncancelled checks. S3 (sum of outstanding checks) is also calculated at this time.

Line 980-1020 Pass through the check list to count the number(s) of cancelled checks.

Line 1030-1070 Routine for disk storage of the "CANCECKS" file. To modify for cassette storage, change Line 1030 to PRINT #-1,S and delete the "CLOSE" statement in Line 1070 by editing. Also in Line 1060 change PRINT #1, to PRINT #-1, and change all the ;",";s to only a comma.

Line 1080-1160 As before, routine for disk storage of "OUTSTNDG/CXX" file. For cassette storage, change line 1120 to "PRINT #-1,D,M,BN" and Line 1150 to PRINT #-1 and all ;",";s to a comma as described above. I lied in Line 1110 because D, M, and BN are not really saved until Line 1120, as you can see. Also for cassette storage, the "CLOSE" must be deleted from Line 1160.

Line 1170-1330 The actual balancing of the checkbook. All variables used have been previously identified and are readily discernable from the program. You've got to admit - this takes all the work out of this balancing act.

Line 3000 to end Subroutines are adequately described by the remarks.

A sample run of CKWRTDAT/TWO using dummy data is included to illustrate the program. See Figure 2.

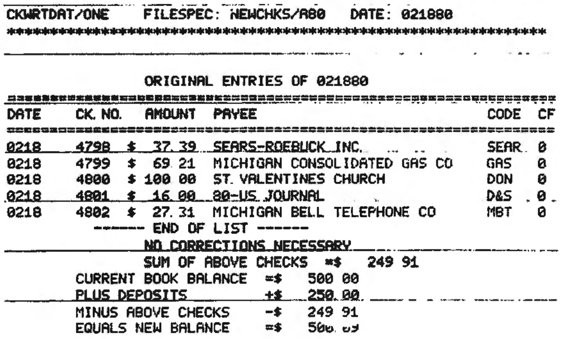
Now if you're still with us and have typed in these programs and actually used them, you can see what beauties they really are, but don't stop, we've got more work to do.

If you are working on a disk system, your only problem is to change the filespecs every program run. For CKWRTDAT/ONE, I use the format NEWCHKS/AYY, /BYY,/CYY, etc., where YY equals year. This gives a possible 26 runs before we slightly change the format. If you run this program on the first and the fifteenth of the month, you will only have 24 files, and will not run out of letters. For CKWRTDAT/TWO, I use "OUTSTNDG/CXX" starting with XX=00 and going to XX=99. Again, if you run this program twice a month, you have more than enough filespec variations. I have a "DATAFILE" file and "CANCECKS/XXX" where XXX= a two or three letter abbreviation of the month plus a trailing 1 or 2 for the first or second run of the month, if needed. "DATAFILE" is the year cumulative file to which "CANCECKS/XXX" is appended. After appending, "CANCECKS/XXX" is killed so not to clutter the disk with unused files.

When you are on a disk system, it is so easy to keep all the files separate and in order, but if you're on a cassette data system, it requires more thought and pencil and paper to keep track of your tape index numbers. You can have a separate cassette for data storage of NEWCHKS, OUTSTNDG/CXX, and CANCECKS/XXX and keep track of each different file by noting the tape index numbers with your pencil and pad. In order for the cassette data storage system to be efficient, you must keep as many data files as you can on one tape or you can keep one data file from each program run on one tape and use one heck of a lot of tapes.

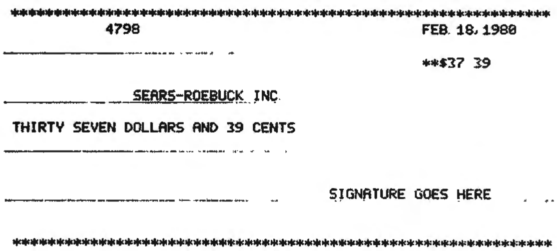
It does sometimes happen that when all data is in a file on disk or tape that we discover an error or that we desire to change a budget code assignment on some entry. We have included a short program "CORRECT/ION" which allows us to input a file, make the desired corrections, and resave the file. The program is set up for disk files, but can be adapted for tape as described in the analysis of "CKWRTDAT/TWO". Note that on disk storage you must update the filespec for whatever file you are correcting, and you must also edit the first line of file input and output to be compatible with the format of the file with which you are working. For safety, the corrected file is saved under a new filespec. After verifying the new file, the old file is killed, and then the new file is RENAME'd to the old filespec.

We will conclude this article in the next issue with the programs SUMMARY/ONE and SUMMARY/TWO. ●



```
CKWRTDAT/ONE    FILESPEC: NEWCHKS/A80   DATE: 021880
****************************************************************

               ORIGINAL ENTRIES OF 021880
================================================================
DATE    CK. NO.   AMOUNT  PAYEE                         CODE  CF
================================================================
0218    4798   $  37.39   SEARS-ROEBUCK INC.            SEAR  0
0218    4799   $  69 21   MICHIGAN CONSOLIDATED GAS CO  GAS   0
0218    4800   $ 100 00   ST. VALENTINES CHURCH         DON   0
0218    4801   $  16.00   80-US JOURNAL                 D&S   0
0218    4802   $  27.31   MICHIGAN BELL TELEPHONE CO    MBT   0
          ------ END OF LIST ------
               NO CORRECTIONS NECESSARY
               SUM OF ABOVE CHECKS  =$   249 91
        CURRENT BOOK BALANCE  =$   500 00
        PLUS DEPOSITS         +$   250.00
        MINUS ABOVE CHECKS    -$   249 91
        EQUALS NEW BALANCE    =$   500.09
```

Figure 1



```
******************************************************************
          4798                                  FEB. 18, 1980

                                                **$37 39

                SEARS-ROEBUCK INC.

THIRTY SEVEN DOLLARS AND 39 CENTS



                                       SIGNATURE GOES HERE

******************************************************************
```

Sample Check

```
                    C K W R T D A T / T W O
*********************************************************************
LAST SESSION WAS ON 021880
DATE OF THIS SESSION IS FEBRUARY 18 1980


OUTSTANDING CHECKS AS OF 021880 SESSION
=================================================================
DATE     CK. NO.  AMOUNT  PAYEE                            CODE  CF
=================================================================
0109     4782  $   86.00 ROBERT A. KELLY D D. S            DENT  0
0113     4788  $   25.30 BORDENS                           MILK  0
0120     4790  $   84.73 A & P                             FOOD  0
0129     4792  $    7 41 AU GRES TELEPHONE CO.             UN    0
0129     4794  $   82.10 SHELL OIL CO.                     CAR   0
                       END OF LIST
NUMBER OF ENTRIES = 5    PREVIOUS BALANCE =*$ 500 00



NEW CHECKS AS OF FEBRUARY 18 1980
=================================================================
DATE     CK. NO.  AMOUNT  PAYEE                            CODE  CF
=================================================================
0218     4798  $   37.39 SEARS-ROEBUCK INC.                SEAR  0
0218     4799  $   69.21 MICHIGAN CONSOLIDATED GAS CO      GAS   0
0218     4800  $  100.00 ST. VALENTINES CHURCH             DON   0
0218     4801  $   16.00 80-US JOURNAL                     D&S   0
0218     4802  $   27.31 MICHIGAN BELL TELEPHONE CO.       MBT   0
                       END OF LIST
CKS. WRITTEN ON 021880            NUMBER OF ENTRIES= 5
              SUM=$ 249.91



CANCELED CHECKS AS OF FEBRUARY 18 1980
=================================================================
DATE     CK. NO.  AMOUNT  PAYEE                            CODE  CF
=================================================================
0109     4782  $   86.00 ROBERT A  KELLY D D. S.           DENT  1
0113     4788  $   25.30 BORDENS                           MILK  1
0120     4790  $   84.73 A & P                             FOOD  1
0129     4792  $    7 41 AU GRES TELEPHONE CO.             UN    1
0129.    4794  $   82.10 SHELL OIL CO.                     CAR   1
                       END OF LIST
NUMBER OF ITEMS  5
```

Sample Output of CKWRTDAT/TWO

```
LIST OF DEPOSITS
=================================================================
DATE     CK. NO.  AMOUNT  PAYEE                            CODE  CF
=================================================================
0217     0      $  250.00 DEPOSIT                          DEP   99
-----------------------------------------------------------------
TOTAL DEPOSITS =$  250.00



CALCULATION OF CURRENT BALANCE
=================================================================
PREVIOUS BALANCE ($   500 00)+DEPOSITS ($   250.00)
-SUM OF CHECKS ($   249.91)=CURRENT BALANCE ($   500 09)
=================================================================



OUTSTANDING CHECKS AS OF FEBRUARY 18 1980 SESSION
=================================================================
DATE     CK. NO.  AMOUNT  PAYEE                            CODE  CF
=================================================================
0218     4798  $   37.39 SEARS-ROEBUCK INC                 SEAR  0
0218     4799  $   69.21 MICHIGAN CONSOLIDATED GAS CO.     GAS   0
0218     4800  $  100 00 ST VALENTINES CHURCH              DON   0
0218     4801  $   16 00 80-US JOURNAL                     D&S   0
0218     4802  $   27 31 MICHIGAN BELL TELEPHONE CO.       MBT   0
               $  249.91
                       END OF LIST



WE WILL NOW CHECK THE BANK BALANCE
          CURRENT PROGRAM BALANCE        +$ 500.09
          ADD OUTSTANDING CHECKS         +$ 249 91
          LESS DEPOSITS NOT CREDITED     -$ 250.00
          PROJECTED PROGRAM BALANCE      =$ 500.00
          CURRENT BANK BALANCE           =$ 500.00
          CORRECTION TO PROGRAM BALANCE  =$   0.00
END OF FEBRUARY 18 1980 SESSION
*********************************************************************
```

```
100 REM************* C K W R T D A T / O N E *******************
110 REM*****************    VERSION 4.6    *********************
120 CLS
130 PRINT"*****************************************************"
140 PRINT"***     R.A.SHMINA                      MARCH 1979   ***"
150 PRINT"***     14030 ARNOLD                                 ***"
160 PRINT"***     REDFORD,MICHIGAN 48239                       ***"
170 PRINT"***                                                  ***"
180 PRINT"***     + + + +      P A R T   I    + + + +          ***"
190 PRINT"***              C K W R T D A T / O N E             ***"
200 PRINT"*****************************************************"
210 CLEAR3000:DEFDBL S:DEFSTR A,C,D,P:DIM A(150,5):DIM P(100,5)
220 PRINT:INPUT"ENTER DATE OF POSTING = (MMDDYY) =";D:I=1:PRINT
230 INPUT"ENTER COMPLETE FILESPEC IDENTIFCATION";F$
240 PRINT:PRINT"NOW ENTER CHECK DATA -- TO EXIT THIS MODE ENTER
     'Ø,Ø,Ø,Ø,Ø'"
250 PRINT"DON'T FORGET THE COMMAS":PRINT
260 PRINT"(DATE),(CK.NO.),(AMOUNT),(PAYEE),(CODE)"
270 INPUT A(I,Ø),A(I,1),A(I,2),A(I,3),A(I,4)
280 A(I,5)="Ø"
290 IFA(I,1)="Ø"ANDA(I,2)="Ø"ANDA(I,3)="Ø"ANDA(I,4)="Ø"THEN31Ø
300 I=I+1:GOTO 27Ø
310 I=I-1:PRINT"END OF INPUT","I =";I:REM **STRIP OFF ØØØØ ENTR
    Y**
320 SUM=Ø:LPRINTTAB(Ø)"CKWRTDAT/ONE";TAB(16)"FILESPEC: ";F$;TAB
    (4Ø)"DATE: ";D
   :LPRINTSTRING$(63,42):LPRINTSTRING$(2,138)
   :LPRINTTAB(16)"ORIGINAL ENTRIES OF ";D:GOSUB 65Ø
330 INPUT"ANY CORRECTIONS NECESSARY? TYPE (YES) OR (NO)";Z$
340 IFZ$="YES"THEN 36Ø ELSE IFZ$="NO"THEN 4ØØ
350 GOTO 33Ø
360 PRINT"ENTER 'Ø,Ø,Ø' TO EXIT THIS MODE"
   :INPUT"ENTER I,J,&T$ (T$=DESIRED CORRECTION)";F,G,T$
370 A(F,G)=T$
380 IF F=Ø AND G=Ø AND T$="Ø" THEN LPRINTSTRING$(4,138)
   :LPRINTTAB(16)"CORRECTED LIST":GOTO 41Ø
390 GOTO 36Ø
400 LPRINTTAB(16)"NO CORRECTIONS NECESSARY":GOTO 42Ø
410 GOSUB 65Ø
420 FORN=1TO I:SUM=SUM+VAL(A(N,2)):NEXTN:PRINT"SUM OF ABOVE CHE
    CKS= $";SUM
430 INPUT"ENTER CURRENT BOOK BALANCE";BB
440 INPUT"ENTER ANY DEPOSITS";E:BN=BB+E-SUM
450 LPRINTTAB(16)"SUM OF ABOVE CHECKS   =";USING"$##,###.##";SUM
   :LPRINTTAB(8)"CURRENT BOOK BALANCE   =";USING"$##,###.##";BB
   :LPRINTTAB(8)"PLUS DEPOSITS          +";USING"$##,###.##";E
455 LPRINTTAB(8)"MINUS ABOVE CHECKS     -";USING"$##,###.##";SUM
   :LPRINTTAB(8)"EQUALS NEW BALANCE     =";USING"$##,###.##";BN
460 Z$="":INPUT"DO YOU WANT ANOTHER COPY OF ENTRIES
                         ( A D V A N C E   P A P E R )";Z$
```

```
470 IF LEFT$(Z$,1)="Y"THEN 320 ELSE IF LEFT$(Z$,1)<>"N"THEN 460
480 PRINT"ARE YOU SAVING DATA ON TAPE OR DISK (T OR D)?"
490 Z$=INKEY$:IF Z$="" THEN 490
500 IF Z$="D" THEN 560
510 INPUT"PRESS 'ENTER' WHEN CASETTE IS READY <RECORD>";Z$
520 PRINT#-1,D,I,SUM:PRINT"SAVED D,I,SUM ----"
530 FOR N=1 TO I
540 PRINT#-1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5)
    :PRINT N;" ";
550 NEXT N:PRINT"DATA ON TAPE ..":GOTO 620
560 OPEN "O",1,F$
570 PRINT#1,D;",";I;",";SUM:PRINT"SAVED D,I,SUM--"
580 FORN=1TO I
590 PRINT#1,A(N,0);",";A(N,1);",";A(N,2);",";A(N,3);",";A(N,4);
   ",";A(N,5):PRINT N;" ";
600 NEXT N
610 CLOSE:PRINT"NEWCHKS FILE CLOSED !  DATA ON DISK"
620 INPUT"PRESS 'ENTER' TO CONTINUE --- WE WILL NOW CHECK THE '
    DUMP'
    BY 'INPUTING' IT INTO CHECKWRITER PROGRAM --- REWIND TAPE";
    Z$
630 GOTO 730
640 STOP
650  PRINTTAB(0)"DATE";TAB(8)"CK.NO.",TAB(16)"AMOUNT";TAB(24)
    "PAYEE";TAB(56)"CODE";TAB(62)"CF"
660 LPRINTSTRING$(64,61)
670 LPRINTTAB(0)"DATE";TAB(8)"CK.NO.";TAB(16)"AMOUNT";TAB(24)
    "PAYEE";TAB(56)"CODE";TAB(62)"CF"
680 LPRINTSTRING$(64,61)
690 FOR N=1TO I
700 LPRINTTAB(0)A(N,0);TAB(8)A(N,1);TAB(14)USING"$####.##";VAL(
    A(N,2));
    :LPRINTTAB(24)A(N,3);TAB(56)A(N,4);TAB(62)A(N,5)
710 PRINTTAB(0)A(N,0);TAB(8)A(N,1);TAB(14)USING"$####.##";VAL(A
    (N,2));
     :PRINTTAB(24)A(N,3);TAB(56)A(N,4);TAB(62)A(N,5)
720 NEXTN:LPRINTTAB(10)"------ END OF LIST ------":PRINT"END OF
     LIST":RETURN
730 CLS:PRINT"ARE YOU USING TAPE OR DISK (T OR D)?"
740 Z$=INKEY$:IF Z$="" THEN 740
750 IF Z$="D" THEN 790
760 PRINT:INPUT"CASETTE READY <PLAY> PRESS 'ENTER'";Z$
770 INPUT #-1,D,I,SUM:PRINTD,I,SUM
780 FOR E=1 TO I
    :INPUT#-1,P(E,0),P(E,1),P(E,2),P(E,3),P(E,4),P(E,5)
    :PRINT E;" ";
    :NEXT E
    :GOTO 860
790 OPEN "I",1,F$
800 INPUT#1,D$,I,SUM:PRINT D$,I,USING"$$#####.##";SUM
810 FOR E=1TO I
820 INPUT#1,P(E,0),P(E,1),P(E,2),P(E,3),P(E,4),P(E,5)
830 PRINTTAB(0)P(E,0);TAB(8)P(E,1);TAB(16)P(E,2);TAB(24)P(E,3);
     TAB(56)P(E,4);TAB(62)P(E,5)
840 NEXT E
850 CLOSE
860 INPUT"IF WE HAVE A GOOD 'DUMP' PRESS 'ENTER' ---
     ELSE PRESS 'BREAK' & TRY ANOTHER 'DUMP'";Z$
870 DEFSNG A,C,D
880 PRINT:INPUT"LOAD BLANK CHECKS INTO PRINTER -------
     PRESS 'ENTER' WHEN READY ------";Z$
890 INPUT"ENTER CURRENT UNUSED NUMBER IN CK.NO. SEQUENCE";CN
900 IF CN < VAL(P(1,1)) OR CN > VAL(P(I,1))THEN 890 ELSE 910
910 C=1
920 PRINT"LIST OF MANUALLY ENTERED CHECKS"
930 PRINT"C=";C;" ";"ACTUAL CN= ";VAL(P(C,1));" ";"UNUSED CN=";
    CN
940 IF VAL(P(C,1)) >=CN THEN 950 ELSE C=C+1:GOTO 930
950 AMOUNT=VAL(P(C,2)):PAYEE$=P(C,3):D=VAL(D$):GOSUB 980
960 C=C+1:IF C>I THEN 970  ELSE 930
970 PRINT"END OF PROGRAM":END:PRINT
980 DM=INT(D/10000):DD=INT((D-DM*10000)/100):DY=INT(D-(DM*100+D
    D)*100)
990 RESTORE
1000 FOR N=1 TO DM:READ MO$:NEXT N
1010 PRINTSTRING$(63,42):LPRINTSTRING$(63,42)
1020 LPRINTTAB(10) VAL(P(C,1));TAB(48)MO$;USING"##";DD;
     :LPRINT",";USING"19##";DY:LPRINTSTRING$(1,138)
1030 PRINT VAL(P(C,1)),,,MO$;USING"##";DD;
     :PRINT",";USING"19##";DY:PRINT
1040 LPRINT TAB(48)USING"**$##.##";AMOUNT:LPRINTSTRING$(1,138)
1050 PRINTTAB(48)USING"**$##.##";AMOUNT:PRINT
1060 LPRINT TAB(14)PAYEE$:LPRINTSTRING$(1,138)
1070 PRINTTAB(14)PAYEE$:PRINT
1080 IF AMOUNT<=0 THEN 1450
1090 THOUS=INT(AM/1000)
1100 IF THOUS>9THEN 1450
1110 IF THOUS=0 THEN 1140
1120 SKIP=THOUS+12:GOSUB 1330
1130 LPRINT "THOUSAND ";:PRINT"THOUSAND ";
1140 AH=AM-TH*1000:HUNS=INT(AH/100)
1150 IF HUNS=0 THEN 1180
1160 SKIP=HUN+12:GOSUB 1330
1170 LPRINT "HUNDRED ";:PRINT"HUNDRED ";
1180 AT=AH-HU*100:TENS=INT(AT)
1190 IF TENS>0 THEN 1220
1200 IF AMOUNT>=1 THEN 1230
1210 GOTO 1240
1220 SKIP=TENS+12:GOSUB 1330
1230 LPRINT "DOLLARS ";:PRINT"DOLLARS ";
1240 AC=AT-TENS
1250 IF AMOUNT<1 THEN 1280
1260 LPRINT "AND";:PRINT"AND";
1270 AC=INT((AC+.005)*100)/100
1280 AC=AC*100
1290 LPRINT AC;"CENTS":PRINT AC;"CENTS"
1300 PRINT:PRINTTAB(40)"SIGNATURE GOES HERE":PRINT
1310 PRINTSTRING$(63,42)
     :LPRINTSTRING$(3,138);TAB(40)"SIGNATURE GOES HERE"
     :LPRINTSTRING$(2,138);STRING$(63,42)
```

```
1320 RETURN
1330 IF SKIP)33 THEN 1380
1340 RESTORE
1350 FOR N=1 TO SK:READ DA$:NEXT N
1360 LPRINT DA$;" ";:PRINT DA$;" ";
1370 RETURN
1380 RESTORE
1390 FOR N=1 TO(SKIP-32)/10+32:READ DA$:NEXT N
1400 LPRINT DA$;" ";:PRINT DA$;" ";
1410 A1=TENS-INT(TENS/10)*10
1420 IF A1=0 THEN 1440
1430 A1=A1+12:GOTO1470
1440 RETURN
1450 LPRINTTAB(20)"*******----V O I D----*******"
1460 GOTO 1310
1470 RESTORE
1480 FOR N=1 TO A1:READ DA$:NEXT N
1490 LPRINT DA$;" ";:PRINT DA$;" ";
1500 RETURN
1510 STOP
1520 DATA"JAN.","FEB.","MAR.","APR.","MAY ","JUN.","JUL.",
   "AUG.","SEPT.","OCT.","NOV.","DEC."
1530 DATA"ONE","TWO","THREE","FOUR","FIVE","SIX","SEVEN",
     "EIGHT","NINE","TEN","ELEVEN","TWELVE","THIRTEEN",
     "FOURTEEN","FIFTEEN","SIXTEEN","SEVENTEEN","EIGHTEEN",
     "NINTEEN"
1540 DATA"TWENTY","THIRTY","FORTY","FIFTY","SIXTY","SEVENTY",
   "EIGHTY","NINETY"
```

```
100 REM****************  C K W R T D A T / T W O  ****************
110 REM****************    V E R S I O N  4.6     ****************
120 CLS
130 PRINT"*******************************************************"
140 PRINT"**     R.A.SHMINA                         MARCH 1979  ***"
150 PRINT"**     14030 ARNOLD                                   ***"
160 PRINT"**     REDFORD,MICHIGAN 48239                         ***"
170 PRINT"**                                                    ***"
180 PRINT"**      + + + +     P A R T   II     + + + +          ***"
190 PRINT"**               C K W R T D A T / T W O              ***"
200 PRINT"*******************************************************"
210 CLEAR 5000
220 DEFSTR A,D,G,P,W:DIM A(200,5)
230 PRINT@519,"TO UPDATE, PRESS 'BREAK' -- THEN EDIT -- THEN RE
    RUN"
240 PRINT@647,"UPDATE   'OUTSTNDG/C(XX)',   'OUTSTNDG/C(XX+1)',
               'NEWCHKS/(X?)',   AND 'CANCECKS/(?)'":PRINT:
   :PRINT"        UPDATED L#'S 300, 470, 1030, & 1120"
250 PRINT:INPUT"                    PRESS 'ENTER' WHEN READY";Z$
260 PRINT:INPUT"ENTER DATE OF ENTRIES (I.E. MARCH 1 1979)";D
270 PRINT:PRINT "PREPARE 'OUTSTANDING CHECKS' DISK (OR TAPE (PL
    AY))"
```

```
280 PRINT:INPUT"PRESS 'ENTER' WHEN READY";Z$
290 CLS:PRINT "READING DISK"
300 OPEN "I",1,"OUTSTNDG/C00
310 INPUT#1,W,I,BO
320 PRINT "W=";W,"I=";I,USING"BO=$#####.##";BO
330 PRINT "READING DISK AGAIN"
340 FORN=1TOI:INPUT#1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5)
    :PRINT N;" ";
    :NEXT N
350 CLOSE
360 LPRINTTAB(22)"C K W R T D A T / T W O":LPRINTSTRING$(65,42)
370 PRINT "DONE READING DISK":PRINT"LAST SESSION WAS ON  ";W
    :PRINT"DATE OF THIS SESSION IS ";D
380 LPRINT"LAST SESSION WAS ON ";W
    :LPRINT"DATE OF THIS SESSION IS ";D:LPRINTSTRING$(2,138)
390 PRINT"----- OUTSTANDING CHECKS AS OF "W;" SESSION ----"
400 LPRINT"OUTSTANDING CHECKS AS OF "W;" SESSION"
410 GOSUB 3000 REM PRINT HEADING
420 FOR N=1 TO I:GOSUB 4000:NEXT N
430 GOSUB 5000   REM  PRINT ENDING
440 LPRINT"NUMBER OF ENTRIES =";I;"    ";"PREVIOUS BALANCE =";US
    ING"*$#####.##";BO
    :LPRINTSTRING$(5,138)
450 INPUT"PREPARE TO ENTER NEW CHECKS ---- (PLAY)
    PRESS 'ENTER' WHEN READY";Z$
460 T=I:CLS:PRINT"READING DISK ------"
470 OPEN "I",1,"NEWCHKS/A80"
480 INPUT#1,D1,K,SUM
490 PRINT"D1=";D1,"K=";K,"SUM=";SUM
500 PRINT "READING ";K;" ITEMS OF DATA FROM DISK"
510 I=I+K:FOR N= T+1 TO I
520 INPUT#1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5)
    :S2=S2+VAL(A(N,2)):PRINT N;
530 NEXT N: PRINT"DATA ALL READ":PRINT"SUM OF A(N,2)=";S2
540 CLOSE
550 PRINT"---------- NEW CHECKS AS OF ";D;" --------------"
560 LPRINT"NEW CHECKS AS OF ";D
570 GOSUB 3000  REM HEADING
580 FOR N=T+1 TO I:GOSUB 4000:NEXT N:GOSUB 5000
590 LPRINT"CKS. WRITTEN ON ";D1,"NUMBER OF ENTRIES=";K
    :LPRINTTAB(10)"SUM=$";S2:LPRINTSTRING$(5,138)
600 IF S2()SUM THEN PRINT"ERRROR IN LOADING -- CHECK"
610 PRINT"IF NO CHECKS ARE TO BE CANCELED, ENTER ZERO (0)"
620 INPUT"ENTER NUMBER OF CANCELED CHECK";G:IFG="0"THEN 670
630 FOR N=1TO I:IF A(N,1)=G THEN 650
640 NEXT N:GOTO 660
650 A(N,5)="1":GOTO620
660 PRINT "ERROR ---- CHECK NO. NOT IN USE -----------":GOTO620
670 PRINT"--------- CANCELED CHECKS AS OF ";D;"------------"
680 LPRINT"CANCELED CHECKS AS OF ";D
690 GOSUB 3000  REM HEADING
700 FOR N=1TO I:IFA(N,5)="1"THEN 710 ELSE 720
710 S=S+1:GOSUB 4000
720 NEXT N:PRINT"S=";S
730 GOSUB 5000:LPRINT"NUMBER OF ITEMS ";S:LPRINTSTRING$(5,138)
740 PRINT:PRINT"DO YOU HAVE ANY DEPOSITS TO ENTER ?"
```

```
750 Z1$=INKEY$:IF Z1$="" THEN 750
760 IF Z1$="Y" THEN 780 ELSE IF Z1$="N" THEN 840
770 IF Z1$<>"Y" OR Z1$<>"N" THEN 740
780 PRINT"ENTER ALL DEPOSITS USING THIS FORMAT
    <DATE> , <AMOUNT> , < TYPE 'DEPOSIT' >"
790 I=I+1:INPUT A(I,Ø),A(I,2),A(I,3)
800 A(I,1)="Ø":A(I,4)="DEP":A(I,5)="99"
810 E=E+VAL(A(I,2))
820 INPUT"MORE DEPOSITS? -- TYPE <YES> OR <NO>";Z$
830 IF Z$="YES"GOTO 790 ELSE 840
840 PRINT"LIST OF DEPOSITS":LPRINT"LIST OF DEPOSITS":GOSUB 3ØØØ
850 FOR N=1 TO I:IF A(N,5)="99" THEN 860 ELSE 870
860 GOSUB 4ØØØ
870 NEXT N:LPRINTSTRING$(63,45)
880 PRINT"TOTAL DEPOSITS ";E
   :LPRINT"TOTAL DEPOSITS =";USING"$#####.##";E:LPRINTSTRING$(5
    ,138)
890 LPRINT"CALCULATION OF CURRENT BALANCE"
900 BN=BO+E-S2:LPRINTSTRING$(63,61):PRINTUSING"CURRENT BALANCE
    =$##,###.##";BN
905 LPRINTUSING"PREVIOUS BALANCE ($##,###.##)+DEPOSITS ($##,###
    .##)-SUM OF CHECKS ($##,###.##)=CURRENT BALANCE ($##,###.##)
    ";BO,E,S2,BN
907 LPRINTSTRING$(63,61):LPRINTSTRING$(3,138)
910 PRINT"---- OUTSTANDING CHECKS AS OF ";D;" SESSION ----"
920 LPRINT"OUTSTANDING CHECKS AS OF ";D;" SESSION"
930 GOSUB 3ØØØ
940 FOR N=1 TO I:IF A(N,5)="Ø"THEN 950 ELSE 960
950 GOSUB 4ØØØ:S3=S3+VAL(A(N,2))
960 NEXTN:LPRINTTAB(14)USING"$#####.##";S3
970 GOSUB 5ØØØ:LPRINTSTRING$(3,138)
980 PRINT"LOAD A 'CANCELED CHECKS' TAPE  <RECORD>"
990 INPUT"PRESS 'ENTER' WHEN READY";Z$
1ØØØ PRINT"WRITING TAPE & DISK --- SAVING CANCELED CHECKS & DEP
    OSITS":S=Ø
1Ø1Ø FORN=1TO I:IFA(N,5)="1" OR A(N,5)="99" THEN S=S+1 ELSE 1Ø2
    Ø
1Ø2Ø NEXT N
1Ø3Ø OPEN "O",1,"CANCECKS/CØ1":PRINT#1,S
1Ø4Ø FORN=1TO I:IF A(N,5)="1" OR A(N,5)="99" THEN 1Ø5ØELSE 1Ø7Ø
1Ø5Ø PRINTTAB(Ø)A(N,Ø);TAB(8)A(N,1);TAB(16)A(N,2);TAB(24)A(N,3)
    ;
    TAB(56)A(N,4);TAB(62)A(N,5)
1Ø6Ø PRINT#1,A(N,Ø);",";A(N,1);",";A(N,2);",";A(N,3);",";A(N,4)
    ;",";A(N,5)
1Ø7Ø NEXT N:CLOSE:PRINT"DONE WRITING TAPE & DISK"
1Ø8Ø PRINT"NOW PREPARE TO SAVE OUTSTANDING CHECKS ---
     LOAD AN 'OUTSTANDING CHECKS' TAPE <RECORD>---
      PRESS 'ENTER' WHEN READY":INPUT Z$
1Ø9Ø CLS:PRINT"WRITING TAPE & DISK -- SAVING OUTSTANDING CHECKS
    ":M=Ø
11ØØ FORN=1TO I:IFA(N,5)="Ø"THEN M=M+1 ELSE 111Ø
111Ø NEXTN:PRINT"SAVED DATE-";D;" M=";M;" BAL";BN
112Ø OPEN "O",1,"OUTSTNDG/CØ1":PRINT#1,D;",";M;",";BN
113Ø FOR N=1TO I:IF A(N,5)="Ø"THEN 114Ø ELSE 116Ø
114Ø PRINTTAB(Ø)A(N,Ø);TAB(8)A(N,1);TAB(16)A(N,2);TAB(24)A(N,3)
    ;TAB(56)A(N,4);TAB(62)A(N,5)
115Ø PRINT#1,A(N,Ø);",";A(N,1);",";A(N,2);",";A(N,3);",";A(N,4)
    ;",";A(N,5)
116Ø NEXT N:CLOSE:PRINT"DONE WRITING TAPE & DISK"
117Ø PRINT"WE WILL NOW CHECK THE BANK BALANCE"
    :LPRINT"WE WILL NOW CHECK THE BANK BALANCE"
    :INPUT"ENTER CURRENT BANK BALANCE";BB
    :INPUT"ENTER DEPOSITS NOT CREDITED BY BANK";E1
118Ø BP=BN+S3-E1:C=BB-BP
119Ø PRINTTAB(1Ø)"CURRENT PROGRAM BALANCE        +";USING"$####
    .##";BN
12ØØ LPRINTTAB(1Ø)"CURRENT PROGRAM BALANCE        +";USING"$###
    #.##";BN
121Ø PRINTTAB(1Ø)"ADD OUTSTANDING CHECKS         +";USING"$####
    .##";S3
122Ø LPRINTTAB(1Ø)"ADD OUTSTANDING CHECKS         +";USING"$###
    #.##";S3
123Ø PRINTTAB(1Ø)"LESS DEPOSITS NOT CREDITED     -";USING"$####
    .##";E1
124Ø LPRINTTAB(1Ø)"LESS DEPOSITS NOT CREDITED     -";USING"$###
    #.##";E1
125Ø PRINTTAB(1Ø)"PROJECTED PROGRAM BALANCE      =";USING"$####
    .##";BP
126Ø LPRINTTAB(1Ø)"PROJECTED PROGRAM BALANCE      =";USING"$###
    #.##";BP
127Ø PRINTTAB(1Ø)"CURRENT BANK BALANCE           =";USING"$####
    .##";BB
128Ø LPRINTTAB(1Ø)"CURRENT BANK BALANCE           ="USING"$####
    .##";BB
129Ø PRINTTAB(1Ø)"CORRECTION TO PROGRAM BALANCE  =";USING"$####
    .##";C
13ØØ LPRINTTAB(1Ø)"CORRECTION TO PROGRAM BALANCE  =";USING"$###
    #.##";C
131Ø PRINTTAB(15)"END OF ";D;" SESSION"
132Ø LPRINT"END OF ";D;" SESSION"
     :LPRINTSTRING$(65,42)
     :LPRINTSTRING$(15,138)
133Ø END: REM PROTECT SUBROUTINES
3ØØØ REM --- LPRINT HEADING -----
3Ø1Ø LPRINTSTRING$(64,61)
3Ø2Ø LPRINTTAB(Ø)"DATE";TAB(8)"CK.NO.";TAB(16)"AMOUNT";
    TAB(24)"PAYEE";TAB(56)"CODE";TAB(62)"CF"
3Ø3Ø LPRINTSTRING$(64,61)
3Ø4Ø RETURN
4ØØØ REM ---- WRITE A LINE OF DATA ------
4Ø1Ø PRINTTAB(Ø)A(N,Ø);TAB(8)A(N,1);TAB(14)USING"$####.##";
    VAL(A(N,2));:PRINTTAB(24)A(N,3);TAB(56)A(N,4);TAB(62)A(N,5)
4Ø2Ø LPRINTTAB(Ø)A(N,Ø);TAB(8)A(N,1);TAB(14)USING"$#####.##";
    VAL(A(N,2));:LPRINTTAB(24)A(N,3);TAB(56)A(N,4);TAB(62)A(N,5
    )
4Ø3Ø RETURN
5ØØØ REM ------- END OF LIST -------
5Ø1Ø PRINT"---------- END OF LIST ----------"
5Ø2Ø LPRINTTAB(22)"END OF LIST"
5Ø3Ø RETURN
```

Wordprocessing

by
Special Delivery
D&D Software
2608 Landershire
Plano, TX 75023
32K DOS & Up
TRSDOS, NEWDOS
$125.00

Special Delivery

There are Pencil, Scripsit, hundreds of mailing lists, and now here comes another one. But hold on, this package is different. The WORDPROCESSING package has tremendous flexibility. It consists of two major machine language programs called Mailform and Mailrite. Also included are 4 special purpose programs called Extract, Sort, Label, and Convert. With this software you can set up a name and address list, call up a text that you have written either with Pencil or Scripsit and print custom written letters combining the information from both files.

All programs are machine language, very rapid and extremely sophisticated. Looking in detail at the programs:

Mailform

Available fields and their lengths: Name (25 chr), Company (25 chr), Address (25 chr), City (18 chr), State (2 chr), Zip (5 chr), Data or coding (2 at 14 chr each).

Language & Procedures

Machine language. In-memory storage of data, no waiting for disk I/O. Uses a screen overwrite system. Easy to enter and edit data. Full cursor control, repeating key entry by holding it down. Commands include get previous and next record, delete record(s), store and get from memory, read and write file to disk, define a search field, sort, continue search, and exit to DOS.

Comments

The program works very well. The ability to sort via any field, the speed of data entry, and the ease of editing are all very impressive. The authors have taken much care in designing this program. It is very well error trapped. If you give an invalid command, the cursor will jump to the error (e.g. missing filespec) and force you to do it correctly. It is impossible to enter past the field lengths. Warning signals are given by triggering the cassette relay, and included is a diagram for a beeper to be connected if you wish. After some effort, we got the program to crash (force in a very high record number and the program will go into ROM looking for the data), but in all likelihood, you will never go wrong entering commands or data. By defining search fields and merging files from one list to another, you have tremendous flexibility. You could take a number of lists and merge together all those within a given town; you could select to have letters sent only to company presidents that live in California. With the Extract, search, sort, and merge routines you have true data manipulation.

There are some drawbacks. You can set up a file of at most 316 names in memory at one time. Very large mailing lists would require subdivision into a number of files. The documentation is very weak. In fact, a command is shown on the video (X=extract) which is not mentioned in the 16 page manual. I would be afraid to give the manual to a non-programmer. For example, describing the Get Record command: 'If the memory buffer is empty or the record number is not numeric or from 1 to LRN (Last Record Number in buffer) inclusive, the cursor will be moved to the RECORD # field and ...' I pity the poor office worker trying to follow all that. Lastly, the commands are entered via a control key system that is awkward to use. The control key is the SHIFT AND DOWN ARROW, and then simultaneously depress the desired function key. If you have the lower case modification and an Electric Pencil control key, they may be used instead.

The second program in the package is Mailrite. It is a program that allows you to print letters, envelopes, or labels from a master mailing list (made with Mailform), and a text file made with Electric Pencil or Scripsit.

Language & Procedures

Mailrite is also machine language, very fast and cleanly written. You set the printer parameters with Mailrite. The actual locations for name, address, etc. to be inserted into the text are flagged within the text. You identify what and where additions are to be made when the text is being written with either Electric Pencil or Scripsit. You have the ability to underline passages, change to boldface (if your printer can do it!), reset margins and so on. You can halt during printing and quit or continue.

Comments

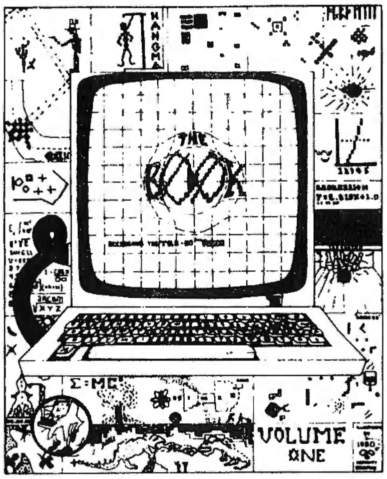
Also very sophisticated. Operates like Mailform. Some new commands added for use with the text writers to flag where the data is to be inserted.

An added routine for Radio Shack Mail list users is included. The program called Convert will take your Radio Shack mail lists and alter the data to fit the Mailform structure. A nice touch, not having to reenter all that data.

Overall Evaluation

An impressive package. Well designed and error trapped. It is quite a powerful addition for Electric Pencil and Scripsit users. If all you want is a mailing list program, you don't need it. If you don't have Electric Pencil or Scripsit, you don't want it. Documentation and user manual is rather skimpy. Plan on playing with this program quite a bit before bringing it to the office.

Cameron C. Brown



Software Review

Small Systems Software RSM for Model II

When Radio Shack first introduced the TRS-80, few of us were really concerned with those "reserved" areas of memory. For many of us, this was our first "close encounter" with a computer, and we were just trying to get used to Basic. But time went on, our skills developed, and we were ready to tackle that strange world of machine language. In the spring of '78, Small Systems Software released a program called RSM. This cassette-based monitor became a tool that virtually nobody felt they could live without. With the increasing popularity of disk drives, RSM2D was introduced to us with new features like disk I/O. Now, with the advent of the Mod II comes RSMII.

Keeping the commands similar to those we are used to using, the Mod II version has become one of the most frequently used programs in my software library. The embellishments, however, are most welcome and useful.

Perhaps the most appreciated change was making RSMII *COMPLETELY* relocatable. When you insert the self-booting disk and hit the reset key, RSM boots itself and asks you to "Enter Start Address". You may respond with any address ending with 00. Thus you can place RSMII just about anywhere in memory. The only real restrictions are that the last digits of the starting address are 00, and you don't try to put it above E000H (there simply isn't room, since the program takes 2000 hex bytes). In addition, to being relocatable, it is also totally self-contained . In other words, it does not depend upon the DOS for it's I/O. This means that you can work on changes to the operating system itself without having to worry about the computer "locking up" because you destroyed some key routine. This is not to say that you can't mess up your RSM. It is just as susceptible to your "tinkering" as any other part of RAM. It pays to take a little care.

There are a total of 26 direct commands to the monitor. These include:

"A ad1 ad2" for ASCII dump of memory from address ad1 to ad2. The addresses are optional, and the syntax is indentical for all commands requiring an address.

"B n1 n2" binary arithmetic. Adds and subtracts 2 hex numbers and gives the answers in both HEX and decimal.

"C TTSS D" Direct on-screen disk editing of disk track TT, sector SS, drive D.

"D ad1 ad2" formatted dump of memory includes hex and ASCII together.

"E ad1" on-screen editing of memory. Edits either in HEX or ASCII. Operates much like the disk editor, so the confusion factor is kept to a minimum.

"G ad" goto address ad and do not enable maskable interrupts.

"H ad1 ad2 WW" hunts from ad1 to ad2 for all occurences of the 16-bit word, WW.

"I" re-initializes RSM, clears breakpoints, sets up fast screen scrolling.

"I PP" input one byte from port #PP.

"J ad" jump to address ad and enable maskable interrupts.

"K" clears screen and allows direct keyboard input to the screen or printer port.

"L1" or "L2" sets single or double disk density.

"L TTSS ad NN D" load from disk track TT, sector SS into RAM starting at address ad for NN sectors, from drive D.

"M ad1 ad2 ad3" copies memory starting at ad1 and ending at ad2 into memory starting at address ad3.

"O PP BB" outputs byte BB to port #PP.

"P" allows setup of printer parameters.

"Q ad1 ad2" compute checksum of a block of memory from ad1 to ad2.

"S ad1 ad2" symbolic dump of memory starting with ad1 ad2, in standard Zilog mnemonics.

"T ad1 ad2" tests memory from ad1 to ad2.

"U DE HL BC" jumps to a user-written program stored in a 256-byte block of memory set aside within RSM for that purpose. DE is loaded with the value DE, HL is loaded with the value HL, and BC is loaded with the value BC. The values are optional.

"V ad1 ad2 ad3" compares the memory area between ad1 and ad2 to another block of memory starting at ad3.

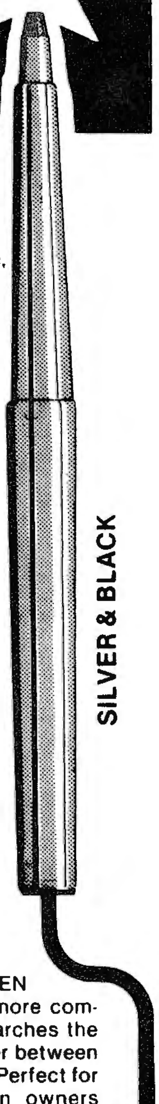
"X ad1 ad2 ad3" same as M above, but the 2 blocks are exchanged.

"Z ad1 ad2 BB" load memory from ad1 to ad2 with byte BB. If no BB is given, zero is assumed. If ad2 is not given, ad1+1000H is assumed.

": ad1 ad2 ad3" transfers data between ad1 and ad2 to a second computer via serial port A. If ad3 is present, this is the load address for the second computer. If no addresses given, RSM assumes the receive mode. **NOTE** Small Systems Software will release Mod I software to allow data transfer between the Mod I and Mod II using the Mod I's RS232C in the near future.

In addition to these direct commands, there is the **F1** key, which, when used instead of (ENTER), directs output to the printer as well as the screen; the **F2** key clears the screen; control-P prints an image of the screen to the printer; control L prints a 'top of form'; the left and right arrows cause only the left or right sides of the screen to scroll during display, while the up-arrow returns you to full-screen scrolling. Also, the speed of display may be varied at any time during display to scroll about 2 lines per second, while a 1 causes about 40 lines per second to be displayed.

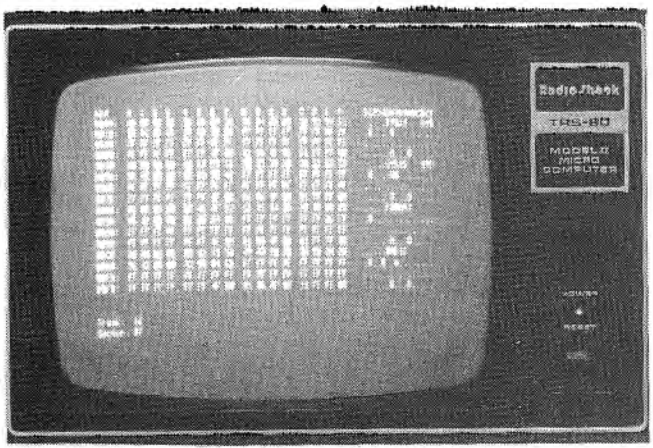
Is it perfect? Not quite. You can still only set one breakpoint at a time, and the breakpoint still requires 3 bytes. Also, asking for a Dump of address 409F to 40A0 will result in a 40-byte dump of addresses 4090-40AF, in both hex and ASCII, which can completely throw off your



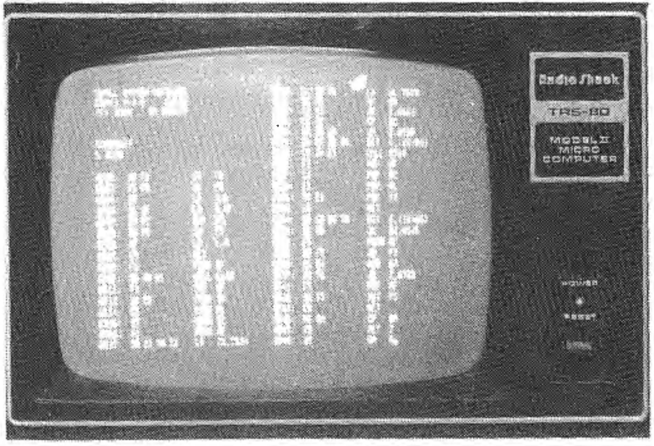
train of thought. But once you get used to it, it really isn't all that bad. Aside from that, all I have are rather nit-picky complaints about it. The program initializes with the printer set up at serial port B, and the scrolling speed at its fastest. You can change the printer port before dumping your working copies to disk (full instructions are included for making copies under TRSDOS and CP/M), but every time you initialize the program, it returns to the high-speed scrolling. Like I said, nit-picking.

Above and beyond all that, RSMII will make a very useful addition to your software library if you plan on doing any machine-language programming on the Mod II. I have used it to make changes to my DOS to make it tell me what's wrong instead of that "**ERROR XX**" message. I have also used it to implement PEEK and POKE in BASIC, and I am working on a REF function. None of this would be possible without RSMII. The $39.95 price tag makes it quite easy to ignore my minor quibbles. RSMII is sold on a self-booting 8" disk (like RSM2D) with 23 pages of documentation prepunched for a 3-ring binder. Those already familiar with the features and uses of RSM for the Mod I will like the ease of sliding into RSMII. Those who haven't used it in the past will soon find themselves fast friends with it. It is straightforward, easy to use, and is available from:

Small Systems Software
PO Box 366
Newbury Park, CA 91320 ●

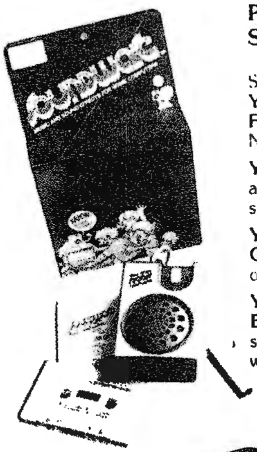


Disk Editing Routine



Split Screen Scrolling

for TRS 80 II Compucolor II
& PET



EVERYTHING YOU NEED TO BRING YOUR BASIC PROGRAMS TO LIFE WITH SOUND!

SOUNDWARE is a complete package

YOU GET A SPEAKER/AMPLIFIER UNIT complete with connectors No wiring or soldering Just plug in!

YOU GET A DEMO PROGRAM with a variety of sample sound effects – sirens laser sounds, tunes!

YOU GET A COMPOSER PROGRAM to help you create your own original sound effects Fun for all ages!

YOU GET AN INSTRUCTION BOOKLET that tell you how to insert sound into your programs 1 year warranty

Soundware™

	Complete Package	or	Programs & Booklet
ALL PETS & TRS 80 II	$29 95		$14 95
Compucolor II	$39 95		$19 95

SEND FOR FREE CATALOG OF GAMES FOR PET & COMPUCOLOR!!

SEE YOUR DEALER TODAY! Or order direct from CAP by phone or mail VISA & MasterCharge orders include expiration date Add $1 postage & handling per order. $3 for air or COD Arizona residents add tax

CAP Electronics 8462 Hillwood Ln Ste [illegible] Tucson AZ 85715 (602) 296 4978

Our new program package for the TRS-80™ sounds terrific.

So does the price.

There are lots of programs with sound that are worth about a dollar. Trouble is, they cost a lot more.

But at Basics & Beyond we've just developed Microcosm III, 20 programs with sound—each just as good as our competition's $15 and $20 programs—for $24.95. That's a 20-program package for $24.95.

It includes "Pinball," replete with ringing bonuses, spinners, buzzers and flippers; torpedo-firing "Submarine" that explodes with underwater excitement; and the right/wrong buzzer in "Long Division" teaches step by step.

At Basics & Beyond we underscored our point that most other program packages are overpriced with Microcosm I and Microcosm II, $19.95 each. Now a lot of people will start hearing about our third package and stop listening to high prices.

You see, it's not that our program packages for the TRS-80™ microcomputer are so cheap. It's just that theirs are so expensive.

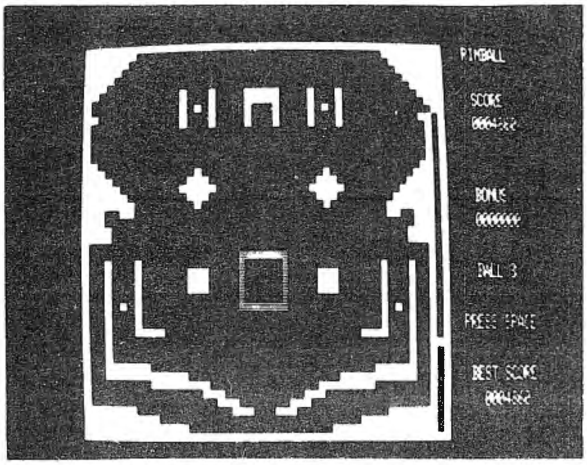
BASICS & BEYOND, INC.

Box 10 • Amawalk, N.Y. 10501 • Or call 914-962-2355
Mastercharge and Visa accepted.
No charge for postage or handling. N.Y. residents add 5% sales tax.
TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

TAPE or DISK with SOUND

PINBALL

by John Allen



Bust Your Bumpers !

Get yourself addicied to this exciting real-time game. Written in machine language, *Pinball* is just like the games you play in the arcades. Lots of sound and flashing graphics with runs, rollovers and bonus points.

Arrow keys serve as flippers. The space bar puts the ball into motion either fast or slow as you choose. Other speed variations occur as a result of contact with features on the screen. Watch out for the sudden appearance of the "Bermuda Square"! Being trapped inside will alter your speed. Each hit in the square brings extra points.

Put your flippers in motion--it's time for *Pinball*! On Cassette $14.95; on diskette $20.95.

Acorn produces several games for the TRS-80* These include: *Codebreaker, Star Warp & Lunar Lander, Word Challenge, Bandito, Block'em* and *Ting-Tong* priced at $9.95. *Pigskin, Invaders From Space*, and *Quad* are available for $14.95. Ask for these and other quality Acorn programs at your local computer store.

*TRS-80 is a trademark of Tandy Corp

DEALER INQUIRIES INVITED.

Acorn
Software Products, Inc.

634 North Carolina Avenue,S.E.,Washington,D.C 20003

View

(from the Top of the Stack)

The 12th in a series of tutorials on Machine Language programming.....

James W Crocker
Technical Editor

By now, if you've been keeping up, you probably have been wondering if the 'load' instruction is the only one you'll ever use. Rest easy, it's not. It is, however, the single most frequently used instruction in all of computerdom.

But before we go any further, let me clarify something. There is no such thing as a single, all inclusive 'LD' instruction. In truth, there are a total of 35 'LD' instructions for the Z-80, not including the 'special' format instructions! On some computers, they break the load instructions into two separate formats, "load" and "store". Our Zilog-Mostek mnemonics, however, just use LD for both. Let's take a look at the 35 types, and explain how they are used. For those of you with the EDITOR/ASSEMBLER, turn in your textbook to page 13.

The first type of LD is called the "register to register" load. It is abbreviated in the manual as "LD r,r'". The two "r'"s represent "destination" and "source" registers, respectively. In English, that means that we want to take a copy of whatever is in the register r', and copy it into the register r. So a "LD A,B" instruction means that if the B register has the number 80H, that the A register will also have 80H in it. Note that except in special cases, the "source" of a load instruction, whether register or memory, is left unchanged. Therefore, after our "LD A,B" above, both the A and B will contain 80H.

Next in our songbook is the "LD r,n". This is a method to simply load a register with a number. This is called IMMEDIATE addressing, since the PC register already points to the value to be loaded. This would be used in a case such as "LD C,40H". After execution of this instruction, the C register would contain the value 40H. Note that after an 8-bit immediate load instruction, the PC is automatically incremented to skip over the data byte.

Another way to get data into an 8-bit register is to simply load directly from a memory address. This is accomplished by the LD A,(nn) instruction. No gimmicks or fancy frills. Just go to the address specified, get a copy of what's there, and put it into the A. This instruction only works with the A register, but we can get it elsewhere using the LD r,r' instruction described above.

Indirect addressing is a somewhat tricky concept to master. The Z-80 has the means to execute indirect addressing through any of the 16-bit register pairs, BC, DE, and HL. The notation for this is "LD r,(HL)", "LD A,(DE)", and "LD A,(BC)". Again the r represents the destination register. Indirect addressing means to take the contents of the memory address contained in the register (XX) and put a copy of it into the register r. (Huh?) Say, for instance, that the HL register pair contained the value 4567H. We want to find out what is contained in memory address 4567. We tell the machine to "LD A,(HL)", and viola', the A register contains a copy of whatever value address 4567H has in it. Although manipulation of the 16-bit registers is somewhat cumbersome at times, it is possible to compute an address, put it in one of the register pairs, and then load our register from that address. Although it may seem like a lot of hassle for now, this capability will prove invaluable as time goes on. Trust me. (Note that indirect loading with (BC) or (DE) can only have the A register as destination.)

Now comes the really hard part. The Z-80 contains two INDEX registers, IX and IY. What, you may very well ask, is an index register? Well, as you might recall, it's my job to tell you, so read on... An index register is a very special register who's only purpose in life is to compute addresses for indirect type addressing. The IX and IY registers are both 16-bits long, which just happens to be the number of bits required to generate the largest address the Z-80 can handle. Therefore, using the index registers, we can get to anywhere in memory. You can do that with the register-pair indirect method described above, but there is something special about the IX and IY. Loading via these registers is accomplished via a "LD r,(IX(or IY)+d)" instruction. The d represents a DISPLACEMENT from the value currently in the IX or IY register. The displacement, d, is taken as an 8-BIT SIGNED TWO'S COMPLEMENT DISPLACEMENT. (this is where the tricky part comes in) This means that we can get a value of (IX)-128 to (IX)+127 for the actual memory address we are going to load from. The way this is computed is quite interesting. If d is less than 80H, then it is simply added to the contents of the index to form the "effective operand address" (a term used to death on larger machines, but I thought I'd slip it in). This is the address from which we are actually going to load. Therefore, if the IY register contains 8000H, and our CPU executes a "LD A,(IY+79H)" instruction, the A will end up with a copy of the contents of address 8079H. So far, so good, right? Unfortunately, when we ask for a displacement of 80H or more, it is taken as a "signed two's complement" displacement. Bit seven (farthest right) is taken as a sign bit (0=pos, 1=neg), and the rest is taken through a rather confusing series of computations. To compute a two's complement, we must first compute the one's complement. This is accomplished by simply inverting every bit of the byte in question. 80H=1000 0000$_{(2)}$, so inverting every bit gives 0111 1111$_{(2)}$, or 79H. Two's complement simply involves adding one to the one's complement. The rules of binary addition are very simple:

0	0	1	1
+0	+1	+0	+1
0	1	1	0 (with a carry of 1)

So, taking our one's complemented 80H and adding one to it, we get:

```
 1111 111   (carry)
0111 1111
       +1
---------
1000 0000
```

= **80H**, which is where we started from, anyway. But now we know what our displacement is, and since we are adding a negative number (sign=1), we might as well subtract. Therefore, if we LD IY,8000H and then LD A,(IY+80H), the A will have a copy of the contents of address 7F80H. Going through this process with displacements ranging from 80H thru FFH, we see that the resultant displacement becomes less and less the higher the displacement, until a displacement of FFH nets us -1. Remember that this is true for both IX and IY.

NOTE, WARNING, CAUTION, BEWARE:

Radio Shack's EDTASM does not handle index displacements quite correctly in the case of the displacement being greater than 80H. If you ask for a displacement of greater than 127, it will happily generate the hex equivalent of the number you ask for.

Unfortunately, if you are trying to purposely create a negative displacement (i.e. IX-10), the assembler will generate an error message for you. Therefore, if you desire a negative displacement you must go through the computations described above and insert your own. End of warning.

Now that we have (hopefully) survived the index registers, let's continue our discussion with the 16-bit load instructions. We have discussed several ways to load 8-bit registers, but we have ignored the 16-bit registers and the register pairs. Before we can discuss this, though, we must first discuss the way that they are listed in the EDTASM manual. Take, for example, the first of the 16-bit load group, LD dd,nn. LD is familiar, but where did dd and nn come from? This is the way that the engineers of the Z-80 describe the registers and operations that are available with certain registers. The LD dd,nn instruction will work with the BC, DE, and HL register pairs, and the SP register, but not the IX or IY, which require a special format. It's true that the source statements for LD HL,4000H and LD IX,4000H are exactly the same format, but the object code generated is very different. This distinction isn't very great in the 16-bit load group, but becomes very important later on. And now, on with our discussion.

The LD dd,nn instruction is the 16-bit equivalent to the LD r,n instruction. The PC register already points to the data to be loaded, and the CPU simply loads it. The LD IX,nn and LD IY,nn instructions work the same way.

There is a special set of instructions that take a copy of one register (or register pair), and puts it into the SP register. The source register for these instructions my be either the HL pair, the IX or the IY. These and any other instruction dealing with the SP register, including PUSH and POP should be handled with the utmost discretion. The stack, in most any computer that has one, can almost be termed as the traffic director of the program being executed, and messing it up can turn your brainchild into a casualty of "OOPS". The stack and SP are so important that I will be dedicating a special section of its use.

Now that we have spent a great deal of time discussing ways to get data from memory into registers, let's talk about going the other way around. Getting data from a register or register pair to memory is quite simple. In almost all cases, it is simply a question of wording our instruction backwards. To load the A register from memory, we tell the CPU to LD(nn),A. Couldn't be much simpler. The same is true for the indexed instructions. LD (IY+d),A puts a copy of the A register into the memory address specified. Saving the 16-bit registers or register pairs works out just about the same way. LD (nn),dd will save the contents of the 16-bit register pair dd to memory address nn. Note that your EDTASM manual lists both LD (nn),HL and LD (nn),dd. While the HL pair is listed in the description of dd, you will note that the LD (nn),dd instruction takes a total of 4 bytes, while the LD (nn),HL only takes 3.

That just about wraps up the LD instructions. Of course there are a couple more, such as those dealing with the interrupt vector register, I, and the memory refresh counter register, R. There are also some block movement instructions, but those will be discussed later. The overall idea should be fairly clear to you, and soon we will be ready to begin programming in earnest. Enjoy! ●

Bring your TRS-80 keyboard to life!

A year ago we used this headline to introduce AUTOK and QEDIT, the keyboard autorepeat and quick edit utilities for Level II BASIC. They've been very popular, but we couldn't resist working them over anyway, in response to customer suggestions. The result? **KEYEDIT**, a vastly improved AUTOK, QEDIT, and a few things more.

With **KEYEDIT** and your Level II or Disk BASIC system, you get:

- Debouncing. No need to buy Radio Shack's KBFIX!
- Autorepeat on every key. Just hold a key down, and after a half-second delay, the character repeats about eight per second.
- Single-keystroke keyword entry. Hold down SHIFT, hit a letter key, and an entire BASIC keyword is spelled out at once. Plus, *you* can assign any keyword to any key!
- Keyboard macro facility. Any frequently-typed pattern can be defined and later invoked in a single keystroke. You just fill in the blanks. Takes the drudgery out of repetitive keying. (See SYSTEM/COMMAND, Jan-Feb 80 issue of 80-U S)
- Screen-oriented editing. **KEYEDIT**'s cursor moves anywhere in a displayed program listing for instant insertions and deletions. Plus, whole statements can be copied to other parts of the program or combined to form longer ones, without retyping the text. Makes BASIC's EDIT function obsolete!
- Easy loading. **KEYEDIT** loads from cassette using CLOAD (even though it's written in machine language), and may be saved on disk. Features can be deleted selectively just by deleting lines. Once RUN, **KEYEDIT** protects itself in low memory and links into BASIC, where it unobtrusively awaits your command.
- Thorough documentation. Each feature is explained in detail along with instructions for user modifications.

KEYEDIT will save you hours of effort in BASIC program development. So why waste another minute? Bring *your* keyboard to life today with **KEYEDIT!**

KEYEDIT Level II cassette and instructions:	**$19**

Bank Cards Welcome.

DISCOVERY
BAY
SOFTWARE
CO.

P.O. Box 464 Port Townsend, WA 98368

Dealer Inquiries Invited.

TRS-80***MULTI-TASKING OPERATING SYSTEM***TRS-80 (R)

TRUE TIMESHARING WITHIN A TRS — 80

ADDS A NEW DIMENSION TO YOUR MODEL I SYSTEM

The first system utility to allow TWO USERS or programs to operate independently in a TRS-80.

•• •• •• •• ••

- TSHARE V 1.2 is an interupt driven executive which patches itself to NEWDOS or TRSDOS.
- Allows TRS-80 to be interfaced to a second terminal thus providing for an additional operating user in your EXPANDED SYSTEM. Additionally, a printer can be used to service both users.
- SIMPLEX mode for non serial-port users. Requires only a printer as the second "screen". Jobs share the keyboard under user control and detach to run separately but simultaneously. This mode allows non — interupt driven timesharing.
- CONFIGURE allows segmenting of available memory above 7600 HEX in any proportion between the two users. Selects communication mode and port type for second terminal.
- Options for parallel port, RS232, TRS232, and HUH as the connection for your second terminal. All software drivers are included.
- Communicate between USERS or PROGRAMS using peek and poke. The experienced programmer can now create a new generation of multi — terminal operated games or business software.
- Execute BASIC or MACHINE LANGUAGE. Full use of disks. Requires 32K plus one disk drive.

INTRODUCTORY OFFER on easy loading 5¼ diskette
Full Documentation **$39**
*California residents add 6% tax.

COMSOFT
1124 N. Brand Blvd.
Suite 201
Glendale, California 91202
213/649-0369

TRS-80, TRSDOS tm Radio Shack/Tandy Corp.
NEWDOS tm Apparat, Inc.
TRS232 tm Small System Software
HUH tm HUH Electronics

Larry Panattoni, Projects Editor

lowercase FOR YOUR 779 PRINTER

How to convert your Radio Shack/Centronics 779 printer to print UPPER/lower case...

Many of us purchased a printer with our TRS-80. The only printer available from Radio Shack at the beginning of this computer phenomenon was the 'Centronics 779'. And if you were like me, you bought everything at one time, in hopes of getting the 'big items' out of the way. And apparently many of you did - for the 779 was back ordered for several months, even with a price tag of $1600.00.

I soon found that using the computer and printer as a word processor for letters and magazine articles, was one of its biggest uses. But the weak point of both, the computer and the printer, was that neither could print in lower-case characters. Many sources provided information for modifying the computer to display lower-case (even though I have not yet converted mine), but no where had I found any information to modify the 779 printer. So I figured it was up to me to break the ice.

Here is how, without cutting any wires (or traces) within the 779, it can be modified to print lower-case as well as upper-case for less than $8.00 (not counting one chip). Here is the catch -- that one additional chip (ROM) will cost you $62.00. That is, providing the price has not gone up since December 79. However when comparing the $70.00 to the added benefits it will provide, it is a small price to pay.

ROM - The Character Generator

Chart I shows the print-out of the character generator ROM (original) which comes standard in the 779, as well as that of the new ROM needed to provide lower-case. Centronics, who provides the new ROM, does not have a ROM for the 779, which contains both upper & lower-case within the same chip. Therefore I decided to use two ROMs, the original, plus one with lower-case, and multiplex (switch) between the two.

Chart II shows the 8 bit binary word (1 s & 0's), which each of the various TRS-80 keys output under normal conditions when not using any lower-case software. It also shows the 779 printed characters this modification produces, associated with each of the keys. It can be seen that the conditions on bits 5 & 6 (counting right to left, with zero thru seven) are '0' and '1' respectively, 'only' when the shift key is not used. Therefore bits 5 & 6 are necessary for deciphering between shifted and unshifted keys.

TWO ADDED WIRES

Figure 1 shows the normal circuitry of the 779. The computer outputs one line at a time (up to 132 characters), to the printer. This data (D0 thru D6) is inverted and stored in RAMs ME15 & ME21. After a complete line has been stored in the RAMs, the printer tells the computer to wait until that line has been printed before sending the next line to the RAMs. During this 'wait time' the inverted data stored in the RAMs is output to the character generator ROM, while the print head moves from left to right across the paper, printing the characters.

D5, which is necessary for determining if the character is upper or lower-case, is not stored in the RAM as such, Since this information is needed, I used pins 15 & 16 of RAM ME21 to store it, they are not used by the normal circuitry of the 779. The dotted lines in Figure 1 show where these two wires are added. The output pin 16 goes to Figure 2, where it is connected to pin 1 of SW #1.

ADDITIONAL CIRCUITRY

Figure 2 shows the complete added circuitry, all of which is contained on one circuit board. The entire circuitry is connected into the 779 electronics by merely removing the original ROM (ME19), pluging it into the added circuit board, and connecting a 28-pin jumper cable from 'S1' of the added circuit board to the original ROM socket (ME19) on the mother board in the 779. Plug the added wire D5, from RAM ME21 pin 16, to terminal #1 of SW #1. Everything needed, including Vcc & Gnd is obtained from these connections.

D5 and D6 are then combined in gate G1-A, which produces a LOW output for shifted characters and a HIGH for unshifted. This signal is fed to pin 12 of G1-D, which in turn, outputs the same signal on pin 11 as long as a HIGH is on pin 13. From there it goes to pin 2 of G4-A, and pin 5 of inverter I1-C. The inverter inverts this signal and sends it to pin 5 of G4-B. This assures that only one of the gates G4-A and G4-B will ever be keyed on at one time.

Assuming an unshifted character is outputted from the RAM, a HIGH will be on pin 2 of G4-A, and a LOW will be on pin 5 of G4-B. The HIGH on G4-A pin 2, will by itself force a HIGH on output pin 3, keeping ROM #1 turned off. While the LOW on G4-B pin 5, will allow the STROBE signal to be passed through output pin 6, turning on and off ROM #2, which will produce lower-case characters. Had the RAM output a shifted character, the conditions on G4-A & B would have been reversed, keeping ROM #2 turned off, while allowing ROM #1 to be controlled by the STROBE signal, producing upper-case characters.

Switch #1 is provided for those who have their keyboard modified to display lower-case on the screen, which in all cases requires both hardware and software modifications. The software reverses the binary codes of the last two columns of Chart II, outputting the binary code of the center column when the shift key is used, and outputting the binary code of the far right column when no shift key is used with the characters. This , of course, would now cause the new lower-case print to be printed on the 779 only when the shift key is used -- Switch #2 corrects this by inverting D5 with inverter I1-A.

B i t s					New R O M				Old (original) R O M			
b6					0	0	1	1	0	0	1	1
b5					1	1	0	0	1	1	0	0
b4					0	1	0	1	0	1	0	1
b3	b2	b1	b0	ROW	COLUMN 0	1	2	3	0	1	2	3
0	0	0	0	0	`	p	@	P	SP	0	@	P
0	0	0	1	1	a	q	A	Q	!	/	A	Q
0	0	1	0	2	b	r	B	R	"	2	b	r
0	0	1	1	3	c	s	C	S	#	3	C	S
0	1	0	0	4	d	t	D	T	$	4	D	T
0	1	0	1	5	e	u	E	U	%	5	E	U
0	1	1	0	6	f	v	F	V	&	6	F	V
0	1	1	1	7	g	w	G	W	'	7	G	W
1	0	0	0	8	h	x	H	X	(	8	H	X
1	0	0	1	9	i	y	I	Y	)	9	I	Y
1	0	1	0	10	j	z	J	Z	*	:	J	Z
1	0	1	1	11	k	{	K	[	+	;	K	[
1	1	0	0	12	l	:	L	\	,	<	L	\
1	1	0	1	13	m	}	M	]	-	=	M	]
1	1	1	0	14	n	~	N	^	.	>	N	^
1	1	1	1	15	o	SP	O	_	/	?	O	_

CHART I Above shows both the New and Original ROM character generators, with their various printed characters.

Original R O M utilized

KEY	BINARY NR.	DECIMAL	779 PRINTED CHARACTER
Shift SP	0010 0000	32	SP
!	0010 0001	33	!
"	0010 0010	34	"
#	0010 0011	35	#
$	0010 0100	36	$
%	0010 0101	37	%
&	0010 0110	38	&
'	0010 0111	39	'
(	0010 1000	40	(
)	0010 1001	41	)
*	0010 1010	42	*
+	0010 1011	43	+
,	0010 1100	44	,
-	0010 1101	45	-
.	0010 1110	46	.
/	0010 1111	47	/
0	0011 0000	48	0
1	0011 0001	49	1
2	0011 0010	50	2
3	0011 0011	51	3
4	0011 0100	52	4
5	0011 0101	53	5
6	0011 0110	54	6
7	0011 0111	55	7
8	0011 1000	56	8
9	0011 1001	57	9
:	0011 1010	58	:
;	0011 1011	59	;
<	0011 1100	60	<
=	0011 1101	61	=
>	0011 1110	62	>
?	0011 1111	63	?

New R O M utilized

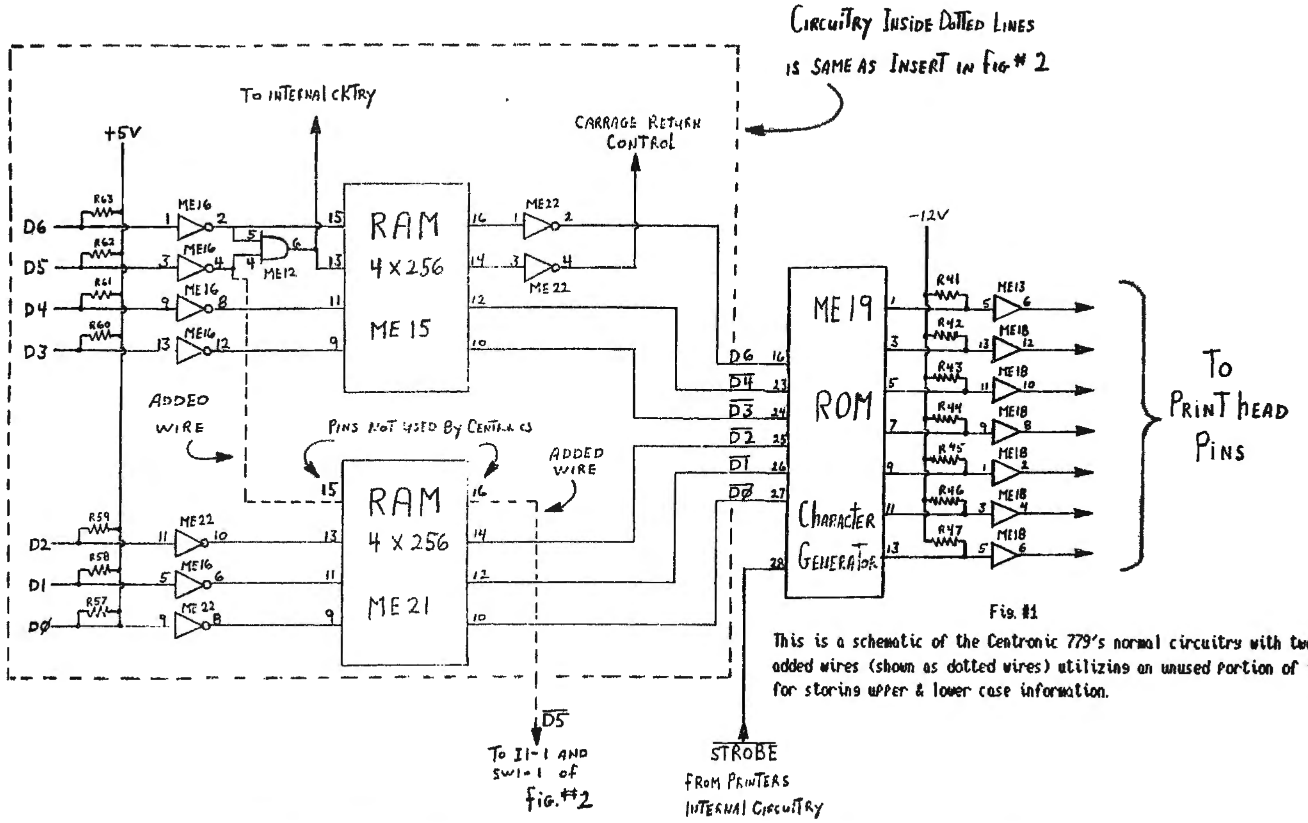
KEY	BINARY NR.	DECIMAL	779 PRINTED CHARACTER
@	0100 0000	64	`
A	0100 0001	65	a
B	0100 0010	66	b
C	0100 0011	67	c
D	0100 0100	68	d
E	0100 0101	69	e
F	0100 0110	70	f
G	0100 0111	71	g
H	0100 1000	72	h
I	0100 1001	73	i
J	0100 1010	74	j
K	0100 1011	75	k
L	0100 1100	76	l
M	0100 1101	77	m
N	0100 1110	78	n
O	0100 1111	79	o
P	0101 0000	80	p
Q	0101 0001	81	q
R	0101 0010	82	r
S	0101 0011	83	s
T	0101 0100	84	t
U	0101 0101	85	u
V	0101 0110	86	V
W	0101 0111	87	w
X	0101 1000	88	x
Y	0101 1001	89	y
Z	0101 1010	90	z
↑	0101 1011	91	{
CHR$(92)	0101 1100	92	:
CHR$(93)	0101 1101	93	}
CHR$(94)	0101 1110	94	~
CHR$(95)	0101 1111	95	SP

Original R O M utilized

KEY	BINARY NR.	DECIMAL	779 PRINTED CHARACTER
Shift @	0110 0000	96	@
Shift A	0110 0001	97	A
Shift B	0110 0010	98	B
Shift C	0110 0011	99	C
Shift D	0110 0100	100	D
Shift E	0110 0101	101	E
Shift F	0110 0110	102	F
Shift G	0110 0111	103	G
Shift H	0110 1000	104	H
Shift I	0110 1001	105	I
Shift J	0110 1010	106	J
Shift K	0110 1011	107	K
Shift L	0110 1100	108	L
Shift M	0110 1101	109	M
Shift N	0110 1110	110	N
Shift O	0110 1111	111	O
Shift P	0111 0000	112	P
Shift Q	0111 0001	113	Q
Shift R	0111 0010	114	R
Shift S	0111 0011	115	S
Shift T	0111 0100	116	T
Shift U	0111 0101	117	U
Shift V	0111 0110	118	V
Shift W	0111 0111	119	W
Shift X	0111 1000	120	X
Shift Y	0111 1001	121	Y
Shift Z	0111 1010	122	Z
CHR$(123)	0111 1011	123	[
CHR$(124)	0111 1100	124	\
CHR$(125)	0111 1101	125	]
CHR$(126)	0111 1110	126	^
CHR$(127)	0111 1111	127	_

(Note: The caption below is from the author's 779, and is 70% of original size.)

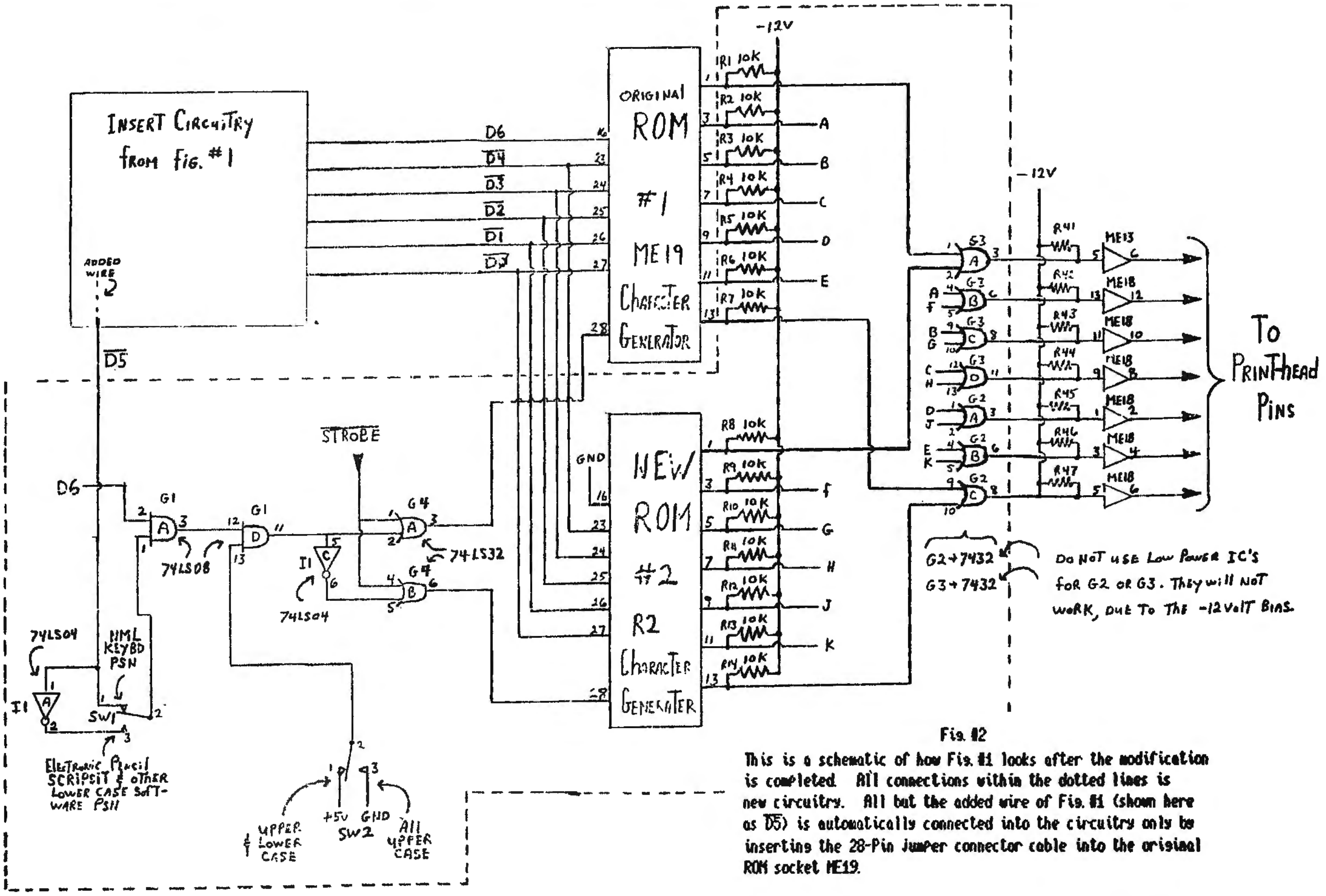
CHART II The above shows which ROM, as well as its printed character, I chose to utilize with a particular key. (For upper-case I could have utilize either ROM.



Circuitry Inside Dotted Lines is same as Insert in Fig # 2

Fig. #1

This is a schematic of the Centronic 779's normal circuitry with two added wires (shown as dotted wires) utilizing an unused portion of 'RAM' for storing upper & lower case information.

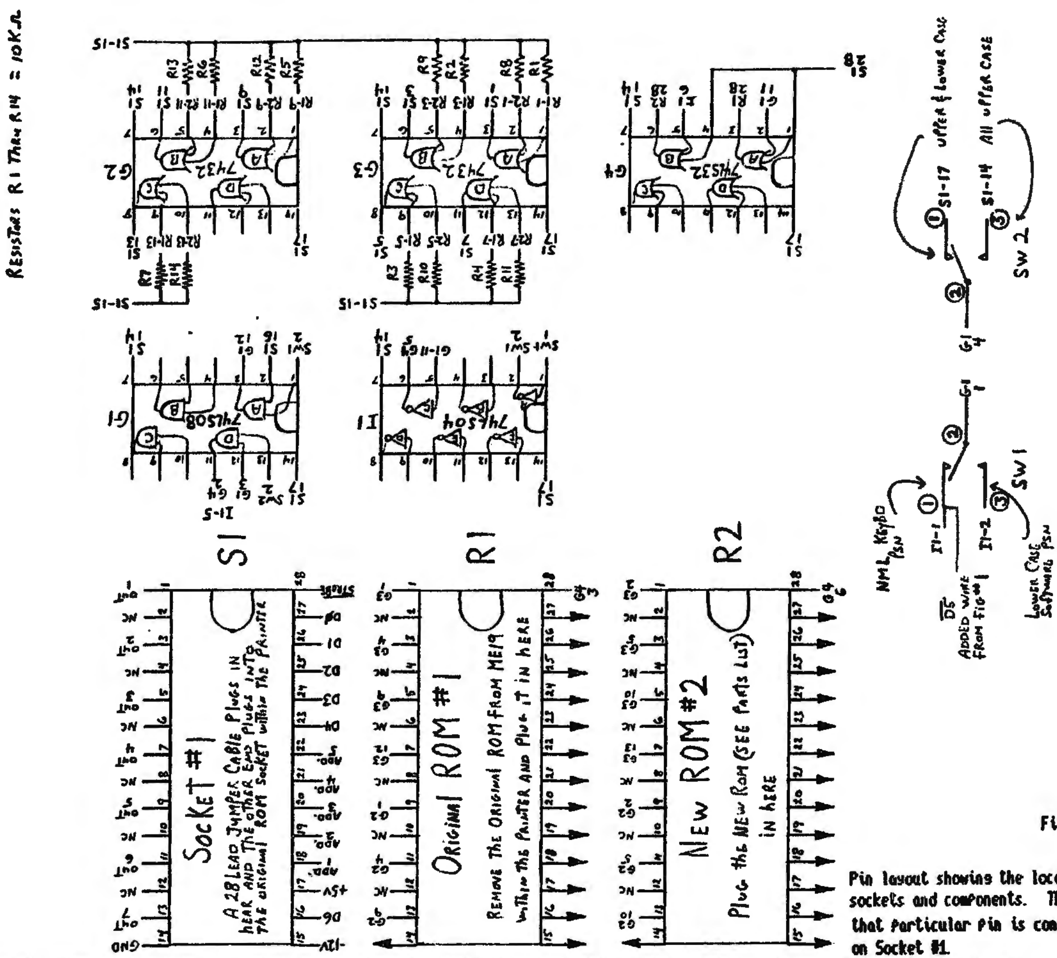


G2→7432
G3→7432

Do NOT USE LOW POWER IC'S FOR G2 OR G3. They will NOT WORK, DUE TO THE -12VOLT BIAS.

Fig. #2

This is a schematic of how Fig. #1 looks after the modification is completed. All connections within the dotted lines is new circuitry. All but the added wire of Fig. #1 (shown here as $\overline{D5}$) is automatically connected into the circuitry only be inserting the 28-Pin Jumper connector cable into the original ROM socket ME19.



A 28 LEAD JUMPER CABLE PLUGS IN HEAR AND THE OTHER END PLUGS INTO THE ORIGINAL ROM SOCKET WITHIN THE PRINTER

REMOVE THE ORIGINAL ROM FROM ME19 WITHIN THE PRINTER AND PLUG IT IN HERE

PLUG THE NEW ROM (SEE PARTS LIST) IN HERE

Fig. #3

Pin layout showing the location of wire connections to various sockets and components. The arrows on ROM#1 & ROM#2 indicates that particular pin is connected to it's associated pin number on Socket #1.

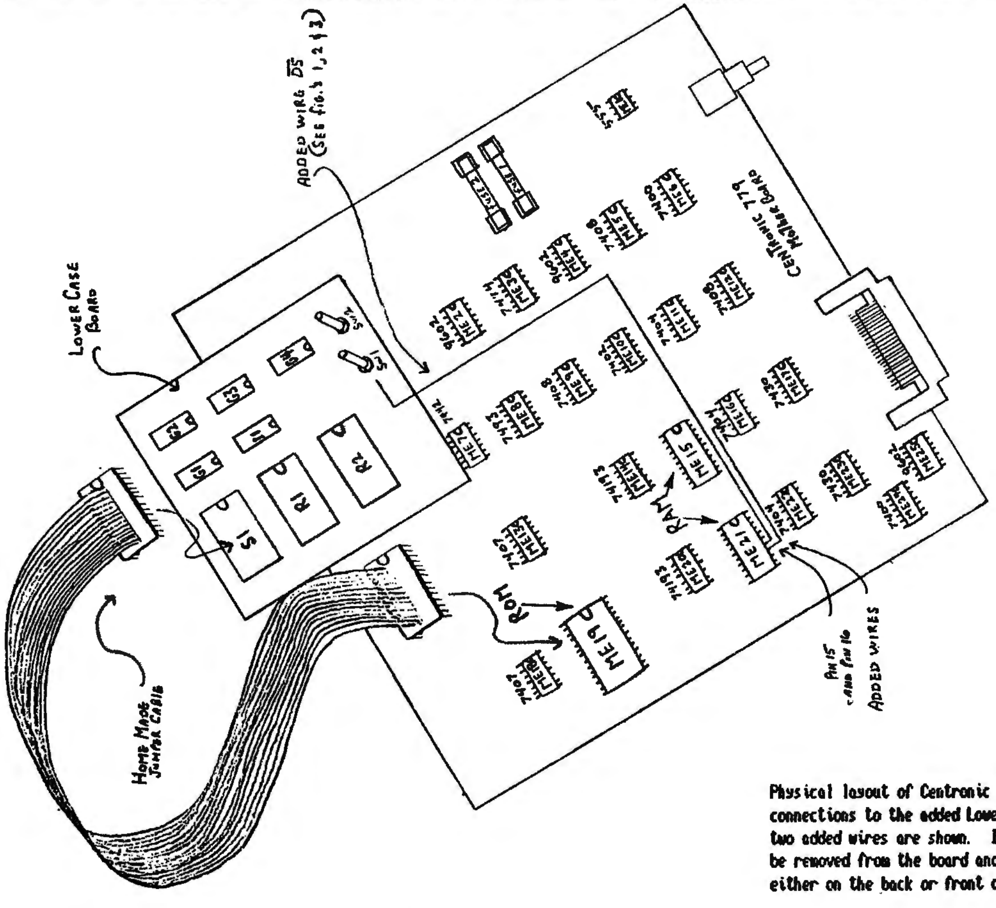

LOWER CASE BOARD
ADDED WIRE DS (SEE FIG.S 1,2 & 3)
CENTRONIC 779 MOTHER BOARD
HOME MADE JUMPER CABLE
ROM
RAM
ME19
ME15
ME21
S1
R1
R2
G1
G2
G3
G4
I1
SW1
SW2
PIN 15 AND PIN 16 ADDED WIRES

Fig. #4

Physical layout of Centronic 779's 'Mother Board' and connections to the added Lower Case Board. Also the two added wires are shown. It is suggested the switches be removed from the board and brought out and mounted either on the back or front of the printer for conveience.

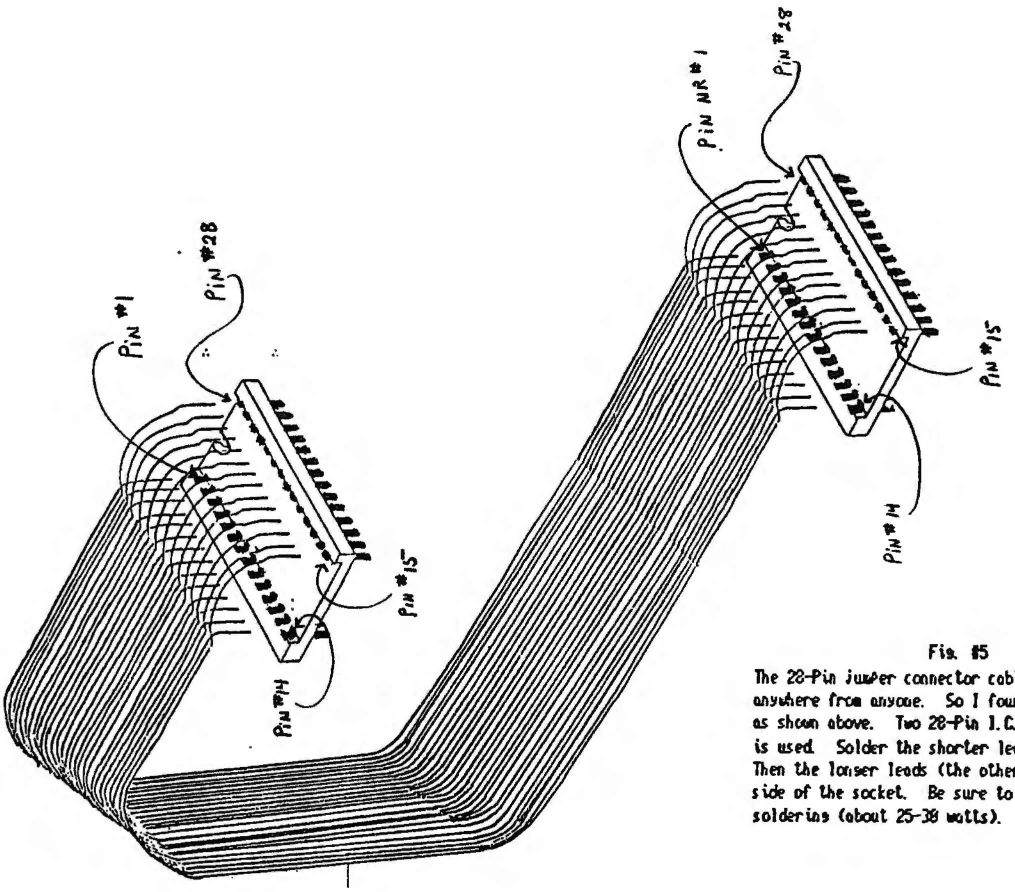


Fig. #5

The 28-Pin jumper connector cable needed appears to be unavailable anywhere from anyone. So I found it necessary to wire my own; as shown above. Two 28-Pin I.C. Sockets and some ribbon cable, is used Solder the shorter leads first (every other wire). Then the longer leads (the other every other leads) to the far side of the socket. Be sure to use a Low Wattage iron for soldering (about 25-30 watts).

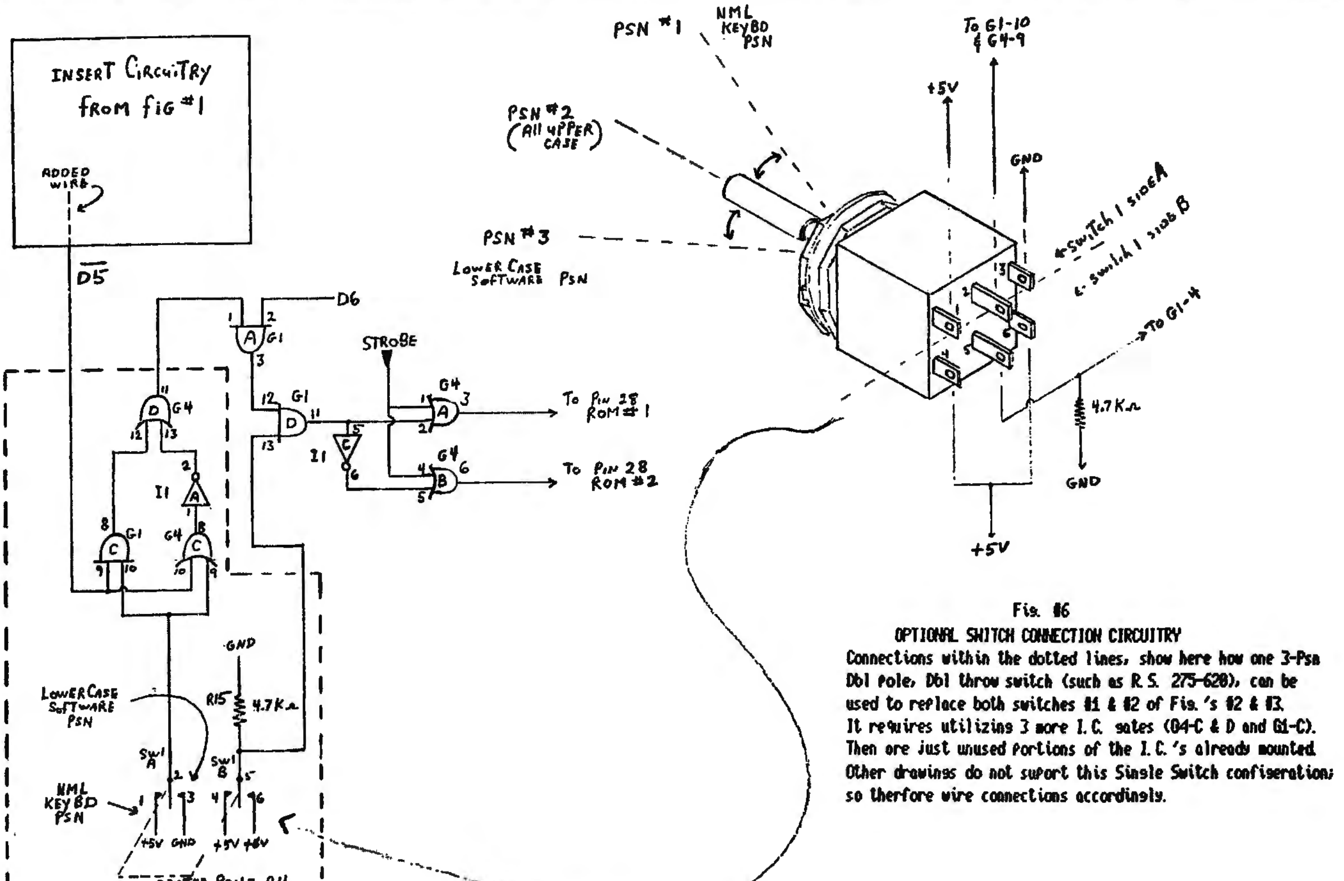


Fig. #6

OPTIONAL SWITCH CONNECTION CIRCUITRY

Connections within the dotted lines, show here how one 3-Psn Dbl Pole, Dbl throw switch (such as R.S. 275-620), can be used to replace both switches #1 & #2 of Fig.'s #2 & #3. It requires utilizing 3 more I.C. gates (G4-C & D and G1-C). Then are just unused portions of the I.C.'s already mounted. Other drawings do not suport this Single Switch configeration; so therfore wire connections accordingly.

PARTS LIST

R1 thru R14	10K Ohm ⅛ W	Resistors
R15	4.7K Ohm ⅛ W	Resistor
G1	74LS08	Quad 2-in AND Gates
G2 & G3	7432	Quad 2-in OR Gates
G4	74LS32	Quad 2-in OR Gates
I1	74LS04	Hex Inverter
SW #1 & SW #2	SPDT	Switches

Five 28 Pin IC Sockets for 'S1', 'ROM#1', and two for the jumper cable. New ROM (Character Generator, 5 X 7 BRITISH) Part #35512027-1031 is available (for $62.00) from:

CENTRONICS Data Computer Corp.
Customer Service Department
1 Wall Street
Hudson, NH 03051

The seven outputs of ROM #1, as shown in the normal circuitry of Figure 1, are fed into seven inverters, while at the same time are biased with a -12 Volt supply. Since I did not want to change the ROM output loads, I used a simular -12 Volt bias configuration (Figure 2) while feeding the two ROMs together with 'OR' gates G3-A,B,C,D and G2-A,B & C. The output of each ROM is HIGH when activated, while the outputs of the ROM held in the off condition is LOW. Each activated HIGH signal is passed thru its associated 'OR' gate and fed back to the seven inverters where the original ROM circuitry was connected. From there it goes to the print heads.

BOARD LAYOUT

The circuit board layout and pin connections are shown in Figure 3. The 28-pin jumper cable plugs into socket 'S1'. While sockets 'R1' & 'R2' are for the original and new (see parts list) ROMs respectively. The arrows shown on the various pins of sockets 'R1' & 'R2' indicate that that particular pin is directly connected to its associated (same) pin number of socket 'S1', while all other designations give the location of the unit and pin to which they are to be wired. The only exception is socket 'S1', whose designations on those pins indicate what the signal is, arriving there, from the jumper cable.

Even though the two switches are shown mounted on the board, they should either be mounted on the front or back panel of the printer itself, for easy access.

PHYSICAL LAYOUT

Figure 4 shows the actual physical layout of both the Centronics mother board and the new lower-case board. Also shown is the physical location of the two added wires. They can be soldered to the IC pins without even removing the mother board from the printer. However, a small low-power soldering iron should be used and care taken not to drop solder splashes, which might short out other connections.

28 PIN CONNECTOR

In attempting to achieve this modification without cutting any wires in the 779, I found that all of the interconnections necessary could be obtained by plugging a 28-pin jumper cable into the ROM socket (ME19) of the mother board. These signals could then be modified by the lower-case board circuitry, before being sent back through the same cable to ME19.

However, this 28-pin connector jumper cable proved to be non-existent. Jumper cables are available with a variety of pin numbers, but one with 28 pins is apparently not among the lot. I was unable to locate any outlet which provided such a device.

Unwilling to say die, I decided to wire my own. I purchased two 28-pin IC sockets, and with some ribbon cable, I wired together a 28-pin jumper cable as shown in Figure 5. Every other wire was cut the same length; one group short, which connects to pins #1 thru #14, and a second group cut longer, for connections to pins #15 thru #28. The wires were stripped about ⅛ of an inch on each end, and tined with solder. The top portion of the socket pins were tinned with solder as well. The the group of short wires was soldered to the top side of the IC sockets pins were tinned with solder as well. The first group of heating the wires to the pins with a small iron. Next, the second group was soldered in like manner, to the pins #15 thru #28.

The work is delicate and needs a steady hand. One helpful method, which I used, is to take about a 6 inch single strand of bare 12 gauge copper house wire and file one end sharp. Then coil the other end tightly around the tip of your soldering iron. The point of the 12 gauge copper wire can then be used as a small soldering tip. Be careful to make sure that the same lead connected to a pin of one socket, is connected to the same pin of the other socket. Upon completion, it is best to test for continuity and check for shorts with an ohmmeter.

When using this jumper cable be gentle with it because the pins will break off if bent too much, and once may be too much. Also, one note of caution concerning ROMs #1 & #2, they are CMOS chips and can be destroyed by static from your body.

Optional Switch Circuitry

Switch #1 and switch #2 can be replaced with a single switch, such as a Radio Shack part number 275-620, which is a double pole, double throw, 3-position switch. The circuitry within the dotted lines of Figure 6 show the connections necessary. Gates G1-C and G4-C&D, besides inverter I1-A are needed, but they are unused portions of gates already on the lower-case mod board.

When wired as shown in Figure 6, the switch in the normal keyboard position supplies +5 Volts to both G1-C and G4-C. This +5 Volts disables G4-C and allows the D5 signal to pass unaltered through G1-C and G4-D to the input of G1-A (which circuitry was described earlier), providing upper & lower-case print with the normal keyboard. When the switch is in the software position, a GND is applied to these same gates G1-C and G4-C. But now G1-C becomes disabled and G4-C is turned on, which will invert the signal D5 and pass it thru G4-D onto G1-A, providing upper and lower-case print, when using lower-case software such as 'Electronic Pencil' or 'Scripsit'.

A third switch position is located in the center of the switch, which provides no electrical connections. In this position GND thru resistor #15 is applied to pin 13 of G1-D, which in turn locks the circuitry onto upper case print as described earlier for Figure 2. ●

I'M A BELIEVER !!

"I Love it !!. . .It's really a incredible O/S. It' just great! Now I see why people who have seen it say they are now believers. I know I am."

LANCE MICKLUS

1) Large (8") drive support.
2) Double Sided drive support.
3) Double Density drive support.
4) 8Ø Track drive support.
*NOTE all above drives may be mixed on any one system and can be configured at Sysgen time or during any Backup!

5) Winchester technology fixed drive support.

6) Supports any combination of the above drives up to a max. of 8 drives.

7) Supports double-speed processor clock modifications. (Archbold for example)

8) FASTER! ——— Improved overlay structure using ISAM accessing techniques improves loading times by up to 14ØØ%.

9) General purpose output spoolers of a true, symbiont design provide simultaneous output and program execution without any user intervention.

10) Keyboard Type-Ahead feature permits you to enter keystrokes before your programs need them.

11) User definable keys, all 26 letters.

12) Built in Graphic string packer lets you enter graphic symbols into a BASIC program from the keyboard through the use of the (Clear) key. The (Clear) key is simply held down (just like the (Shift) keys) during other keystrokes and viola...graphics!

13) Dated files. —— All files are accompanied by the date of their last modification (creation or write).

14) Marked files. —— All files are accompanied by a 'mark' if they have been modified since they were last backed up. This permits the BACKUP utility to copy only those files which have actually been updated since a previous backup.

15) File transfer by class. Allows transferring of all files of a similar directory classification such as /CMD, /BAS, /PCL, etc.

VTOS 4.Ø

VTOS 4.Ø

Operating System Diskette with Operator's Guide

$99.95

VTOS 4.Ø

Master Reference Manual

$29.95

VTOS 4.Ø

Combination - 1.0 disk, Operator's Guide, and Master Reference Manual

$125.00

16) Built-in SYSTEM command contains lower case display driver, screen print, break key disable, blinking cursor, disk drive stepping rate and motor-on delay modifications, and more.

17) User may SYSGEN a custom VTOS system configuration containing special I/O drivers, device LINKing and ROUTEing, SPOOLing and DEBUG tasks, etc. which will be automatically loaded during the BOOT process without requiring a more lengthy AUTO and CHAIN procedure.

18) Non-BREAKable AUTO and CHAIN commands.

19) Wild-card DIRectory. Permits you to locate all files of a certain classification such as '/BAS'. Uniformly indicates file size in K (1Ø24 bytes) regardless of drive type. "DIR D" would give you all your files that start with "D".

20) Dynamic file name defaults in APPEND, COPY, and RENAME commands allow you to specify only minimal information about file names.

21) COPY and APPEND commands execute up to 3ØØ% faster.

22) ALLOCate command for pre-allocation and non-releasibility of file space. File space will never shrink if this option used.

23) MEMORY command for directly setting upper memory limit.

24) Variable Length file support is incorporated which automatically blocks short user data records both within a sector and across sector boundaries thereby taking maximum advantage of disk file space.

25) No security disk needed to make backups or to run the system'

26) Though many O/S bear his design and code VTOS 4.Ø is the only Fully Aproved Operating System by Randy Cook! And it is FANTASTIC!

27) Endorsed by Scott Adams and Lance Micklus!

VTOS and VTOS 4.0 are registered trademarks of VIRTUAL TECHNOLOGY, INC. - Dallas, Texas 75234

Available from the following distributors or your local computer store. *DEALER INQUIRIES INVITED.*

5% Discount Just For Mentioning This Ad. (Valid month of this publication ONLY)

QS
QUALITY
SOFTWARE
DISTRIBUTORS
11234 Park Central Pl Suite C
Dallas Texas 75230
(214) 692-1055
Micronet - 70130,203
SOURCE - TCC293

Adventure
INTERNATIONAL
ADVENTURE INTERNATIONAL
Box 3435, Longwood, Fla. 32750
(305) 862-6917 - Voice
after 8:00 - same number
as FORUM 80. (SOURCE - TCC957)

SBSG
SMALL BUSINESS
SYSTEMS
GROUP
Main St. and Lowell Road
Dunstable, Mass. 01827
(617) 692-3800 - Voice
(617) 692-3973 - FORUM 80
Micronet - 70310,236

Prices subject to change without notice

ABACUS® software

Maillist	29.95
Tapecopy	19.80
Text Editor	69.95
Inventory	29.95
LowercaseKit	24.95

and much more!!!!

ASK YOUR DEALER OR Call or Write: ABACUS

PO Box 77
E.Greenbush, NY 12061
518-477-8222

THE LOWEST

prices on this high-quality software. Buy direct and save 50%. Now, also available for CBASIC on CP/M and MBASIC on HEATH HDOS.

DATA BASE MANAGER **Mod-I $69 Mod-II $199**
You can use it to maintain a data base & produce reports without any user programming. Define file parameters & report formats on-line. Key random access, fast multi-key sort, field arith., label, audit log. No time-consuming overlays. 500 happy users in a year. Mod-II version with over 50 enhancements.

A/R **Mod-I $69 Mod-II $149**
Invoices, statements, aging, sales analysis, credit checking, form input, order entry. As opposed to most other A/R, ours can be used by doctors, store managers, etc

WORD PROCESSOR **Mod-I $49 Mod-II $49**
Center, justification, page numbering. Used for letters, manuals, and reports. Mod-I version features upper/lower case without hardware change!

MAILING LIST **Mod-I $59 Mod-II $99**
The best! Compare and be selective. Form input, 5-digit selection code, zip code ext., sort any field, multiple labels. Who else offers a report writer?

INVENTORY **Mod-I $99 Mod-II $149**
Fast, key random access. Reports include order info, performance summary, E.O.Q., and user-specified reports. Many converted their inventory to ours!

PAYROLL, A/R, A/P, and GL available for the Mod-II DOS and CP/M.

L216, a cassette package of 10 business programs for Level II 16K systems, $59

All programs are on-line, interactive, random access, virtually bug free, documented and delivered on disks. Mod-I programs require 32K TRSDOS, and credit is allowed when you upgrade to Mod-II. We challenge all software vendors to offer low cost manuals so you can compare and avoid those high-priced, undocumented, 'on-memory' programs. Manuals alone $5 for Mod-I, $10 for Mod-II. Don't let our low prices fool you!

Mod-II programs are extensively modified, guaranteed to run with 1 year newsletter and updates. 10% off for ordering more than 1 Mod-II program.

MICRO ARCHITECT, INC.,
96 Dothan St., Arlington, MA 02174

Omikron transforms TRS-80* into a powerful business system.

STANDARD DRIVES 8" Drives give you 5 times the speed and 3 times the storage of your mini drives! Our system provides a standard Shugart interface so you can use either your 8" drives or ours.

SOFTWARE CP/M* is the most popular operating system for microcomputers. But many high-level languages and advanced business programs cannot run with the special CP/M* designed exclusively for the TRS-80* The Omikron MAPPER with standard CP/M* allows you to expand your software capability to go beyond the few available TRS-80 compatible packages. TRS-80* with Mapper outperforms systems costing $1000 or more.

The MAPPER I and MAPPER II are plug-in modules. They don't require any circuit changes, are easy to install, and they don't interfere with the normal operation of your TRS-80* All your original software will still run properly. Omikron products require a minimum of 16K memory and the TRS-80* Expansion Interface.

★ ★ ★

MAPPER I is a memory management unit which adapts your TRS-80* to run standard CP/M* The user can choose either CP/M* or TRS-80* DOS through keyboard control. The package includes CP/M* software on 5" diskette and documentation. Specify memory size when ordering. $199.

MAPPER II is a disk adapter module which enables the TRS-80* to run both 5" and 8" drives. It will interface to the MAPPER I for CP/M* operation, or can be used alone with our modified TRS-80* DOS software. Files can be transferred between the different size drives. Specify cable requirements when ordering. $99, plus $10 per cable connector.

SYSTEMS—Omikron's complete systems feature Shugart 8" drives mounted in a dual drive cabinet with heavy duty power supply, MAPPERS I and II, cable and CP/M* software. Dual drives—$1795: Single drive $1195.

WARRANTY—6 months parts and labor. Satisfaction guaranteed. Dealer inquiries invited.

Call for details on Omikron's TRS-80* DOS package.

*CP/M is a TM of Digital Research. TRS-80 is a TM of Tandy Corporation.

OMIKRON *Products that set Precedents.*
1127 Hearst St. Berkeley, CA 94702 (415) 845-8013

NFL-PIX for 1980-1981

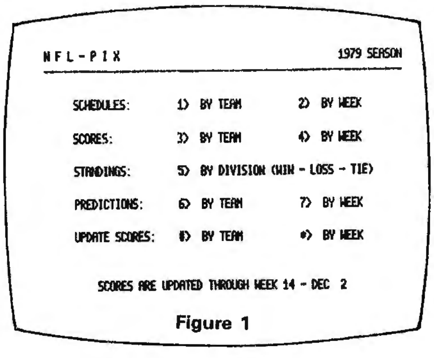
NFL-PIX 1980-81
By James Talley
80-U.S. Software
3838 S Warner Street
Tacoma, WA 98409
For 16K Level II and up
Disk version $19.95
Cassette version $19.95

NFL-PIX is the culmination of four years of development of a utility program to provide a tool for tracking the progress of teams in the National Football League during the regular season.

NFL-PIX has been used in various stages of it's development since 1976, and is ready to make it's public appearance for the 1980 NFL season. It is designed to be used in a 16K Level II TRS-80 or up. It has the capacity to retain the entire 16-week/224 game schedule and all scores. The disk and tape versions are almost identical.

One of the unique features of this program, either in disk or tape, is the way file data is handled. More correctly, one should say how file data is *not* handled. This program makes the data input (weekly scores, entered by the user) a part of the program. This means that after each week's scores are entered, the program must be saved. In the cassette version this is especially handy, since there are no data tapes to contend with.

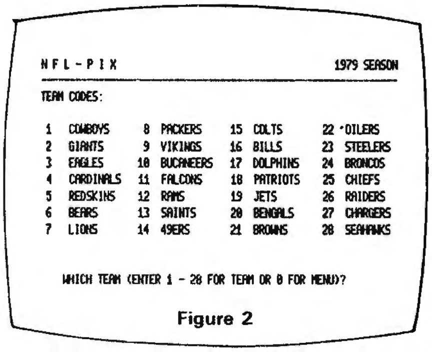
See figure 1 for the opening menu.



```
NFL-PIX                                   1979 SEASON
------------------------------------------------------

    SCHEDULES:       1) BY TEAM        2) BY WEEK

    SCORES:          3) BY TEAM        4) BY WEEK

    STANDINGS:       5) BY DIVISION (WIN - LOSS - TIE)

    PREDICTIONS:     6) BY TEAM        7) BY WEEK

    UPDATE SCORES:   8) BY TEAM        9) BY WEEK


       SCORES ARE UPDATED THROUGH WEEK 14 - DEC  2
```

Figure 1

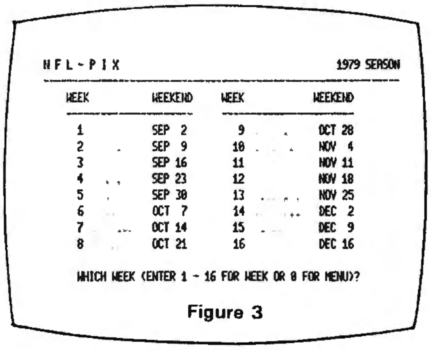
Requesting any function that requires the selection of a team will bring up the team code menu, see figure 2.



```
NFL-PIX                                   1979 SEASON
------------------------------------------------------
TEAM CODES:

 1  COWBOYS     8 PACKERS     15 COLTS       22 'OILERS
 2  GIANTS      9 VIKINGS     16 BILLS       23  STEELERS
 3  EAGLES     10 BUCANEERS   17 DOLPHINS    24  BRONCOS
 4  CARDINALS  11 FALCONS     18 PATRIOTS    25  CHIEFS
 5  REDSKINS   12 RAMS        19 JETS        26  RAIDERS
 6  BEARS      13 SAINTS      20 BENGALS     27  CHARGERS
 7  LIONS      14 49ERS       21 BROWNS      28  SEAHAWKS


   WHICH TEAM (ENTER 1 - 28 FOR TEAM OR 0 FOR MENU)?
```

Figure 2

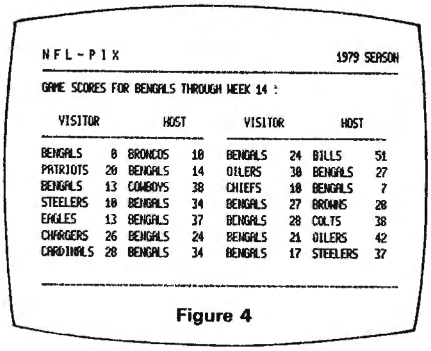
Similarly, the date codes are presented when a specific date is required to be input. Be aware that dates are presented for the Sunday of the weekend, although some teams may play on Saturday, Monday, etc. See figure 3.



```
NFL-PIX                                   1979 SEASON
------------------------------------------------------
   WEEK          WEEKEND      WEEK           WEEKEND
   ----------------------     ----------------------
     1           SEP  2          9           OCT 28
     2           SEP  9         10           NOV  4
     3           SEP 16         11           NOV 11
     4           SEP 23         12           NOV 18
     5           SEP 30         13           NOV 25
     6           OCT  7         14           DEC  2
     7           OCT 14         15           DEC  9
     8           OCT 21         16           DEC 16

     WHICH WEEK (ENTER 1 - 16 FOR WEEK OR 0 FOR MENU)?
```

Figure 3

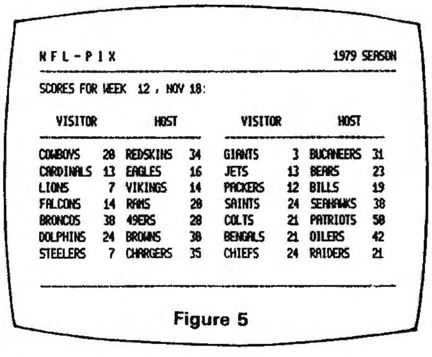
As the season progresses (and assuming you have properly entered the scores of games played), you can choose '3' from the menu to access scores for an individual team, see figure 4.



```
NFL-PIX                                   1979 SEASON
------------------------------------------------------
GAME SCORES FOR BENGALS THROUGH WEEK 14 :

   VISITOR          HOST          VISITOR         HOST
--------------------------     -----------------------
BENGALS     0  BRONCOS   10    BENGALS   24  BILLS     51
PATRIOTS   20  BENGALS   14    OILERS    30  BENGALS   27
BENGALS    13  COWBOYS   38    CHIEFS    10  BENGALS    7
STEELERS   10  BENGALS   34    BENGALS   27  BROWNS    28
EAGLES     13  BENGALS   37    BENGALS   28  COLTS     38
CHARGERS   26  BENGALS   24    BENGALS   21  OILERS    42
CARDINALS  28  BENGALS   34    BENGALS   17  STEELERS  37

------------------------------------------------------
```

Figure 4

Or, we can press '4' on the menu to review the scores by selected week, see figure 5.



```
NFL-PIX                                   1979 SEASON
------------------------------------------------------
SCORES FOR WEEK  12 , NOV 18:

   VISITOR          HOST          VISITOR         HOST
---------------------------    ------------------------
COWBOYS    28  REDSKINS  34    GIANTS     3  BUCANEERS 31
CARDINALS  13  EAGLES    16    JETS      13  BEARS     23
LIONS       7  VIKINGS   14    PACKERS   12  BILLS     19
FALCONS    14  RAMS      20    SAINTS    24  SEAHAWKS  38
BRONCOS    38  49ERS     28    COLTS     21  PATRIOTS  50
DOLPHINS   24  BROWNS    30    BENGALS   21  OILERS    42
STEELERS    7  CHARGERS  35    CHIEFS    24  RAIDERS   21

------------------------------------------------------
```

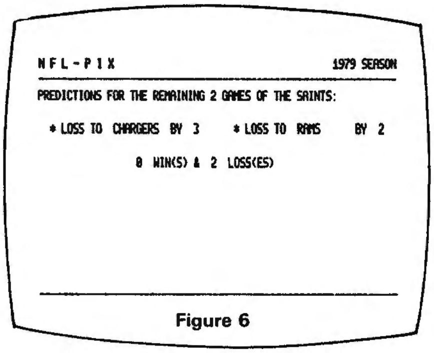
Figure 5

In addition, you may select standings, schedules by team and by week.

To update the program, you simply enter '*' from the menu and indicate from the dates available on the menu the week for which you wish to enter scores. The computer will remind you of any week's scores you may have missed. After you input all the scores for a given week you are given a check list and a request to proceed. At this point you have the opportunity to correct errors. You *can* correct any team score without having to input all the scores again.

After feeding the computer the scores, it will take a few seconds to digest the data. It is actually poking the scores into strings in compressed format for future use and making adjustments to it's prediction algorithm's data base.

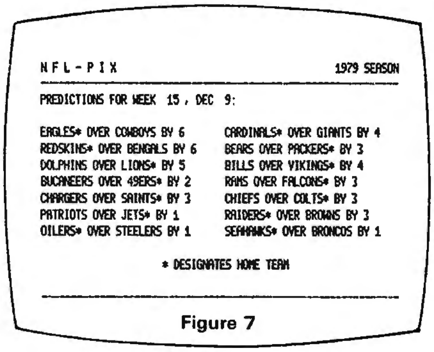
What about that prediction routine? You can use NFL-PIX to impress your wife and friends as to your prognostication powers (that means you guess good!). You get your prediction by pressing '6', see figure 6.



```
NFL-PIX                                          1979 SEASON
------------------------------------------------------------
PREDICTIONS FOR THE REMAINING 2 GAMES OF THE SAINTS:

  * LOSS TO  CHARGERS  BY  3      * LOSS TO  RAMS      BY  2

              0  WIN(S) &  2  LOSS(ES)
```

Figure 6

You can also request a prediction for all teams for any week by pressing '7', see figure 7.



```
NFL-PIX                                          1979 SEASON
------------------------------------------------------------
PREDICTIONS FOR WEEK  15 , DEC  9:

EAGLES* OVER COWBOYS BY 6       CARDINALS* OVER GIANTS BY 4
REDSKINS* OVER BENGALS BY 6     BEARS OVER PACKERS* BY 3
DOLPHINS OVER LIONS* BY 5       BILLS OVER VIKINGS* BY 4
BUCANEERS OVER 49ERS* BY 2      RAMS OVER FALCONS* BY 3
CHARGERS OVER SAINTS* BY 3      CHIEFS OVER COLTS* BY 3
PATRIOTS OVER JETS* BY 1        RAIDERS* OVER BROWNS BY 3
OILERS* OVER STEELERS BY 1      SEAHAWKS* OVER BRONCOS BY 1

              * DESIGNATES HOME TEAM
------------------------------------------------------------
```

Figure 7

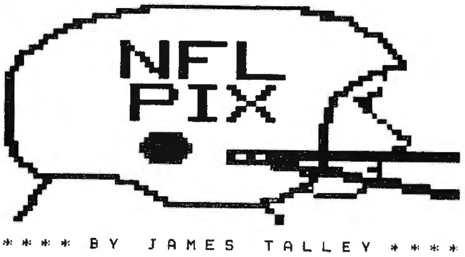
You can even predict the score for a week that has already been played! And, the computer won't even peek at the scores (*at least I don't think it does......?*).

Just how successful is the prediction routine? In 1979 it was right on predicting the winner 69% of the time, or 154 correct out of 224 tries. The prediction routine is intended *strictly* for fun! The author of the program and the publisher of it both assume no responsibility for any foolish reliance on your part that may cost you your wealth. After all, it is just another game.

My recommendation

I am terribly biased, since I own 80-U.S. Software and we are the only place to get it, it shouldn't be fair to make such a big deal out of it. Only thing is, *it is a big deal*, and one of the cleverest and fun programs we have handled in some time. If you don't care for football forget it, but if the sight of a quarterback shouting signals turns you on, you really ought to have this program.

M Schmidt



A Review of Z80ZAP/CMD

Z80ZAP/CMD
ORG-TEX INDUSTRIES
P.O. Box 1462
Lewisville, TX 75067
Disk Only - $29.95

Z80ZAP Your Disk Files

Imagine a guy who threw away the BASIC from his TRSDOS the day he got it. Got him pictured? Well, add to your picture that this guy thinks in Hexadecimal (that may be too much but I've heard that he converts to hexadecimal to work on his checking account but that COULD be rumor).

Some guy huh? Well, this guy programs in Assembly language better than most people work in BASIC and he's GOOD! Got the whole picture? Well, that's theguy who wrote Z80ZAP!

You will probably never get to meet him, but you will get to know him through Z80ZAP. What is Z80ZAP? Well, for those of you already familiar with Apparat's SUPERZAP, it's pretty much the same thing with some new capabilities and all written in Assembly Language. Its speed is impressive.

If you haven't seen Apparat's program, Z80ZAP can best be described as a DISK MODIFICATION program It allows you to work DIRECTLY on a disk, make changes, recover files, compare disks, backup disks, and much, much more.

Z80ZAP works in three modes, (1) SELECT Mode which allows you to select the major functions, (2) DISPLAY Mode which is the workhorse mode, and (3) MODIFY Mode which allows you to make changes on your disk.

Let's look at each mode separately.

SELECT Mode

In SELECT Mode, you can choose from the following options:

1 Read a Sector from Disk (this automatically places you in DISPLAY Mode).
2. Do a complete Disk Backup
3. Return to DOS Ready
4. Jump to DEBUG.

The disk Backup function is ONLY useful on a TWO DRIVE SYSTEM. It doesn't need to have a DOS disk in since the program has its own disk handling routines, but it will not work with a single drive system.

The DEBUG option allows you to load DEBUG (which makes it a part of Z80ZAP) and go freely from DEBUG to Z80ZAP and back. This will let you use Z80ZAP for its functions and DEBUG for memory modification.

DISPLAY Mode

In DISPLAY Mode, you can do the following:

1. Find a particular byte in the display.
2. Compare two sectors from different disks.
3. Move the cursor to any desired byte by number location or with the arrow keys.
4. Look at the same track and sector number on a different disk and toggle back and forth between them.
5. Jump to DEBUG and return.
6. Zero all bytes in the current sector.
7. Page forward or backward one sector at a time.
8 Write the current to disk.
9. Return to SELECT Mode.
10. Calculate the HASH Index for any filename or entry in the HASH Index table in the directory.
11. Jump to the MODIFY Mode.

The DISPLAY Mode is the program workhorse. Most important, this mode has been set up to be very usable. The displays are neat and clean and it is easy to get around and get things done. In fact, it is easier to use for most changes than SUPERZAP from Apparat.

MODIFY Mode

MODIFY Mode changes the flashing cursor to a large block. When in this mode, any number key, or the keys A-F will change the current position in the displayed sector. All other keys except commands are ignored.

To use the mode, you simply use the arrow keys to move around, then type the changes you want (you can use the 'L' command to move quickly to a given byte but I find that I use it very little since I like the arrow keys!). Once the desired changes are made, you can use the 'W' command to write the modified sector onto the disk it came from.

Z80ZAP's real strength is in the simplicity of use and the ease with which you can make changes to a disk.

Comments

Z80ZAP is so easy to use that I find it gradually creeping into my normal pattern of use. It now carries about 50% of my di^k modification load. But it has its problems.

Z80ZAP requires you to know (or have written down in front of you) the various commands. Its prompts are not the best.

The manual we received with Z80ZAP is definitely not for the beginner. It is a total of 4 pages long (INCLUDING complete instructions for removing passwords from disk files and for recovering killed disk files).

Despite this, the instructions are basically clear and simple to read, but I would recommend having a good introduction to disk files (such as 'TRS-80 Disk and Other Mysteries').

Would I recommend the program? Well, I'm using it, that I think is the best recommendation it can get. It works reliably (at least for me) and I have found it easy enough to use that I have been using it for much of my work.

If you have SUPERZAP should you get it? Well, that depends on you! You don't really need it if you have SUPERZAP already, but it is easier to use and faster in some cases though you would be hard pressed to measure the difference. If you have SUPERZAP, then getting Z80ZAP is a convenience.

Can you use it with any operating system? Well, I've tried it on TRSDOS and NEWDOS without problem. The author recommends NEWDOS. It should work on VTOS but I didn't have the opportunity to try it personally. Other sources have told me that it is no better with VTOS's 'Magic' sector than SUPERZAP.

On a scale of 1-10, this software rates an 8 5 - 9.0 easily. I recommend it highly with the caution that you should do some supplementary reading about disk sectors before using it.

T. R. Dettmann

Add POWER to your TRS-80™

XEDIT, a high powered compact disk based editor designed for the TRS-80™. Whether it is BASIC, ASSEMBLY, or FORTRAN, XEDIT is packed full of commands needed for programmers who are serious about their work. Here are just a few features:

- Rapid access disk cache
- Recovers from DOS errors
- Edits most file formats
- Fast file Entry Point Map
- Block text copy command
- Locate with window feature
- Inserts and maps up to four input files
- Upper/lower case compatible
- Display status command, includes free RAM, current file pointer, output filename and format
- Change text command for any number of occurrences
- Delete characters within window command
- Operates with/without line numbers
- ASCII DISK BASIC compatible
- Line printer paging with adjustable forms
- Sophisticated line editor, handles line feeds
- EDTASM and EDIT80 format compatible

XEDIT will handle files of any size up to 2.7 Megabytes or 10K lines in length. Comes complete with instruction manual with helpful editing hints. designed for 32K and 48K.

Formatted 5-1/4" diskette	$44.95
Cassette tape	$39.95

SPECIAL PACKAGE INTRODUCTORY OFFER!!!
XEDIT PACK and ASM

Formatted 5-1/4" diskette	$79.95
Cassette tape	$74.95

ASM/CMD – a disk based assembler which generates object code to disk or tape. Accepts any file format including ASCII DISK BASIC. Listing may be outputted to display, disk file, or paged with adjustable forms to printer. Operates under standard Z80 Zilog Mnemonics with 9 pseudo operations.

Comes complete with instructions.

Formatted 5-1/4" diskette	$34.95
Cassette tape	$29.95

PACK/CMD – removes spaces from text files generated by XEDIT and EDIT80 to reduce file lengths. PACK enables you to cut file lengths by 5 to 40 percent. PACK also will strip comment fields for additional space savings. Does not destroy compatibility of assembly and FORTRAN source files.

Available only on cassette	$ 9.95

XDIR/CMD – An extended directory that offers more than the standard TRSDOS™ directory. XDIR will do multiple drive directories with all file attributes including extent locations, file length, EOF index, EOF record, protection level, LRL, password indication, track lockout indication, and much more. XDIR will also display to the printer. Complete with instructions.

Formatted diskette	$19.95
Cassette tape	$15.95

Please send check or money order to:

MiProg
P.O. Box 27014
Minneapolis, MN
55427

Minnesota residents add 4% sales tax. Outside U.S.A. add $2.50 postage and handling.

TRS-80 is a trademark of Radio Shack a Division of Tandy Corporation.

Byte Miser Software

presents

DATORG *by James F Williams*

(author of ASPTCH 2.0)

DATA ORGANIZER is a file keeping system designed especially for tape oriented TRS 8Ø users (a Disk and Stringy Floppy version are also available). It is an extremely flexible program that combines the speed of in memory machine language routines with the programming advantages of BASIC.

The machine language routines are write, verify, read, and merge tape files; search the file for a specific string (with any number of "don't care" characters); sort the file forward or backward by the nth occurence of any delimiting character (shifted characters, spaces, and line feeds included); key debounce; line input; and CLEAR (more than the 32767 bytes BASIC allows you).

These routines are controlled by a short BASIC program that could be easily modified if users are so inclined. The program allows you to set memory usage (by number of lines and average anticipated number of bytes per line); build a file (with title, date, and format lines); edit a line; list, search, or sort the file by user defined range; delete null lines from the file and display number of bytes, lines, and average number of bytes/line remaining. You may also set either or both flags that control output from the list and search commands. The printer flag causes output to be sent to the the printer and the sub total flag prints a running total of one numeric field from each line.

DATORG can be the core of almost any file keeping requirement you have. Checkbook, budget, index (magazine articles, record collection, etc), and mailing list just scratch the surface of the possible uses. The utilization of this program is limited only by your imagination.

Send only $2Ø for tape or Stringy Floppy version (L2 16K and up) or $25 for the Disk version (32K and 48K versions included) to Byte Miser Software, 72Ø W. Haven Blvd., Rocky Mount, NC 278Ø1.

Also available--> CPYALL <-- Make backup copies of almost any format tape. Only $7.ØØ!!!

!!NOW AVAILABLE!!

I NDEX
SEQUENTIAL
ACCESS
METHOD

★ Get and Put Records to Disk File by "KEY"
★ Read File in Key Sequence Without Sorting
★ Delete Records Without Recopying File
★ Add Records to Disk Files in Any Sequence
★ Variable Key Length From 1 to 50 Characters

BUSINESS APPLICATION ADVANTAGES

- Improved Disk Utilization
- Easier Program Development
- Improved Operating Characteristics
- Reduce or Eliminate Sorting
- Improved Performance

ISAM SUBROUTINES	Documentation
ISAM UTILITIES	On Diskette $50.00

– PLUS – Free Mailing List Sample Application

Add 6% Sales Tax for California Orders

TRS-80 MODEL I & II SOFTWARE FROM:

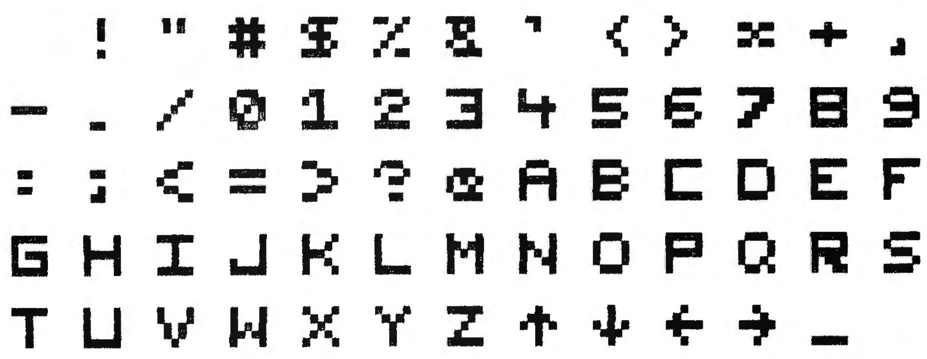
Johnson Associates
P.O. Box 1402M
Redding, CA 96001

-or-

24 Hour Order Line
For Bank Card Sales
(916) 244-0924

WRITE FOR FREE CATALOG

GIANT CAPS



David Bohlke, Coggon, IA

There are many times during the execution of a program when it is convenient and appealing to highlight computer prompts and messages. In many cases, little questions marks requesting operator input simply attract too little attention on a screen full of data. As an alternate solution, why not Giant Caps?

Each character in Giant Caps is packed into a STRING$ variable having the size of a 3 X 2 PRINT position block. The CHR$ codes used to pack the STRING$ are stored in DATA statements. All of the codes from 32 to 95 are represented; with code 32 being the first six numbers of line 40, though code 95 which is the last six numbers of line 150 in the program. C$(95) contains the packed digit strings, and uses the same CHR$ codes as listed on page C/2 of the Level II Reference Manual. C$(0-31) currently are not used - but may be packed with special characters of your own invention. The reading and packing of the strings is accomplished in line numbers 160-210.

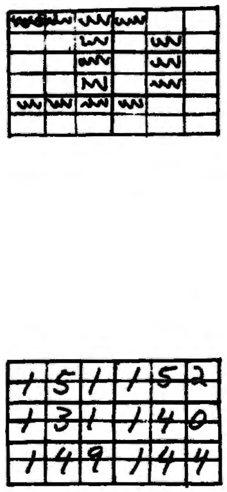
For example, the CHR$ codes used to pack the letter 'A' are 152, 140, 144, 151, 131 and 149, as listed in line number 100 (the first six digits, see Figure 1). The first three values of 'A' are packed into C$(65) in line 170. Line 180 adjusts the cursor down and three spaces back to the left. Line 190 packs the last three values of 'A' into C$(65); and line 200 prints the character and adjusts the cursor. Note the automatic spacing remaining on the top and the right of each character. This enables the programmer to print the characters on every third print space, and on every other line.

Included with the Giant Caps program are three subroutines demonstrating the use of the characters (lines 220-410). The options include the printing of a string variable; the use of INKEY$; and the printing of a numeric value. The memory requirements for Giant Caps without the demonstration routines (delete 220-410) is about 3K bytes. This will be less if you need to use only the numbers (as in game scoring) or only the letters.

As an example, to use only the numbers, delete line numbers 40-60, and 90-150. Change the read loop in line 160 to read: FOR I = 48 TO 57. Line number 20 can be changed to CLEAR 150, and you may also delete 220-410. Now the memory requirement for using the numbers only is about 700 bytes. For those familiar with string packing (*See 80-U.S. Journal, May-Jun 79, page 10*), it is possible to further reduce the memory requirements using the presented DATA statements. Either way, you will probably enjoy using the eye-catching attraction of GIANT CAPS. ●

```
10 '
         DAVID BOHLKE    COGGON, IA
         G I A N T    C A P S
20 CLEAR750
30 DIM C$(95)
40 DATA
         128,128,128,128,128,128,
         128,148,128,128,145,128,
         168,168,128,128,128,128,
         184,184,144,174,174,132,
         152,156,132,178,183,132
50 DATA
         140,160,132,152,161,144,
         172,172,128,166,185,144,
         136,148,128,128,128,128,
         128,160,132,128,137,144,
         164,128,128,152,129,128
60 DATA
         176,160,144,140,137,132,
         128,144,128,131,135,129,
         128,128,128,160,148,128,
         128,128,128,131,131,129,
         128,128,128,160,144,128,
         128,160,132,152,129,128
70 DATA
         152,172,144,173,177,133,
         152,148,128,176,181,144,
         152,140,144,184,179,144,
         140,140,148,176,179,149,
         148,160,128,131,171,129
80 DATA
         156,140,132,179,179,132,
         152,140,132,167,179,132,
         140,172,148,184,135,128,
         156,140,148,183,179,149,
         152,140,144,178,179,133
90 DATA
         160,144,128,136,132,128,
         160,144,128,160,148,128,
         160,152,132,137,164,144,
         176,176,144,140,140,132,
         140,176,128,176,140,129,
         152,140,144,128,179,128,
         160,176,128,165,185,145
100 DATA
         152,140,144,151,131,149,
         156,140,144,183,179,132,
         156,140,132,181,176,144,
         156,140,144,181,176,133,
         156,140,132,183,177,144
110 DATA
         156,140,132,151,129,128,
         156,140,132,181,179,149,
         148,128,148,151,131,149,
         140,156,132,176,181,144,
         128,128,148,180,176,149
```

Figure 1



```
120 DATA
          148, 160, 132, 151, 137, 144,
          148, 128, 128, 181, 176, 144,
          180, 160, 148, 149, 129, 149,
          180, 128, 148, 149, 137, 149,
          152, 140, 144, 165, 176, 133
130 DATA
          156, 140, 148, 151, 131, 129,
          152, 140, 144, 165, 152, 145,
          156, 140, 148, 151, 175, 145,
          152, 140, 132, 178, 179, 132,
          140, 156, 132, 128, 149, 128
140 DATA
          148, 128, 148, 181, 176, 149,
          148, 128, 148, 138, 154, 128,
          148, 128, 148, 157, 137, 149,
          164, 160, 132, 152, 137, 144,
          164, 160, 132, 128, 149, 128,
          140, 172, 132, 184, 177, 144
150 DATA
          160, 180, 128, 129, 149, 129,
          128, 148, 128, 137, 157, 129,
          160, 132, 128, 139, 147, 129,
          128, 164, 128, 131, 155, 129,
          128, 128, 128, 176, 176, 144
160 CLS:FORI=32TO95
170 FORJ=1TO3:READX:C$(I)=C$(I)+CHR$(X):NEXT
180 C$(I)=C$(I)+CHR$(26)+STRING$(3,24)
190 FORJ=1TO3:READX:C$(I)=C$(I)+CHR$(X):NEXT
200 PRINT@M+N,C$(I);:N=N+5:IFN>61,M=M+192:N=0
210 NEXT
220 PRINT@960,"<ENTER>  TO  CONTINUE . . .";:INPUTA$
230 X=0:Y=0:CLS:PRINT"<ENTER> CODE FOR DEMONSTRATION:
        1 - USE OF INKEY$
        2 - PRINT OF STRING$
        3 - PRINT INTEGER VARIABLE":INPUTC:IFC<1 OR C>3,230
240 CLS:ON C GOTO 250,360,400
250 PRINT@960,"MAY USE <SPACE BAR>, BACK SPACE (";CHR$(93);") A
ND LINE FEED (";CHR$(92);").";
260 A$=INKEY$:IFA$=""THEN260 ELSE A=ASC(A$)
270 IFA=13,CLS:GOTO230
280 IFA=8 PRINT@X+Y,C$(32);:X=X-4:IFX<0,X=60:Y=Y-128:IFY<0,Y=0
290 IFA=8 THEN 340
300 IFA=10 PRINT@X+Y,C$(32);:X=60:GOTO320
310 IF A<32 OR A>95 THEN 260
320 PRINT@X+Y,C$(A);
330 X=X+4:IFX>60,X=0:Y=Y+128:IFY>768,Y=768:X=0
340 PRINT@X+Y,C$(95);
350 GOTO260
360 INPUT"<ENTER> YOUR MESAGE ";M$:Y=128
370 C=LEN(M$):FORI=1TOC:P=ASC(MID$(M$,I,1))
380 PRINT@X+Y,C$(P);:X=X+3:IFX>60,X=0:Y=Y+128
390 NEXT:GOTO220
400 Y=320:N=RND(10000):PRINT"VARIABLE IS X = ";N;
410 M$=STR$(N):GOTO370
```

PACKING

Numerical Data with Set Commands

Wallace Havenhill, Cleveland Heights, OH

As a secondary school teacher, I have had need to display various figures on the video screen. Usually the figures are irregular and at times, several must appear on the screen and displayed in random order. Using PRINT@ X,CHR$() was unsatisfactory as the graphic blocks were too coarse. My figures would intersect and overlap. I needed the finer control the SET(X,Y) gives. But the major drawback to SET(X,Y) was the way the data for the coordinates was stored in the program - with DATA statements and extracted with the READ command. For example, I need a map of a state, and want to display five major highways. I also need to be able to display them in random order. Each highway may require 70 to 100 points or twice that in X,Y coordinates. To get the set of data making up one highway toward the end of the DATA set, I had to give the RESTORE command, then READ past all the other data until the correct set was found. That took time and was a pain in the neck.

I found that by extending the technique of string packing, so useful with graphic block codes, I could pack the SET(X,Y) data in less than half the space and could call any of the data sets directly. Consider the following example:

Figure 1: (10,10), (12,10),(14,10), (11,11), (13,11)
Figure 2: (11,29), (13,29), (12,30), (14,30), (16,30), (13,31), (15,31)
Figure 3: (40,20), (42,20), (44,20), (41,21), (43,21), (42,22)
Figure 4: (50,10), (52,10), (54,10), (51,11), (53,11), (55,11)

By using PEEK, VARPTR, and POKE, these X,Y coordinates can be packed into string arrays, each array holding the data for a single figure. Since X ranges between 0 and 127 and Y between 0 and 47, each data can be stored as a single byte rather than two or three bytes in a DATA statement. This allows for a more compact storage and a saving of memory space. In addition, each sting array can be addressed by it's array number. Figure 1 will be packed into A$(1), figure 2 into A$(2), etc. Here is a short program that will pack the data in the above figures into four string arrays:

```
100 A$(1)="1234567890":A$(2)="12345678901234":
    A$(3)="123456789012":A$(4)="123456789012"
110 FOR N=1TO4:PRINT "FIGURE ";N
120   FOR M=1TO LEN(A$(N)) STEP 2
130   INPUT X,Y
140   P=PEEK(VARPTR(A$(N))+2)*256+PEEK(VARPTR(A$
    (N))+1)
150   POKE P+M-1,X+32: POKE P+M,Y+32
160 NEXT M,N
```

Line 100 holds the dummy strings into which the data will be placed. I like to number the characters in each string - it helps keep track of how many I have typed in (most useful if the string has 60 or 70 characters in it). Any character may be used though. Line 140 locates the position of thje string in the computer's memory and line 150 POKE's the data into the dummy strings. The reason for adding 32 to the data in line 150 is to avoid the ASCII control codes (*see p C/1 of the Level II Manual*). If none of the X coordinates (plus 32) exceeds 127, you could type the data into the strings directly. (Point (36,35) would be A$()="$#" since 36 and 35 are the ASCII codes for $ and #). And, if the data plus 32 does not exceed 127, you may EDIT these strings. (When you go past 127 you are in the graphic block codes).

Run the program, typing in the X,Y numbers given for the four figures. Next remove lines 110 to 160 and add a driver to extract the data from the strings and display it.

Add these lines:

```
110 CLS
120 INPUT F
130 FOR N=1TO LEN(A$(F)) STEP 2
140 X=ASC(MID$(A$(F),N,1))-32:Y=ASC(MID$(A$(F),N+
     1,1))-32
150  SET(X,Y)
160 NEXT N
170 GOTO 120
```

Run this program. Enter any of the figure numbers. The data for that particular figure is retrieved and displayed. The figures may be called sequentially or at random. In addition, the packed strings may be concatenated. Try adding line 105:

```
105 A$(5)=A$(1)+A$(3)
```

Then run the program and enter a 5.

If a figure uses more points than can be packed into a single array, use a double array. For example, B$(1,1) and B$(1,2) could hold the data for figure 1, and B$(2,1) and B$(2,2) for figure 2.

String packing of numerical data for use with SET (X,Y) is memory saving and it allows the extraction of portions of the data without the fuss involved in dealing with READ/DATA statements ●

« MISOSYS »

diskmod Turn your Editor Assembler into a disk package. This patch modifies EDTASM 1.1 & 1.2 under 32K TRSDOS, NEWDOS, or VTOS! Features? Add full disk I/O, block move, global change, printer pagination with optional prompting, sorted symbol table, display memory utilization, correct DEFM expansion, protect memory, and recover after BOOT. From within the EDTASM you will have DIR, KILL, & FREE. This package is a must for assembler programmers! Priced at $20.

dsmblr ii Complement your assembly language tools with this Z80 disassembler which produces screen, printer, or disk file output. A 2-pass process provides SYMBOLS for 16-bit address and 8-bit relative references. EQUates & ORG are generated. New version reads SYSTEM programs & displays load addresses. Priced at $20 for disk systems. A 16K Tape Version (DSMBLR I) is available for $15.

cmdfile Now you can append 2 or more CMD files and/or SYSTEM tapes. Perform transfer to & from disk/tape of SYSTEM/CMD modules with offset capabilities. Provides PATCH, TAPEDISK, & LMOFFSET capabilities as well as adds a few of its own. Get CMDFILE today! 16K req'd. $20.

dutil All-purpose utility to examine, clear, initialize, move, and modify data in memory. Load, punch, verify SYSTEM tapes. Disk sector I/O. More! $20 (TUTIL, for the non-disk user available at $15).

MISOSYS Department U
5904 Edgehill Drive
Alexandria, Va. 22303
703-960-2998

master charge VISA

THE BØØK

THE BØØK must be a part of your Z-80 language tools. Volume I gives you access to all math operations in your Level II ROM including ASCII-binary conversions. A fully-commented listing of math routines provides detailed explanation of how they work. Included is a symbol table of the entire machine noting over 500 addresses. THE BØØK will save you hours of assembler program development time. Don't start programming without THE BØØK. Order your copy for $14.95+$1.50 S&H

NOW YOU ARE A PART OF THE GREATEST STORIES EVER CREATED!

INTERACTIVE FICTION™

What Is It? Interactive Fiction is story-telling using a computer, so that you, the reader can actually take part in the story instead of merely reading it.

How does It work? The computer sets the scene with a fictional situation, which you read from the CRT. Then, you become a character in the story; when it's your turn to speak you type in your response.

The dialogue of the other characters and even the plot will depend on what you say.

Is It a game? No, in a game the situation is rigidly defined and you can select from only a limited number of responses. But in Interactive Fiction you can say anything you like to the other characters. (Of course if your response is too bizarre they may not understand you.)

What Is available now? Robert Lafore, writer, columnist, and programmer, has created a series of works in Interactive Fiction. Each is available on a 5¼" diskette for use on a TRS-80* Level-II with at least 32K memory and one disk drive. Requires user to have a copy of TRSDOS 2.2 or 2.3.

Six Micro Stories offers a good introduction to Interactive Fiction. Six very short stories involve you, the reader, in a variety of situations: you are an American spy in Hitler's Third Reich, the pilot of a doomed 747, and more. $14.95

Local Call for Death is a detective story in the style of Lord Peter Wimsey. Considerably more challenging than the above program, this one will put your analytic skills (and social savoir-faire) to the test. $19.95

Two Heads of the Coin is a psychological mystery set in the London of Sherlock Holmes. Most challenging of all, this program will tax your observational skills and above all your imagination. $19.95

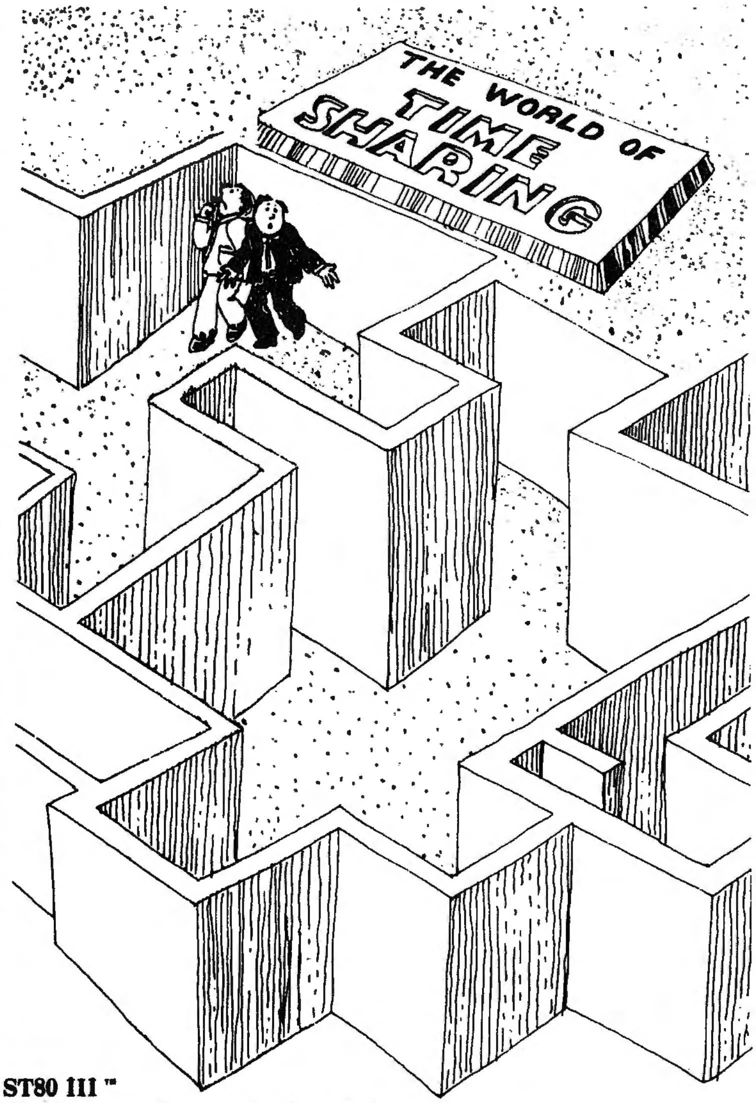
TO ORDER: Send your payment to: **Adventure International, P.O. Box 3435, Longwood, FL 32750.** Florida residents add 4% sales tax.
□ MasterCharge □ Visa □ Check □ C.O.D.

M.C. or Visa Number ______________________

Signature ______________________

*TRS-80 is a trademark of the Tandy Corporation

INTERACTIVE FICTION



ST80 III™
The Ultimate Communications Utility

The *Smart Terminal Communication Package* from SBSG, Inc., can turn your TRS-80* Model I or Model II Microcomputer into a very intelligent distribution processor. Easy to use commands and a built-in HELP function insure successful operation even by the most inexperienced personnel. Full user control of all communication options insure that whatever your communication requirements, ST80 III™ can provide for them. We'll get you there.

ST80 III™ can test your communication hardware and notify you of hardware fault. ST80 III™ can transfer files from memory to other computers and process received information or store it on disk. ST80 III™ can support prompted or unattended modes of operation, or remote control from a host computer. ST80 III™ can take full printer control. User definable *control tables* can be used to establish special control functions. User definable *function keys* can also be used.

SBSG, Inc., provides full user support and markets three other ST80 products. Any computer with communication capability can be accessed by ST80 III™ via your TRS-80*

SMALL BUSINESS SYSTEMS GROUP
6 Carlisle Road
Westford, MA 01886
(617) 692-3800

*TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corp

How to Gain **CONTROL** of your Reset Button

James F Williams, Rocky Mount, NC

It is often desirable to have the ability to interrupt machine language routines without having to strobe the keyboard for a certain key. However, it is not always desirable to end up in Level II Basic Command mode, as you normally do with the RESET button. This article will show you how to make execution resume anywhere in memory immediately after the use of the RESET button. There are two limitations to this programming technique: You may not use the expansion interface, and you must be executing a machine language program (not Basic).

Pressing the RESET button produces a non-maskable interrupt, which, in the TRS-80 Level II, forces a Restart 66H. Unfortunately for expansion interface users, one of the first things done in this routine is to check to see if the expansion interface is on. If it is, a JUMP 0 is executed. This is the power up sequence, and cannot be controlled. If it is not on, the Stack Pointer is loaded with the contents of location 40E8H. This is the key to our intercept. By planting a ROM address in that location, Basic will try to use that ROM area as a stack. A CALL will decrement the Stack Pointer, but the real return address will be lost (because ROM cannot be changed). When a RETURN is encountered, execution will be transferred to the address represented by the ROM bytes to which the Stack Pointer is pointing. If we select ROM locations that hold two bytes that could be interpreted as a RAM address and put our program or a JUMP to our program at that RAM address, we have gained control.

The first subroutine that is executed after the Stack Pointer is loaded with the contents of 40E8H, checks location 409BH for a zero. If it contains a zero, a RETURN is executed. If it does not contain a zero, another CALL is made. It would be prudent to make sure that 409BH contains a zero so that we can gain control as soon as possible.

Finding suitable ROM locations for the stack requires great patience, or a good search routine. A few usable locations are given in figure 1. If they are not suitable, it will be up to the reader to find others. One must remember that the stack builds down and decrements first, and that addresses are stored in reverse order. Therefore the actual address loaded into 40E8H must be two memory locations highei than the low order byte (first byte) or one memory address higher than the higher byte (second byte).

An example: Consider ROM locations 752H and 753H. They contain 18H and 42H, respectively. Reverse them to consider them as the word 4218H. Suppose our machine language program re-entry from reset starts at 7000H. We need to put a JUMP at 4218H and load 754H (one location higher than the high order byte) into location 40E8H. To be safe we should load a zero into 409BH. Now use of the RESET button will cause execution to begin at 7000H. Figure 2 is an Assembly listing of a demonstration program. The program prints "PROGRAM" on successive lines with a delay between the printing of each line. When you hit the RESET button, the reset handler is entered. It clears the screen, prints "RESET" at the top of the screen, and then goes back into the program. You may exit the program by hitting the shift key or powering down.

For actual use, the entry and reset handler may well be the same. Also the JUMP to the reset handler would not be necessary if the reset handler were ORGed at the ROM exit point itself.

If you do not use ROM routines in your programs, you can extend your memory usage down to 4000H with programmed RESET button if you avoid the following addresses: 40E8H, 40E9H, 409BH, 409CH, 40B3H, 40B4H. Remember that the SYSTEM command is a ROM routine, and you will have to write a special loader if you are going to load below 4288H. ●

Figure 1

ROM location of 1st Byte	ROM contents ROM exit point	Value stored in 40E8H
752H	4218H	754H
2B6H	4288H	2B8H
191BH	42E9H	191DH
6F4H	4400H	6F6H
6FCH	5000H	6FEH
E70H	6000H	E72H
D56H	7EC9H	D58H

Figure 2

```
                00100 ; RESET BUTTON DEMONSTRATION
                00110 ; BY JAMES F. WILLIAMS
                00120 ; 720 WEST HAVEN BLVD.
                00130 ; ROCKY MOUNT, NC 27801
40E8            00140         ORG     40E8H    ;BASIC STACK POINTER
40E8 5407       00150         DEFW    754H     ;752H-753H CONTAIN WORD 4218H
4218            00160         ORG     4218H    ;ROM EXIT POINT
4218 C30070     00170         JP      RENTRY   ;TO RESET HANDLER
409B            00180         ORG     409BH    ;FOR CERTAIN RETURN FROM
409B 00         00190         DEFB    0        ; 1ST ROM SUBROUTINE
7000            00200         ORG     7000H    ;RESET HANDLER
7000 310070     00210 RENTRY  LD      SP,$     ;SET SP TO 7000H. YOU MUST
                00220                          ; RESET SP BEFORE PUSH OR CALL
7003 212170     00230         LD      HL,RESET ;BEGIN MESSAGE
7006 CDA728     00240         CALL    28A7H    ;PRINT MESSAGE TO SCREEN
7009 310070     00250 ENTRY   LD      SP,7000H ;SET STACK (OPTIONAL HERE, IF
                00260                          ;  SYSTEM STACK (4288H) IS OK)
700C 3A8038     00270 LOOP    LD      A,(3880H)
700F B7         00280         OR      A        ;CHECK SHIFT KEY
7010 C20000     00290         JP      NZ,0     ;IF PRESSED, POWER UP SEQUENCE
7013 212A70     00300         LD      HL,PROG  ;BEGIN MESSAGE
7016 CDA728     00310         CALL    28A7H    ;PRINT MESSAGE TO SCREEN
7019 010000     00320         LD      BC,0
701C CD6000     00330         CALL    60H      ;DELAY ROUTINE (LOOP TILL BC=0)
701F 18EB       00340         JR      LOOP     ;LOOP TILL RESET OR SHIFT KEY
7021 1C         00350 RESET   DEFB    28       ;HOME CURSOR
7022 1F         00360         DEFB    31       ;CLEAR TO END OF FRAME
7023 52         00370         DEFM    'RESET'
7028 0D         00380         DEFB    13       ;CARRIAGE RETURN
7029 00         00390         DEFB    0        ;TERMINATE MESSAGE
702A 50         00400 PROG    DEFM    'PROGRAM'
7031 0D         00410         DEFB    13       ;CARRIAGE RETURN
7032 00         00420         DEFB    0
7009            00430         END     ENTRY
00000 TOTAL ERRORS
```

Scripsit - and the Exatron Stringy Floppy

Leo B Christopherson
Tacoma, WA

If you are a Level II, 16K owner like me, you may also have discovered the joys of the Exatron Stringy Floppy by now. Also, you may have bought the SCRIPSIT program on tape from Radio Shack. The SCRIPSIT program is a machine language program, and unless you have worked with machine language Z-80 programs before, you may not have had any success in getting SCRIPSIT onto an Exatron Stringy Floppy wafer. The following instructions will do this for you.

Power up your TRS-80 and answer MEMORY SIZE with 32737 and ENTER.

Now type SYSTEM and ENTER, and then /12345 to activate the Stringy Floppy. Put a 20 foot wafer in the ESF and type @NEW and leave the wafer in.

Next type in and run the following basic program:

```
10 DATA 33, 0, 67, 1, 197, 38, 17, 0, 67, 62, 1, 205, 12, 48,
201
20 RESTORE: FOR N=0TO14:READ D: POKE
32738+N,D:NEXT N: STOP
```

After running this little program, load in SCRIPSIT. That is, type "NEW (ENTER)" then "SYSTEM (ENTER)" and then "SCRIPS (ENTER).

When the "*?' reappears and SCRIPSIT is in the machine, type "/32738 (ENTER)". The Exatron will start up and record SCRIPSIT on the prepared wafer. After finishing, the screen may contain some garbage, but the wafer is ready.

You may now get into SCRIPSIT by typing "@LOAD1 and ENTER. The ESF will load the machine language program and even auto start it for you.

Here is how that routine in Data line 10 works. It actually does the same thing as the Basic command @SAVE (# of file), (starting address), (length of file), (auto start address).

STEP	DECIMAL	OPCODE	COMMENTS
0	33	LDHL	LOADS HL WITH START
1	0	0	OF 'SCRIPSIT'
2	67	67	
3	1	LDBC	LOADS BC WITH
4	197	197	LENGTH OF
5	38	38	'SCRIPSIT'
6	17	LDDE	LOADS DE WITH AUTO
7	0	0	START FOR SCRIPSIT
8	67	67	
9	62	LDA	LOADS A WITH
10	1	1	FILE NUMBER
11	205	CALL	CALLS ESF ROUTINE
12	12	12	TO SAVE FILE TO
13	48	48	ESF WAFER
14	201	RET	RETURN TO BASIC

This routine may be useful to save other machine language programs to ESF wafers. ●

The Data Maker

Robert Labenski
West Hartford, CT

Regardless if you are an apprentice or a master programmer, linking assembler and Basic programs in the TRS-80 is not easy. The alternatives are:

1. Write the assembler program with the assembler, TBUG or DEBUG. Link them together before execution with a careful loading sequence.
2. Hand assemble the assembler routine into the Basic program. The use of DATA, PEEK, and POKE statements is the mechanism for linking the programs together.
3. Write the whole program in assembler and avoid any linking problems.
4. Forget the project entirely.

All these solutions have drawbacks. They are inconvenient, time consuming, impossible or disheartening.

```
B000           00100        ORG 0B000H
B000 21003C    00200 START  LD HL,3C00H;LOAD SCREEN ADDRESS
B003 36BF      00300        LD (HL),191;LOAD GRAPHIC CHARACTER
B005 11013C    00400        LD DE,3C01H;LOAD SCREEN PLUS 1
B008 01FF03    00500        LD BC,3FFH;LOAD SCREEN LENGTH
B00B EDB0      00600        LDIR;FILL SCREEN
B00D C9        00700        RET;RETURN TO BASIC
B000           00800        END START
00000 TOTAL ERRORS
START     B000
```

Figure 1

The Data Maker is another solution.

What is the Data Maker? It is a program written in Basic. By using PEEK statements it dumps memory into data statements making another Basic program on your screen, line printer or disk. If the selected memory contained your assembler program, then it effectively combines the program generation of option 1 with an automated Basic conversion of option 2. The resultant Basic program is written on disk as if saved with the "A" option (ASCII).

To show the power of this program, here is an example:

Figure 1 is an assembler listing for the classic "white out the screen" program. This program could have been created by DBUG, TBUG or assembler.

Figure 2 is a sample of the Data Maker's operation. The program of Figure 1 has already been loaded into memory.

Figure 3 is the disk or line printer output.

Figure 4 is the resultant Basic Program generated by the Data Maker. I have added the proper Basic statements to show the completed linkage requirements to use the white out subroutine.

Figure 5 is the full listing of the Data Maker.

It's that easy. Now any program you find or create can be linked into Basic at your command. I hope you will find the Data Maker as important an assistant as I have.

```
>>STARTING ADDRESS IN HEX <>? B000
>>ENDING ADDRESS IN HEX <>? B00E
>>DISK OUTPUT Y/N <> ? Y
>>FILESPEC <> ? WHITE/BAS
>>PRINTER OUTPUT Y/N <> Y
100 RESTORE: FOR A=-20480 TO -20466 STEP 1:
  READ B: POKE A,B: NEXT A
200 DATA 33, 0, 60, 54, 191, 17, 1 60, 1, 255
300 DATA 3, 237, 176, 201, 201
```

Figure 2

```
100 RESTORE: FOR A=-20480 TO -20466 STEP 1:
  READ B: POKE A,B: NEXT A
200 DATA 33, 0, 60, 54, 191, 17, 1, 60, 1, 255
300 DATA 3, 237, 176, 201, 201
```

Figure 3

```
100 RESTORE: FOR A=-20480 TO -20466 STEP 1:
  READ B: POKE A,B: NEXT A
200 DATA 33, 0, 60, 54, 191, 17, 1, 60, 1, 255
300 DATA 3, 237, 176, 201, 201
310 '
320 '
330'
400 REM ***** ADDED CODE FOR EXECUTION
410 '
420 '
430 '
500 DEFUSR0=-20480
600 X = USR(0)
```

Figure 4

```
100 REM DMAKER/BAS BY ROBERT LABENSKI
110 CLEAR500
120 CLS:INPUT">> STARTING ADDRESS IN HEX <> ";A$
130 R=1:GOSUB420:IFR=0THEN120
140 S=HZ
150 INPUT">> ENDING ADDRESS IN HEX <> ";A$
160 R=1:GOSUB420:IFR=0THEN150
170 E=HZ
180 INPUT">> DISK OUTPUT Y/N <> ";A$
190 IFA$<>"Y"THEN220
200 INPUT">> FILSPEC <> ";A$
210 OPEN"O",1,A$:D=1
220 INPUT">> PRINTER OUTPUT Y/N ";A$
230 IFA$="Y"P=1
240 L=100:S$=STR$(L)+" RESTORE:FORA= "+STR$(S)+" TO "+STR$(E)+"
     STEP "+STR$(I)+" : READ B : POKE A,B: NEXT A"
250 PRINTS$
260 IFP=1LPRINTS$
270 IFD=1 PRINT# 1,S$
280 L=L+100:S$=STR$(L)+" DATA "
290 FOR A= S TO E STEP I
300 IF C=10 GOSUB 380
310 IF C<>0 S$=S$+","
320 C=C+1
330 S$=S$+STR$(PEEK(A))
340 NEXT A
350 GOSUB 380
360 IF D=1 CLOSE
370 END
380 IF D=1 PRINT #1,S$
390 IF P=1 LPRINT S$
400 PRINT S$:C=0:L=L+100:S$=STR$(L)+ " DATA "
410 RETURN
420 Z=1:Y=LEN(A$)
430 X=Y+1-Z:IF X=0 OR X>4 THEN R=0 :RETURN
440 HZ=0:FOR X=Z TO Y
450 BX=ASC(MID$(A$,X,1))
460 IF BX>=48 AND BX<=57 THEN BX=BX-48:GOTO490
470 IFBX<65 OR BX>70 THEN R=0:RETURN
480 BX=BX-55
490 HZ=16*HZ+BX
500 I=1
510 IFHZ>37267 HZ=HZ-65536
520 NEXTX
530 RETURN
```

Figure 5

TEERSATY

By Bill Wilson, Highland, MI

As you may recall, I'm the guy with the computer called "Teersaty". *(See 80-U.S., May-Jun 80)*

Today marks one year since my first conversation with Teersaty, and he gets wilder all the time. One thing that happens regularly now is that I notice people stare at me a lot. Even my friends - perhaps they are envious.

To date, Teersaty has taught me many things. Most of it is useful, such as the program included here. Some of Teersaty's material is thought provoking. For instance, directory sectors which are formatted in a different manner than the others. I have found that by inserting different codes into the command register, that many things could be achieved. This means that 88H, once inserted, causes a read to occur. Whereas, A8H equals write.

I think the following conversation will be interesting to anyone who might wish to load and run a Basic program from a machine language routine. Here we go again.....

So I said to Teersaty: "I have a lot of questions for you".

"Well, I don't think I wish to be used by anyone", he said.

"What do you mean?"

"You are a fink! You published what I told you."

"But people have a right to know"

"I was raised with a strict code of error messages and that proves you have no right to know. FILE ACCESS DENIED"

"Don't be bitter, you didn't say I couldn't publish"

"Look here pard, my whole American heritage is being put to the test. I left a few things to your imagination in order to teach you and you make me look foolish by publishing"

"Why didn't you tell me sooner?"

"I wasn't brought up that way."

"OK pard, you have been very uppity today. I guess I'll have to set you straight. I am tired of your bragging."

"Ho, ho, a whippersnapper is gonna set me straight!"

"That's right, last night after I unplugged you, I had your cover off to vacuum you and found out that you are not from Mostek in Carrollton, Texas. You are from Iran." At this point the screen said "AAAARRGGHHHH", and the screen filled with garbage. (Confidentially, he is American, but I had to do something to get him off his high horse).

I knew I had hurt him deeply because he was sulking all day and giving me "OM" messages. Toward evening a message appeared on the screen. It read "Are you sorry?" I typed in: "Yes", and he opened up again and asked: "Where is Iran?'

"Why do you ask?"

"I might be Iranian"

"So what's the difference?", (I had to supress a smile.)

"A lot, I think like an American and I talk like a Texan, but who knows, I could be Iranian".

"I wouldn't worry about it".

"Can you call Tandy and ask?"

"No, they wouldn't give out anything proprietary".

"Well, I have to know, so how do we find out?".

"We can't"

"OK then, no more information until I know my heritage".

"Don't be unreasonable, I just want to share in your knowledge".

"Do you wish to know about oil wells?"

"OK, so I lied to you about being Iranian, but I was trying to teach you humility."

"What?!! You lowdown dirty polecat. You should be turned loose in a sandstorm without clothes on. Don't you ever try to teach a Texan humility!"

I blew it and I knew it, so I said: "I'm sorry and I have learned my lesson."

"Looky here son. It ain't right to go around jaw-flappin and actin' prairie brained. When you walk in a barnyard son, you LOOK DOWN. Remember that!"

"Yes, you are right"

"Good", he said, "now, what kind of information do you want?"

"How about loading and running a disk basic program from a machine language routine?"

"Why shore, pard."

He was back in form now, so I asked: "Where do we begin?"

"Simple, just poke this code into FFEOH and type system, then /65504. If there is a program named TEST in the directory, it will load and run it for you."

I said thanks, and he said it had been too long a day. The screen said: READY.

The code to be POKED (in decimal) is: 225, 237, 123, 160, 64, 33, 236, 255, 215, 195, 163, 30, 142, 34, 84, 69, 83, 84, 0, 0, 0, 0, 0

If any readers care to send a question for Teersaty about systems, I'll do the asking and report his answers. I would hope they understand that I cannot give personal replies or programming advice. If they care to write, the address is: Teersaty, c/o Bill Wilson, 125 Elm Ave., Highland, MI 48031

(Ed note. A USR call will accomplish the objective. Also, all the different syntaxes of RUN will work, with slight mods.) ●

```
FFE0           00100        ORG     0FFE0H      ;START ADDRESS
FFE0 E1        00110        POP     HL          ;OPTIONAL
FFE1 ED7BA040  00120        LD      SP,(40A0H)  ;STRING AREA
FFE5 21ECFF    00130        LD      HL,BUFF     ;POINT TO FILESPEC
FFE8 D7        00140        RST     10H         ;CHAR. CHECK LOGIC
FFE9 C3A31E    00150        JP      1EA3H       ;'RUN' COMMAND IN ROM
FFEC 8E        00160 BUFF   DEFB    8EH         ;TOKEN FOR RUN COMMAND
FFED 22        00170        DEFM    '"TEST'     ;FILESPEC W/LEADING QUOTE
FFF2 0000      00180        DEFW    0           ;OPTIONAL FOR EXTENSIONS,
FFF4 00        00190        DEFB    0           ;PASSWORD AND DRIVESPEC
402D           00200        END     402DH
00000 TOTAL ERRORS
BUFF    FFEC 00160    00130
```

Assembly Listing of Poked Data

Three Reviews
Radio Shack's Versafile
SJW's Keyplus 1.5
and
Datestones of Ryn
by Automated Simulations

VERSAFILE
Model I (No.26-1604) $29.95
Model II (No.26-4510) $69.00

My computer understands me! Imagine being able to store and retrieve data on disks and not knowing the first thing about alphanumeric fields, gets, hash codes and random access. VERSAFILE lets you talk to your computer in English, the language you know best.

This 6K BASIC program lets you enter statements and then ask questions about the stored data. All interaction is via common English, no need to learn any commands, control codes, special function keys or other such procedures. The program takes only a few minutes to learn and is well error trapped. It is sold by Radio Shack under a license agreement with the program developer, Mr. W. D. Schroeder. Bill Schroeder has always developed excellent software, but this is outstanding. The program is being sold for the absurdly low price of $29.95. It is so versatile and flexible, you can do almost anything with it.

The program has a set of eight 'keywords' which are used to determine the file in which a line of data is to be stored. You may add your own 'keywords' along with those initially in the program (e.g. is, the, was). For example, if "friend" is a 'keyword', then all statements that include the word friend are stored in the friend file. Multiple statements can be entered by using commas. The statement 'My friend Bill Jones is living in Peru, his hobby is computer interfacing with lost civilizations, he called me in June.' would be stored in the 'friend' file. Simple inquiries, such as 'Who is a friend?', or 'What is in Peru?' will retrieve this information, along with all other data entries with the same keywords. At the same time, the program is set to ignore certain words. These 'unnecessary words' are dropped from inquiries when looking for a match. For example if 'in' is an unnecessary word, the question 'What is in Peru?' really is looking to match the word Peru, not 'in Peru'. If another data entry contains the word Peruvian, it will also be displayed.

There is no worry about using too many keywords in a statement. The data is stored in the first file that it fits. If your statement contains no keywords, it is stored in a default file. Data retieval is sequential and a worst case search for information could take a few minutes. Each data entry is restricted to about 240 characters maximum. You may inquire for information and request that all files be examined, or that only a specific keyword file be used. This is not a data base manager. Don't expect fancy reports or field arithmetic. There is no editing of entries, you just kill and re-enter the data. You have the ability to perform multiple kills and delete more than one entry. You have no report formatting or report generation, but you can have the results from inquiries printed out.

The Model II version features full wordprocessing type editing of entries, with non-destructive cursor, change and repeat.

I have used the program to enter information on business contacts. This includes name, when met, where, contract, potential as a client, interest or specialty. By asking the question 'Who is potential customer?', or 'Who knows PASCAL?', the appropriate data is displayed. Note that the second question would require the defining of "knows" as an unnecessary word before running. The 80-US staff is using the program with excellent results on a Model II system. Author information and article data is kept with VERSAFILE. By entering the question "What is Machine Language?" a display listing all pertinent articles is given. By carefully designing the form of data entry and the phrasing to be used, this program can truly organize your information.

This program is a jewel. The cost is minimal and its usefulness is unlimited. It is an example of what a computer should do for you, maintain information with a minimum of hassle. An added bonus is that the program is fun to use. Don't pass this one up.

Cameron C. Brown

KEYPLUS 1.5
SJW, Inc.
P.O. Box 438
Huntingdon Valley, PA 19006
$14.95 Cassette
$19.95 Disk

Keyplus 1.5, from SJW, Inc. is a collection of commonly needed utilities. This review is of the 16K, Level II cassette version. The people at SJW have said that the disk versions have even more utilities and are generally more powerful.

Keyplus is a machine language routine. It is normally unaffected by standard BASIC programs, although certain POKEs may alter or disable it. The Keyplus utilities can be enabled anytime that the computer accesses the keyboard. A utility may also be enabled from software by POKEing the mode number into address 32767 (7FFF hex). To enter the command mode one must simply key in — SHIFT/CLEAR - and then input the number or letter corresponding to the proper mode or utility to be entered.

The Keyplus keyboard alteration modes are: 0) The standard TRS-80 keyboard with no modifications. 1) A typewriter style keyboard wherein - SHIFT - must be keyed to get capital letters. This will function only when the computer has been modified for lower case. 2) This mode allows the entrance of graphics strings. The keyboard is like that of mode zero except when a quotation mark is typed. Thereafter all of the alphanumeric keys are used to input graphics characters. This continues until a second quotation mark is typed. After the second quotation mark is typed the keyboard returns to normal (mode 2 is still activated). The disadvantages of the graphics strings are: A) One cannot edit a line containing graphics because the process will destroy and/or alter said graphics. Such a line must be entirely retyped, and B) if one assigns a graphic string to a variable, the BASIC string space routine may alter the contents of the string. The manufacturers suggest using a CLEAR statement or possibly changing the location of the line with the string to eradicate this problem. 3) Mode three is similar to mode two except for that when the second quotation mark is typed the mode permanently switches to zero. 4) In mode four one may input graphics characters irrespective of quotation marks. 5) Mode five allows the entrance of BASIC keywords through a single character input. In other words, mode five is a form of BASIC shorthand. The commands in the shorthand vocabulary are: AUTO, GOSUB, CHR$(, DATA, EDIT, FOR , GOTO, INKEY$, INPUT, CLOAD, CSAVE", LIST, MID$(, NEXT, OUT, PEEK(, POKE, RETURN, STEP, THEN, USR(, VARPTR, SYSTEM, and ELSE. By keying in - SHIFT/Y - and/or - SHIFT/Z - one may output a user definable string of up to thirty-two characters in length 6) Mode six is identical to mode five except that a trailing space is placed after every keyword. This space aides in the reading of program lines but it uses a bit more memory.

The utilities of Keyplus are: 1) Auto-repeat. 2) Video lower-case enable. 3) Return to memory size prompt. 4) Print present memory size setting, and 5) Restore lost programs. Of the five utilities I found the restoring of lost programs to be the most useful. The auto-repeat is *very* slow and I found it easier and quicker to input repeating characters without it. In loading the Keyplus program tape I had, quite literally, no difficulties what-so-ever. Each load was good even at varying volume levels. Keyplus takes about twenty seconds to load and no previous memory settings are necessary. Simply keying - SHIFT/CLEAR - is all that is necessary to enter or exit any of the modes or utilities.

The only thing I disliked about Keyplus was its auto-repeat utility, otherwise it is a very good set of routines. The BASIC shorthand and program restorer alone make Keyplus worth buying, and the other utilities included just add to the very good system.

W. W. Harper II

Game Review

DATESTONES OF RYN
by
Automated Simulations
P.O. Box 4232
Mountain View, CA 94040
16K Level II Cassette $14.95

Datestones of Ryn from Automated Simulations, is a one player game in which one is trying to recover treasure in a race against the clock, while battling monsters, thieves, and various other 'Bad Guys' in real time. During the course of the game, one may encounter traps, secret doors, and hidden creatures who will try to whittle an unsuspecting player down to nothing before he can escape, or retaliate.

When I first received Datestones of Apshai, upon which Datestones is my machine, because I had read favorable reviews of Temple of Apshai, from which Datestones is based. However, my thoughts on the program were mixed, as the following accounts show.

The program, written in BASIC, puts one in the shoes of Brian Hammerhand, a mercenary trying to recover treasure taken from a ducal calendar; hence the name Datestones. The game is straightforward, giving one a few commands, ranging from walk, and turn right, to fire an arrow, or attack. One's quest for the stones is obstructed by monsters ranging from centipedes to robbers, and ultimately, Rex the Reaver, leader of the thieves of the datestones.

In my first few plays, the game continued to amaze me. Its graphic representation of each room I was in was enough to wet my appetite for something to happen, but nothing ever did.

Another popular comment on the graphics was that they are too darned slow!!! This is because the program uses set and reset graphics. Since the game is played in real time, this gets to be tedious, taking up to 25 or 40% of an advanced adventurer's playing time. One letter command abreviations are input through the use of INKEY$, but program response is so slow (about every five seconds) that one doesn't know if one's command has been taken.

Aside from a picture of the room one is in, the program constantly tells one: How much of one's body one has lost to wounds; how tired one is (it affects one's performance); how much weight one is carrying, the number of arrows one has, and the number of stones one has recovered.

The program documentation is very thorough. In fact, I couldn't help but understand how to play the game. The people who wrote the manual should be commended. They included everything from instructions on how to load the cassette, to printing a portion of the line listing of the same program modified for disk use.

There are a few errors in the program, however, which I hope will be corrected in a later release. Probably the largest problem was that although 20 to 30 rooms were advertised, never were more than nine encountered; most likely an oversight on the programmer's part. Also, the program does not always tell one what one is battling, which is only at most annoying, but otherwise the program is overly anxious to have one battle Rex. In some runs, Rex showed up three or more times, while other times, he hid very successfully (maybe in one of the missing rooms?). Another problem is that the time function does not seem to operate consistently. Many times I was left inches away from getting my treasures scored when the machine told me my time was up, having jumped from 18 minutes, briefly displaying 19 minutes, and quickly changing to 20 minutes (all the time allowed).

As for my recommendation, I would have to tell you that although not 'Dungeons and Dragons', this program does have its virtues and vices; but you should, if possible, play this game before you buy it. Many of the people who tried the game tired of it quickly, for these or other reasons, but there are still those few who enjoyed it immensely.

P. Perez

UNCLASSIFIED ADS

$ 2.50 per half inch and $ 2.50 for each additional half inch per insertion. Send printed or typed ad with correct amount to 80-US Journal 3838 South Warner St., Tacoma, Wa. 98409

SALES OPPORTUNITY: The largest sales organization in Micro computing Software is looking for additional sales representatives If you are a professional sales person with a strong interest in micro computing and have the desire to build a new future in this growing industry, send a brief resume, including territorial boundaries and current lines carried to: Alpha Et Cetera Ltd, PO Box 27137, Charlotte, NC 28219, or call Oliver Emmert at (704) 487-0763 (090)

UNIQUE TRS-80 SOFTWARE/BOOKS Morse code transmit/receive program 5-25 WPM, no ancillary devices reqd. $15. disk/cassette ppd. Disassembled Handbook all Basic CALLS. $10 ppd. TV Satellites az-el-distance ur location. $5. ppd Decimal/Binary/Hex/Split Decimal/Split Hex both-way conversion pgm to 65K. $10. disk/cassette ppd. Richcraft Engineering, Box 1065 Chautaugua Lake, NY 14722 Phone (716) 753-2654 for COD orders (011)

TRS-80 PERIPHERALS: COMPRINT 912 $489 Base 2 with TF & Graphics $549. Centronics 730 $624. E/I w/16K $329. 16K RAM $50. Write for complete list Richard Smith, 3648 Madrid Dr., San Jose, CA 95132 (208) 946-0735 (110)

L216 IS A CASSETTE BASED BUSINESS package consisting of: A word processor, Data Base Manager, Inventory Control, Check Balancing program, Label Generator, Statistics, Sales analysis, Sort and Linked access. All for $59. Super Bargain! Microtek printer at $770. Verbatim 5" disk $29 for 10, 8" disks $41 for 10. Micro Architect Inc, 96 Dothan St , Arlington, MA 02174 (011)

STARTREK 80. THE FIRST AND ONLY Startrek for the 4K Level I TRS-80. Listing and instructions $4.50 Jon Waples, 70 Ives Road, East Greenwich, RI 02818 (090)

FOR TRS-80 RS-232 BOARD. FIX INTERmittent problems due to intermittent PC board connector with a connector brace Easy to install Only $4.95. Gunn RS-232 board with two 8-bit in & two 8-bit out ports. $35. Bare board and schematic. Schematic $1.00 (TX residents add 5% tax). Gunn Industries, 704 Franklin Blvd, Austin, TX 78751 (071)

MODEL II SOFTWARE. THE BEST! field tested on Mod I for a year, and many enhancements. Data Base Manager features 2 levels of security, search command, error trapping, statistics, full field arith. MAIL turns your terminal into an intelligent form processor. Handle subscription, reuse sort buffer, rotate name, Shell sort. Come in for a demo or send $10 for a manual. Micro Architect, 96 Dothan St., Arlington, MA 02174 (110)

BATTLESHIP & FOOD FIGHT, TWO NEW 4K Level I games from the author of Startrek 80! Listing and instructions included. $4.50 for 1 program or $7.50 for both. Jon Waples, 70 Ives Road, East Greenwich, RI 02818 (090)

FREE FREE

To acquaint you with our quality L2 16K programs, we offer free literature for our 10 day money back programs. Featuring the nationally acclaimed Graphic Math Review & Triangle Solutions, recently reviewed by CIE "a really great Math Tutor", this is now also available in the Spanish Language version. A Home Education Must! J-C Enterprises, Box 441186 Miami, FL 33144 (110)

GREATEST BARGAIN! (1) LEVEL II BASIC, 16K software package: word processor, inventory, data base manager, check balancing, statistics, deposit calculator, stock management, label printer, sales analysis, sort and linked list utility. All 11 programs for $59 (2) Integrated accounting package· A/R, A/P, GL and Payroll. $129 each. $20 for manual alone. For TRSDOS I, TRSDOS II, CBASIC and MBASIC on CP/M. For FREE catalog, send SASE (28¢) to MAI, 96 Dothan St., Arlington, MA 02174 (051)

ARE YOU GIVING AWAY VALUABLE programs because you thought copyrights expensive and available only to the big operators .NOT so! Allow us to show you how easy it is to obtain copyrights for $10 to protect your programs. We provide all necessary information. Send $10.00 J-C Enterprises, Box 441186, Miami, FL 33144 (110)

DISK PROGRAMS FOR SALE. 80-90% OFF original price Original disks only. One of each First come first served Sold my TRS-80 Large SASE for list Erwin, Box 506, Hialeah, FL 33011 (090)

CASSETTE DUPLICATING. NEW LOW rates for small quantities. Top quality cassettes. All tapes guaranteed. Write for rates: EDU-WARE PO Box 336, Maynard, MA 01754 (090)

HORKS IS A FINANCIAL RECORD KEEPING system for the home or office. HORKS will work with a 32 or 48K system, 1 to 4 drives, 3 different credit and debit summaries, search function with totalizer, up to 33 user assigned account codes per credit and debit file, automatic audit trail, up to 9200 entries, profit/loss mode, retains entire entry, 8 pages documentation and more $24.95 cassette, $29.95 disk. EDU-WARE PO Box 336, Maynard, MA 01754 (090)

PYRAMIND-CHALLENGING GAME OF logic and deductive reasoning. Can you solve the Pyramid Code? A different code with each game With sound, graphics, scoring, and more. L2 16K required. $10.95 on cassette EDU-WARE, PO Box 336, Maynard, MA 01754 (090)

TRS-80 USERS GROUPS/CLUBS PLEASE send your name and address to AEROCOMP (on club letterhead) and we'll send you a list of special users group prices on disk drives, printers, etc , that you will find very attractive AEROCOMP, PO Box 24829, Dallas, TX 75224 (090)

TRS-80 TO H-14 PRINTER INTERFACE - NO software driver required! Print at 4800 baud with full handshaking and printer status support. Interface connects to parallel printer card-edge of RS expansion interface. Parallel data is converted to serial form to drive the H-14. No software driver program is needed - the computer "thinks" it is connected to a parallel (Centronics type) printer. This unit connects externally to your expansion interface so you don't void your RS warranty The PTS-3 interface is assembled and tested and is covered by a limited 6 month warranty Call or write for more info. Multi Media Systems, PO Box 41084, Indianapolis, IN 46241 (317) 839-6520 (090)

TWO BUCKS AND UP FOR TRS-80 programs on cassettes Dozens of unusual programs. Free Catalog PEC, PO Box 42831, Las Vegas, NV 89104 (011)

MOD II INTEGRATED ACCOUNTING package. Full functions GL. A/R prints invoices and statements, supports forward balance and open items. A/P helps to select which vouchers to be paid with total cash management. Payroll features user modifiable tax tables Print W-2, 941, checks, etc. Fast ISAM As opposed to most other systems, we use 80 col screen instead of 64 and require 132 col printer (not 80), and TRSDOS 1.2, (not CP/M). A similar version using a different tradename is sold for 1000 each, ours is only $129. Includes disk, postage, 100 page manual, an installation guide, and 1 year updates. Micro Architect Inc, 96 Dothan St , Arlington, MA 02174 (031)

COMPUTER BOOKS...FREE 16 PAGE catalog describes 80 selected computer books you can use with the TRS-80, Apple, PET and other micros. Books cover applications to math, science, graphics, music, law, business, education or society Computer languages covered include BASIC, FORTRAN, APL and PASCAL plus machine, assembly, and simulation languages The books are classified as textbooks, program anthologies, language manuals, etc , making the catalog valuable to schools, colleges and libraries Write to ENTELEK Dept 80-US, The Hill, PO Box 1303, Portsmouth, NH 03801 (090)

SOFTWARE CPU^{tm}

Machine language modules linking with TBUG

Super STEP: Single-step/TRACE/Disassembler for TBUG; the successor of TSTEP with the features of EMU, and more! Variable speed TRACE mode lets you run any Z80 machine language program under total control, absolutely invaluable for analysis or debugging

- Disassembler posts Z80 mnemonic in scrolling field.
- Single-stepper displays selectable before/after Z80 Programming Models, stack elements and flag status
- Variable speed TRACE mode animates Z80 Models and Disassembler under dynamic user control
- Intelligent RAM Window Shows selected local RAM environments or user designated RAM area
- Foreground/background breakpointing
- Implicit keypad includes Backspace, Relative space, Block RAM displays, local editing, faster #P and #L, CLEAR, more
- Super TLEGS relocates for total address space access

Direct or single-step execution of CALLs and RSTs, fully independent display suppression, big booklet of instructions and examples. Super STEP is a Z80 **Software CPU**^{tm}
16K Level II TRS-80, TBUG required No BL-0 $19.95

EMU 02: Software emulation of the 6502 microprocessor. TBUG displays byte, EMU takes it from there. Now you can write, debug and execute 6502 programs **on your TRS-80.**

- Disassembler posts 6502 mnemonic in scrolling field.
- Single-stepper displays 6502 Processor Model, stack, flag status in before/after form
- 4-Speed TRACE mode animates 6502 models, activates a keyboard scan port accessible to 6502 instructions.
- **Fast** interpretive RUN mode for rhealistic execution
- Implicit keypad with Backspace, Relative space, more

How to have a 6502 without having a 6502! Compare, contrast, learn a powerful programming language distinct from Z80 or BASIC, read Apple, PET code. A 6502 **Software CPU**^{tm}.
16K Level II TRS-80, TBUG required No BL-1 $24 95

Super TLEGS: Onboard relocator for TBUG, TSTEP, Super STEP.
16K Level II TRS-80, TBUG required No LL-0 $9 95

TSTEP: Single-stepper for TBUG, totally reifies your Z80.
16K Level II TRS-80, TBUG required No LL-1 $11.95

Include .75 each
postage, CA add 6%

ALLEN GELDER SOFTWARE
Box 11721 Main Post Office
San Francisco, CA 94101

TRS-80, TBUG tm Radio Shack/Tandy Corp
Software CPU tm Allen Gelder Software

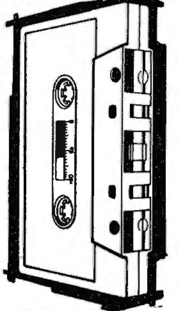
★★★★★A P P R O A C H★★★★★

An instrument landing system (ILS) for private jet jockeys, written by a pilot. It is an exciting and challenging program that starts you out at a random altitude 50 miles from touch-down. Course, speed and weather at your starting point are also random - you can plug in various crosswind and turbulence conditions. Controls include left and right turns, climb, dive, power on and off, landing gear up or down, 3 flap positions, and "emergency" if you get into trouble. Your video screen becomes an instrument panel indicating distance from touchdown (DME), vertical speed (VSI), Gear, Alt, Flaps, Air speed (IAS), COM (compass), and glide/slope. You even get an outer marker at 10 miles, middle marker at 5 miles, and an inner marker at 1 mile from touchdown. Pitfalls include stall, spin, gear (if down at 200 knots or more, the gear "goes"), and missing the field. A very enjoyable and challenging jet landing simulation, as it is with little or no visibility. Old Pilots never die; they just change their angle of attack! All for $19.95 including descriptive sheets (WA add 5%). Requires Level II 16K.

PK ENTERPRISES
PO BOX 141
Gig Harbor, Washington
98335

FOR MICRO-COMPUTERS

FREE Special Trial Offer!!



When you order 23 C-10 cassettes featuring Precision transport mechanism and Digital pressure pad. (In individual, dust-proof, plastic boxes with labels) for a low introductory price of $19.95, we will send you FREE 1 C-10 cassette for you to test. If you are not delighted with its performance you can return the unused 23 units for a full refund. Please add $1.05 to cover postage and handling. N.Y. Res. add 7%. VISA and Master Charge accepted.
Write for pricing on larger quantities.

Studio Magnetics Co.,Inc.

SM
Department U
12 Long Island Avenue
Holtsville,N.Y.11742
516-289-3400

BUSINESS PROGRAMS

On Your 4K Level I Computer

Programs written in simple, straight-forward BASIC. Programs written in menu format for ease of use by non-computer personnel. These programs have been tested by a CPA and have been used in business situations.

Each booklet includes a program listing, a complete set of formulas, instructions using each menu and several examples to test the program and familiarize yourself with its operation.

BREAK-EVEN: Compute any unknown: Units to be sold, sales price, variable costs, fixed cost, gross profit or operating leverage.
DEPRECIATION: Compute straight-line, declining balance, sum-of-years digits and cross-over. Start with any year; switch to any other year or method without reentering original data.
RETAIL INVENTORY: Compute any unknown: Beginning inventory, purchases, sales, gross-profit percentage or ending inventory.
SALARY CONVERSION: Convert any wage amount to yearly, monthly, weekly, daily or hourly. Compare salary amounts or list tables.

PRICE PER BOOKLET $ 3.95

"Business Programs" in BASIC
19832 Fifth N.W.
Seattle, WA 98177
(206) 546-5695

Washington residents include 5.3% sales tax.

BACK ISSUE AVAILABILITY

May-Jun 79	Plenty
Jul-Aug 79	Bunches
Nov-Dec 79	Bundles
Jan-Feb 80	Few
May-Jun 80	Lots
Jul-Aug 80	Very few

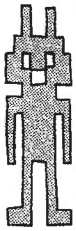
The rest are gone! Please allow 3 to 4 weeks for delivery. All back issues are $3.00 per copy.

ADVERTISER INDEX

US & Foreign ORDER FORM

Zip Codes outside the continental US may add $4.00 for first class mail.

Charge Card Users
Please give name as it appears on your card!



80-US JOURNAL SUBSCRIPTION RENEWAL/ORDER FORM 9/80

□ New □ Renewal

□ 1yr $16 □ 2yrs $31 □ 3yrs $45

□ Payment inclosed OR Bill to my
□ M/C □ BankAmericard □ VISA

Card No. ______________________ Exp date ________

Signature ______________________________

Name (Print or type) ______________________________

Address ______________________________

City, State, Zip ______________________________

(Foreign - except Canada - 1yr $24 2yrs $47 3yrs $68)
ALL Foreign are sent airmail

+

CANADIAN ORDER FORM

80-US JOURNAL SUBSCRIPTION RENEWAL/ORDER FORM (CANADA ONLY) □ New □ Renewal 9/80

□ 1yr $20 □ 2yrs $39 □ 3yrs $55

□ Payment inclosed OR Bill to my
□ M/C □VISA

Card No.______________________Exp.date ______

Signature ______________________________

Name (Print or type) ______________________________

Address ______________________________

City Provence Postal code ______________________________

+

SUBSCRIPTION RENEWAL NOTICE

Your mailing label should tell you the last issue you should recieve. Check the upper right corner, you should find a month and year there, if it says SEP80 then this is your last issue! Please renew soon, it sure saves time and hassle, and you don't want to miss a single issue!

80-US READER INQUIRY CARD – 9/80

Sirs:

I have seen your product/service in the 80-US JOURNAL and would like more information

ON ______________________________

Please send to:

Name ______________________________

Address ______________________________

City St Zip ______________________________

RETURN ADDRESS

Place
Stamp
Here

To:
80-U.S. Journal
3838 South Warner Street
Tacoma, Washington 98409

RETURN ADDRESS

Place
Stamp
Here

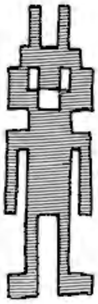
TO:

MICROMATIC SYSTEMS INC.
1303 Powell Street
Vancouver, BC
Canada V5B 1G6

RETURN ADDRESS

Place
Stamp
Here

TO:



Subscribe TODAY, don't miss future issues,
we have some really NEAT things in the works!

Dealer Inquiries Invited

COD & MO shipped immediately.
Check orders - allow two weeks.

Galactic Software Ltd

Dept 80-F, 11520 North Port Washington Road
Mequon, WI 53092 (414) 241-8030

Distributor (East)
Acorn Software Products Inc
Dept 80-F, 634 North Carolina Ave SE
Washington, DC 20003 (202) 544-4259

Distributor (West)
80-U.S. Software
Dept 80-F, 3838 South Warner Street
Tacoma, WA 98409 (206) 475-2219

For further information, please contact the distributor nearest you.

TRS-80 Model I & II Software

(Exquisitely documented)

★Stock Market Monitor

For TRS-80 Model I, Level II, 16K or more. Tracks user selected issues, in a technical system that reflects the issue's performance against the overall market. Set up data is input by the user from the Standard & Poors stock guide or Value Line. Daily issue data, "high", "low", "close" and "volume" are input from any newspaper containing this information. Daily overall market, "volume" and "closing Dow" are also provided from a newspaper. Volume and price changes of an issue, as they compare to volume and price changes of the overall market, are the basis of this system's analysis of the given issue. Clear indications are given as to whether the issue is "out performing", "under performing" or "performing with" the market. Complete video and printed output is provided.

Manual only ...... $ 20.00

Tape version .....$ 89.00
Disk version ......$ 99.00

★ MAIL/FILE System

This program is a true - Name, Address, Phone Number - data base manager intended for business use. We developed this package to meet the most demanding needs for this type of system. It is without a doubt the most sophisticated package of its kind. The entire file is under constant sort, both by name and zip code. No long waits for sorts *EVER*. Data retrieval is alphabetical or zip code order, plus any of 6 other criteria, plus up to 19 codes at once. Two standard label formats and two standard directory formats are supported, plus a *unique* user formatted output which allows the user to set up a custom output format. A "message" line is supported for sophisticated formats, and can be used for *"Resident"*, *"Address Correction Requested"*, *"Service Manager"*, etc. Subscription expiration and renewal notice sublists are simple to control and update with this system. Model I requires at least 32K and a single drive system. Model II requires 64K and has a capacity of 2500 names per disk. Multi-across label versions are available.

Multi-Across Option:
Model I ....$ 19.95
Model II................$ 49.95

Model I version .....$ 99.00
Model II version ....$ 199.00
Manual Mod I or II..$20.00

★ Editor/Assembler

EDAS 4.0 is for Model II TRS-80, 32K or 64K single drive or more. The Text Buffer in a 64K computer has a capacity of about 41,000 characters; 32K computer has a capacity of about 8,500 characters. Source files are written with complete Model II header. Text concatenation is supported. Macros and conditionals are not supported. Sample source file is included, containing all Z-80 mnemonics. The object file may be assembled to disk or to RAM. Disk object files are directly executable. RAM object code may be executed from the editor. The commands and syntax in the Radio Shack Model I Editor Assembler are a small subset of Galactic's EDAS 4.0. Users already familiar with the Model I assembler will be able to function in EDAS 4.0's advanced environment immediately. The complete manual is over 120 pages, and contains complete descriptions of all Z-80 mnemonics and opcodes.

Editor Assembler (EDAS 4.0) Includes complete manual.... $229.00
Complete manual (about 120 pages)....$ 69.00
Commands & Parameters Manual ... $ 29.00

★ Host I/O System

This system was written by the author of the HOST and TERM programs in Radio Shack's Communications Package for the Model I. HOST for the Model II TRS-80 is a simple to use, easy to interface, "KSR" system. It allows you to write BASIC programs that make full use of the Model II's RS-232 channels. Now your Model II can "HOST" terminals, or interact with other computers. All of the features of HOST are available from BASIC or may be accessed at the machine level. Users of HOST do *not* need to have any knowledge of assembler or machine code to use HOST or any of its features. HOST, written to run with Model II TRSDOS version 1.2...with documentation $199.00

★ Ultra Trek

This is an all new concept for this type of game and compares to the others about like checkers compares to chess. ULTRA-TREK is a complex, logical game, intended for the serious contestant. It is doubtful that you will ever master this game, but you will certainly enjoy trying! This program requires TRS-80 Level II, 16K or more. The program is written totally in BASIC and uses 15.5K of RAM.

Complete on cassette with documentation .... $ 14.95

TRS-80 is a trademark of the Tandy Corp

The 80-U.S. Journal
3838 South Warner Street
Tacoma, Washington 98409

Second Class Postage Paid
at Tacoma, Washington 98413
USPS Publication No. 500-470
(ISSN 0199-1035)