80-U.S.

$3.00 PER COPY
$3.50 in Canada

The TRS-80 Users Journal

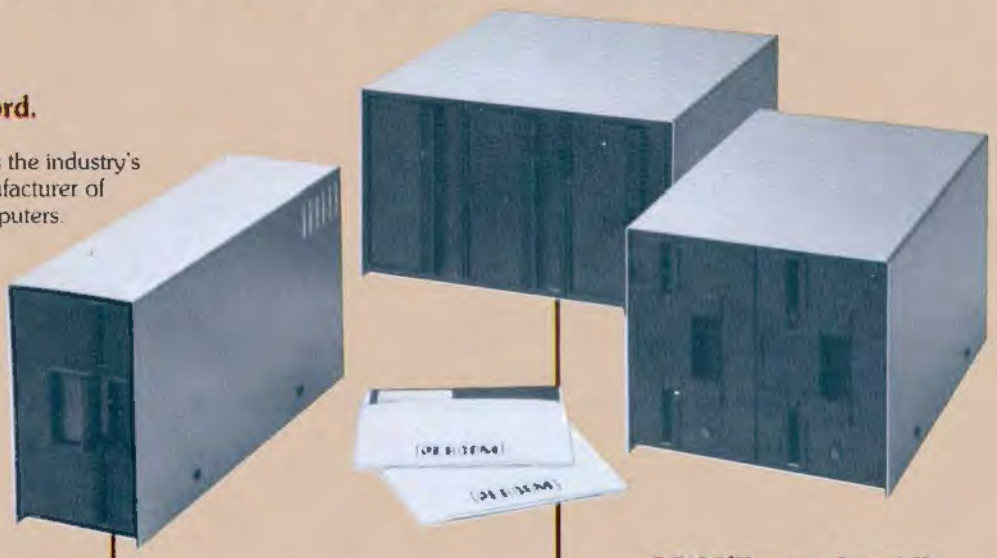
Volume III, Number 6

Nov/Dec 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™ drives and over two-and-one-fourth times as much on TFD-200™ 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™, 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™ 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™ costs only $69.95.

the Separator:™ 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™, 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™ uses Level II BASIC commands

With OS-80™, Level II BASIC commands are used for DOS and Disk BASIC functions.

You extend the OS-80™ utility set indefinitely with your own BASIC disk-resident utilities.

OS-80™ 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™ "Handbook" that you expand and maintain.

All are helpful programming examples.

Price, with instruction manual, $29.95.

NEW! OS-80™ 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

™ 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.

MEMORY EXPANSION FOR TRS-80*

All you have to remember is to plug it in

Memory expansion. It's a field packed with intriguing theories. For instance, it has been suggested that the memory areas of the human brain are transferable from one body to another, like transplanted kidneys. In man or machine, a larger memory is always a welcome acquisition.

If you are interested in expanding your TRS-80 memory without shelling out dollars for a full blown expansion interface, we have just the solution.

Introducing the MT-32. Our new, brilliantly designed Printer/Memory expansion module for the TRS-80. This unit will add 16K or 32K of dynamic RAM to your basic 16K machine. The module also contains circuitry to drive Microtek's MT-80P dot matrix printer, or any other Centronics-compatible printer.

No hardware modification to your TRS-80 is required. Just plug into your bus connector and you are ready to go.

All Microtek products are covered by a one year warranty.

Four configurations are available:

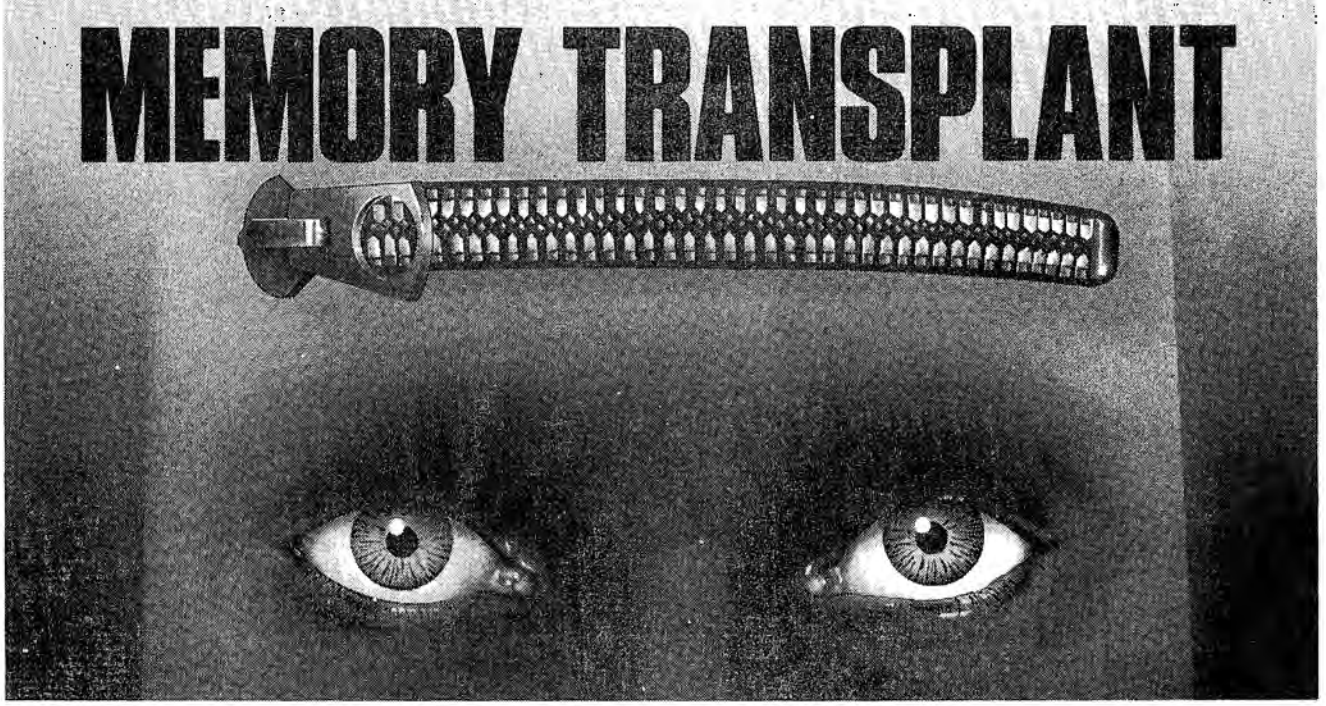
Without RAM in kit form (MT-32K @ $79.50)
Without RAM assembled and tested (MT-32A @ $99.50)
With 16K RAM assembled and tested (MT- 32B @ $159.50)
With 32K RAM assembled and tested (MT- 32C @ $199.50)

Available from Microtek or your nearest computer dealer.

MICROTEK inc.

9514 Chesapeake Drive
San Diego, CA 92123
Tel. (714) 278-0633
TWX 910-335-1269

* TRS-80 is a Registered Trademark of Tandy Corp.



EDITORIAL

There are games, and then, there are games.

Computer games started with Tic-Tac-Toe and Hangman. It was fun to watch the computer act like an intelligent being, making decisions, apparently on its own. Maybe the fascination was that of seeing a machine act semi-human.

A few years ago, David Ahl published a collection of computer games called "Basic Computer Games". It had them all, from "Acey-Ducey" to "Word". It even included a Star Trek game, one of the most popular computer games.

Since then, games have grown up a bit. I remember the first game that really caught my attention a couple of years ago. It was a game called "Star Traders", wherein two or more players tried to establish trade routes among the stars, and could actually buy stock in another player's company. The game was complete with take-overs, mergers and stock splits.

Games took another step with the introduction of Scott Adams' Adventure series. Scott plays out fantasy-adventure fulfillment scenarios which require logic and cunning to complete. Most of them are sprinkled with humor.

Leo Christopherson's games are up-front and open. It's all right there to see, and Leo puts his expertise into detailed fast graphics and a subtle humor into his characters.

Perhaps the next stage in the evolution of games will be something like reading an old classic. Only in the computer version, you are the hero, and can actually affect the outcome to be something the original author hadn't intended. It would also play differently each time, depending on the choices you make. This type of game is not so far removed from the adventures now available.

For some, the whole idea of where games will lead to is a more exciting prospect than playing the current games. A theory of games of strategy was established in 1928 by John von Neumann, who went on with Oskar Morgenstern to develop it as a means of dealing with competitive economic behavior. The advent of the computer creates the possibility of developing simulations of reality.

Real life simulations are still more or less out of the question as far as the microcomputer is concerned. The astronomical number of possibilities simply cannot be handled by the limited memory. Bubble memory may answer the question of memory space, and raise the question of time. (How long would a micro take to search through four gigabytes of look-up table?)

The ultimate game would be one where the players are totally immersed in the play. To create a sense of importance, the players would need to be handicapped, which is to say they should not realize that they are playing a game, and should think of it as reality.

Next, some default values need to be put in to keep the players from copping out. A stomach which needs to be fed would do well in taking care of that. Then we would need some authority figures (governments) and some obvious differences between the players (color?). This will be sure to make any one group feel it may be better than another.

We might even divide the players into two sexes and give each a good healthy sex drive, and then add another group that would provide inhibitions toward using it.

Some game, huh? Does it sound familiar?

And I used to think that games were trivial. ●

Mike

Our Cover

The cover of this issue was the work of Margaret Farrell. It was put together with care, using a photo, clip art and some original oil work. It is our first full four color cover.

The 80-U.S. JOURNAL
VOL III Number 6
Nov/Dec 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 Street, Tacoma, WA 98409.

Return postage is guaranteed
Second Class Postage paid at Tacoma, Washington

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

REVIEWERS
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 House
146 Oxford Street
London, W.1 England

80-U.S.

The JOURNAL for TRS-80 Users

TRS-80 is a trademark of the Tandy Corporation

Vol III Number 6 Nov/Dec 1980

CONTENTS

FEATURES

DEPARTMENTS

REVIEWS

Letters to the Editor

I've been with your Journal from the very start and it has been great to watch your quality expand every issue. My compliments to you and your staff.

Some thoughts on the IBM Selectric printer - I recently returned mine to Micro Computer Devices for service because it would print meaningless characters during listings of programs. I have heard that many SelectraPrint owners have this problem (including you). When my printer was returned MCD included a bulletin which stated that the power up and down sequences are very important and if not done correctly damage to the actuator coils under the typewriter will result. The correct sequence for power up is typewriter on, then electronic package switch on, and for power down, electronic package switch off, then typewriter off. The electronics package switch must be off if the typewriter is to be operated manually.

I thought I would pass this along to other SelectraPrint owners, as the above information was not included when I bought mine and maybe it will save them a $140.00 repair bill.

Received my NEWDOS80 recently and first impression is WOW! I was just getting NEWDOS+ under my belt too. Hope to see some articles on NEWDOS80 and disks in general in the future.

Craig Slutz, Pullman, WA

(Thanks for the information on the SelectraPrint. We also just received NEWDOS80 and are doing our best to digest all its features. Expect an in-depth look at it in the next issue. Ed.)

In your July-August 1980 issue, the second to last paragraph in the right column on page 44 refers to "...another firm operating in Berkeley that claims to have (such) a system (Standard CP/M for the TRS-80 Model I.. ". As we are the only other company that supplies Standard CP/M on 8" drives for the Model I, and as we were located in Berkeley until December 1979, I must conclude that Mr Marler was referring to Parasitic Engineering Inc., and our product, the Maxi Disk system.

I would like to reply as follows:

1. We manufacture our own circuit boards from our own proprietary designs. Our boards have NO extra wires. Our boards provide the lowest error rate and the highest reliability currently available for the Model I.

2. We DO NOT require that orders be prepaid in advance. We do require a 50% deposit on orders. This deposit is refundable, in full, on demand, at any time up to the day we ship. All orders are shipped in the order that deposits are received. In the 12 months that we have been shipping Maxi-Disk systems, the longest delay from receipt of deposit to shipment has been 93 days. The average time has been less than 60 days. The demand for our products has been extraordinary. Customers have even offered to pay us a premium for speedier delivery, although we do not accept such offers. There is an industry-wide shortage of disk drives and we felt that a deposit was the fairest way that we could allocate shipments.

3. Parasitic Engineering Inc. operationally demonstrated the FIRST system that used 8" floppy disk drives and Standard CP/M on the TRS-80 Model I at the Fourth West Coast Computer Faire in May 1979.

4. In July 1979 Parasitic Engineering Inc. shipped the FIRST system that ran Standard CP/M and TRS-DOS on 8" drives on the Model I.

5. Parasitic Engineering Inc. believes that we have shipped more 8" floppy disk systems than any other company. We have satisfied customers all over the world. Many of our customers have given us permission for us to use them as references to verify the quality of Parasitic Engineering's products and service.

6. In March 1980 Parasitic Engineering Inc. demonstrated our production Maxi-Disk systems at the Fifth West Coast Computer Faire. Our booth contained three TRS-80 Maxi-Disk systems with a combined total of over 2.3 million bytes of storage on-line! And one of these systems had over 1.1 million bytes on-line on a single TRS-80! These systems ran CP/M and NEWDOS+ and TRSDOS on 5" and 8" drives! No one else could do this then and no one else can do this now.

7. Any interested person is welcome to call Parasitic Engineering Inc. direct at (415) 839-2636. Every Maxi-Disk ad and brochure we have ever used had our current telephone number printed. Our number is listed in the Oakland-Berkeley telephone directory and is available from information.

8. Interested persons are also welcome to stop by our plant for a demonstration. We are located at 1101 Ninth Ave., Oakland, CA 94606

9. 80-Microcomputing magazine has one of our Maxi-Disk systems (they use it in their Instant Software Division). They have published an excellent review of it on page 15 of the May 1980 issue.

Common sense dictates that we could not still be in business if we held deposits for a year or never shipped complete working Maxi-Disk systems. The company that became Parasitic Engineering Inc. was founded over 5½ years ago. The Maxi-Disk is not our first product, but rather it is the latest in a line of innovative high-performance computer products that have changed the face of the microcomputing industry.

Howard Fullmer, President

Your magazine is proving most popular with my Computing Science students, and rightly so. The magazine contents are usually directly applicable to our hardware and your efforts in adopting the "something for everyone" approach are to be commended. However, the not infrequent occurrence of grammatical and spelling mistakes combined with transposed or missing figure labels is proving to be an increasing annoyance. In a recent edition, the word "hexadecimal" was continually misspelled as "hexidecimal". I would also like to recall your editorial from Vol I, No I, promising an effort to eliminate "continued" articles - such articles have been slipping in.

80-U.S. is an excellent magazine, but it should not become a home for sloppy reviewing and editing.

Peter L Vogel, Vancouver, BC

(You are right on the first count, but all in all, "continued" articles have been held to a minimum. There are times when composition and layout leave no other choice. Thanks for the otherwise good comments. Ed.)

Just had to write and thank you for Robert Labenski's "MULTI" program (*80-U.S. Jul/Aug 80*). I've been looking for something like this for a long time. Needless to say, the "Auto" command did not automatically get your program running. Now with Mr Labenski's program, I just hit the reset button and my program starts running all by itself.

I'm using this for business applications, and it's such a relief to be able to have someone else in the office input data without having to know about operating systems, file allocation, memory sizes and the like.

However, there was a small bug in the program at line 120. The address for 32K systems should be 0B000H and not 0B00H as listed. Other than that the program ran beautifully. Thanks for putting such useful, quality programs in your useful, quality magazine.

Dick Stransky, Madison, WI

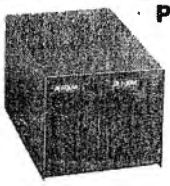
(Yes, we did drop a zero in that listing. The reason it assembled correctly was that it was in a remark statement. We also like MULTI, and use it often. Ed.)

(More Letters on page 6)

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

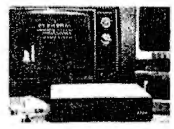
Super fast machine language disk modification utility. Read, Write, Display, and Modify sectors; remove passwords; apply patches, fixes; make backups and much more. On 5" disk with instructions: $29.95.

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 Unlimited 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

Prices and specifications subject to change without notice.

ACCESS UNLIMITED

315 N. Shiloh · Ste. D1 · Garland, TX 75042

(214) 494-0206

(Letters, Continued from Page 4)

The TRS-80 is extremely efficient in converting Hexadecimal numbers to Decimal, but not the other way around. I wrote a short routine to convert decimal to hex since I often needed them for machine or assembly language programs. Perhaps your readers would find this useful also. Keep up the good quality publication.

```
150 CLEAR 1000:DIMX$(200)
160 FORX=0TO15:READX$(X):
    NEXT:CLS
170 INPUT"DECIMAL NUMBER ";D
180 IFD > 65535 PRINT"NUMBER
    OUT OF RANGE":GOTO 280
190 PRINT"HEX NUMBER = ";
200 M=INT(D/256):MX=M*256:
210 L=D-MX
220 FORW=0TO15:FORX=0TO15
230 IF(W*16)+X=M PRINTX$(W)+
    X$(X);:GOTO 250
240 NEXT:NEXT
250 FORW=0TO15:FORX=0TO15
260 IF(W*16)+X=L PRINT X$(W)+
    X$(X):GOTO 280
270 NEXT:NEXT
280 PRINT"ANOTHER ENTRY? "
290 A$=INKEY$:IFA$=""THEN 290
300 IF A$ <> "N" THEN 170
310 DATA 0,1,2,3,4,5,6,7,8,9,A,B,
    C,D,E,F
```

D D Freeman, Garland, TX

I recently installed PROGRAMMA International's 80-GRAPHIX board in my TRS-80. The 80-Graphix, which was advertised on page 31 of the July-August 1980 issue of 80-U.S., actually does what PROGRAMMA claims; it provides the TRS-80 with high-resolution graphics. These are under software control, so that programs using the normal TRS-80 graphics are still compatible with the modified system. In the high-resolution mode, lower case letters - with descenders - can be simulated on the video display. PROGRAMMA supplies a source listing of an assembly language program which creates and places the lower case characters in the video driver. All alphabetic characters appearing on the screen are in lower case unless the shift key is used.

PROGRAMMA cautions in its user documentation that the upper/lower case driver will not work with the Electric Pencil or Scripsit. Nonetheless, I was anxious to try it out with my copy of a Basic text editor program. Everything went along quite smoothly until I printed the text on my line printer. Unfortunately, all shifted characters were printed as lower case and vice-versa.

It began to look as though I would have to make another hardware modification after all. And then I remembered that a "no-hardware" lower case article appeared in a past issue of 80-U.S. I found what I was looking for on page 34 of the 1979 July/August issue: It was Phil Pilgrim's Software Lower Case" article in the System/Command department.

My next problem was my lack of experience in assembly language programming. I only wanted to use that part of Mr Pilgrim's software which would reverse the shift of the characters as they were sent to the line printer. And then, of course, I had to figure out how to work these changes into PROGRAMMA's keyboard driver.

As things turned out, this was no problem at all. A common fault of most program listings appearing in the various microcomputer magazines is the omission of adequate comment statements. In contrast, Mr Pilgrim included an abundance of comments in his program, so that it was easy to see not only which lines I needed, but also how the code functioned. The relevant lines, which I used intact, are:

(See Figure 1, below)

The good news is that after assembling the code, it worked the very first time! Now, thanks to PROGRAMMA, I have upper and lower case letters on my video display when I use my text editor. And thanks to Mr Pilgrim and 80-U.S., I have the same upper and lower case characters in my printed output.

Gareth L Golay, Annandale, NJ

(Good show! It is always nice to see innovative use of what we publish. Ed.)

While reading the July-August 1980 issue of 80-U.S., I find that you printed the graphs for your Survey Results with a Microline 80 Printer or an Okidata Printer. I have just purchased the printer and have been getting good results with it for upper and lower case and the control functions to change type size, line length and line spacing. I do have a problem and that is with graphics. The manual that comes with the unit leaves something out as to programming the graphics mode. I am wondering if one of your staff could make a listing of one of the graphs.

This would give me a starting point which would be a big help. Also if you gents have any further information for the graphics mode it would be appreciated.

The magazine is doing a good job so keep up the good work. Take on more advertisers as this is how a good publication exists.

George F Hatch W9VMG, Ft Wayne, IN

(You're right. The data on using the special commands on the Microline is somewhat lacking. We don't have any data on the Okidata, but here is how to do it on the Microline 80: LPRINTCHR$(27);CHR$(65) will set up 80 characters per line. LPRINTCHR$(27);CHR$(66) sets up 64 characters per line. LPRINTCHR$(29) sets up 132 columns, LPRINTCHR$(30) sets up 80 columns (the default value at power up) and LPRINTCHR$(31) sets up 5 characters per inch (or 40 columns). Also LPRINTCHR$(27);CHR$(54) sets up 6 lines per inch and LPRINTCHR$(27);CHR$(56) sets up 8 lines per inch.

LPRINTing characters 127-191 will result in the same graphics characters as you would see on the screen. If you have NEWDOS, you can use the JKL function to print the graphics if you first POKE 17360,255 J Crocker)

I am sitting here on this hot, muggy New York evening with nothing better to do than fool around with my computer (32K/1 disk), and since I just recently modified one of Phil Pilgrim's programs (KEYMAC Jan-Feb 80 p. 52) to include a repeating keyboard and operate with TRSDOS, I decided to load it.

Well, here I am completely bored and looking at the video display wondering what to do next. I decided since the diskette onboard wasn't too important, I would enter a couple of DOS commands into the MACRO while in DOS and see what would happen. I entered the following commands:

1 BASIC 'ENTER'
2 1 'ENTER' 'ENTER'
3 LOAD"PROGRAM" 'ENTER'
4 RUN 'ENTER' 'SHIFT ENTER'

Naturally, I received a "WHAT?" from DOS since it didn't understand what had just been entered. But upon invoking the MACRO I had just typed I was completely surprised to find that the computer came up with BASIC, 1 file, full memory, loaded and ran the program requested.

Since I had not as yet read anything about this very useful ability of this fine program, I decided to immediately load Scripsit and type this letter.

PS Please keep Phil's System/Command Department going, it is the first thing I look for with each issue.

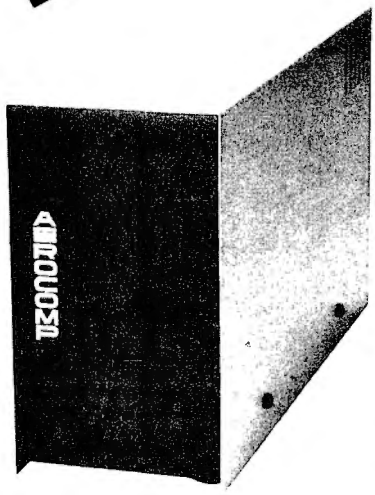
Ross Butera, Levittown, NY

(We will swap you some of our cool weather for some of your free time! Seriously though, we wouldn't think of stoppng Phil's column. We also have a good time waiting to see what he will come up with next. Ed.)

```
            LD      HL,(4026H)      ;LINK IN SPRINT
            LD      (LPRINT+1),HL   ;     .
            LD      HL,SPRINT       ;     .
            LD      (4026H),HL      ;     .
            .
            .
            .
SPRINT      PUSH    AF              ;MUST SAVE FLAGS
            LD      A,C             ;GET CHAR IN A
            OR      20H             ;MAKE LWR CASE IF LTR
            CALL    SHIFT           ;SHIFT IF LETTER
            POP     AF              ;RESTORE FLAGS
LPRINT      JP      $-$             ;PRINT IT.
SHIFT       CP      7BH             ;HIGHER THAN Z?
            CCF                     ;
            RET     C               ; YES, RETURN.
            CP      61H             ;LOWER THAN A?
            RET     C               ; YES, RETURN.
            LD      A,C             ;GET ORIG CHARACTER
            XOR     20H             ;REVERSE ITS SHIFT,
            LD      C,A             ;AND PUT IT BACK.
            RET                     ;ALL DONE.
```

Figure 1

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 wher. 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
● Freight	5.25
Reg.	$479.15
Special	**$399.95**

SPECIAL COMBO EXCEPT WITH NEWDOS/80
Reg. $519.15 **Special $429.95**

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 certified 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!

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

CALL TOLL FREE FOR FAST SERVICE (800) 824-7888, OPERATOR 24

FOR VISA/MASTERCHARGE/C.O.D. ORDERS

California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.

TOLL FREE LINES WILL ACCEPT ORDERS ONLY!

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

Dealers inqiries invited

AEROCOMP

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

ITEMS

At Random

Indian Summers, brisk fall football weather and general elections are back with us. Time flies when you are having fun!

As you have probably seen by now, we have increased our size again. Another 16 pages, and it has finally put us over the 100 page mark. You may also have noted that we have opened up our composition and layout somewhat. We think that with that many pages there ought to be a little space to open it up and give it air.

And by the way, to answer some recent complaints about too many ads, I recently took a survey of my own, using the same measuring stick that is used against us. In their September 1980 issues, 80-Microcomputing had 48.8%, Creative had 49.2% and Byte had a whopping 64.4%. 80-U.S. for that issue had 35.5%. So there. For every page of ads there is a corresponding increase of about two pages of space for articles.

Unfortunately, it also gives us more space in which to screw up. Here are the corrections for the Sep/Oct 80 issue:

Corrections (Ouch! Dept.)

Our Notes in Sep/Oct 80 carried an item on Peek and Poke for the Model II. In the first patch listed, the "find string" (F=C5CD2061) should have been F=AFCD2061. Other than that, it works as advertised. Some callers said that it was not necessary to disable the passwords first, but we found it necessary to do so on our machines.

The article "A Basic Z-80 Disassembler" starting on page 31 needs a line added to correct for certain conditions. Add line 625 as follows:

```
625 IFN1! < -32768THENN3!=N1!+
  32767:N1!=32767+N3!
```

Our NFL-PIX is doing well, it picked 50% the first week and 64% the second. Here is a **correction** for NFL tape only, and only for those few that got out with this glitch: Line 26023 has the word "CHANGB", it should read: CHANGE?". (Replace the B with a question mark and quotes).

In This Issue

Our feature is on page 17, and covers the new Model III from Radio Shack. We had hoped to have "hands on" by this time, but the machines were not shipped yet. We did the best we could from the operator manual.

Sound Ideas - on page 20 is a construction project for those who want noise/music from their TRS-80. The author offers the parts kits in various stages at the end of the article.

Solar heat as an investment is the subject of an article on page 24. Not bad, since the government gives tax advantages if you do it.

Matrix manipulation is the subject of the article starting on page 28. This one should be just right for you algebra students trying to solve for multiple unknowns.

Then there are reviews and other short related subjects (not too numerous to mention, though). We think this will be our best issue yet, but you sort of know we will try and top it next time!

Remember that nice days are made, not had. It's all up to you. And - tell them you saw it in 80-U.S.

Mike

In our July-August 1980 issue on page 44, one of our authors made reference to "another firm operating in Berkekey that claims to have a system such as Omikron's". There are several more references to the "other firm" in that article. Although the other firm was not mentioned by name, Parasitic Engineering, Inc feels they were being referred to (see letters, this issue). It was not the intention of 80-U.S. to slur anyone, and we regret the fact that this implicaton occured.

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) 80 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 1400%.
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 4.0

VTOS 4.0
Operating System Diskette with Operator's Guide
$99.95

VTOS 4.0
Master Reference Manual
$29.95

VTOS 4.0
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. (1024 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 300% 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 4.0 is the only Fully Aproved Operating System by Randy Cook! And it is FANTASTIC!
27) Endorsed by Scott Adams and Lance Micklus!

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)

LTM INCORPORATED

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

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 added later.

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.

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%.

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.

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.

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.

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

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

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

New Products

ED SOFTWARE CATALOG

QUEUE's Catalog #3 is now available. The catalog is a directory of educational software available for Apple, Pet, TRS-80 and Atari. Hunderds of programs from over 40 educational software publishers are grouped by computer, subject matter and grade level. All the programs can be ordered directly through QUEUE. $8.95 from QUEUE, 5 Chapel Hill Dr., Fairfield, CT 06432

MASTER DISK DIRECTORY

MASDIR 1.0, a Master Disk Directory program, has been announced by Micro Systems Software Inc. The program allows one to make a master list of all programs on diskettes and provides video display or printed output. It is capable of listing files by category or extension, file name search, search for free space and more. Available for $29.95 from Micro Systems Software, Inc., 5846 Funston St., Hollywood, FL 33023 (305) 983-3390

EDUCATIONAL SOFTWARE

T.E.S.T., an aid for the classroom teacher is now available from T Y C™ Software. T.E.S.T. contains two programs; a Maintenance Program, and a Test and Drill Program. The Maintenance program allows the user to create a test of up to 35 questions and save it on cassette for use today or whenever needed. Test and Drill is a utility program designed to accept the test prepared by the Maintenance program. With the Test and Drill program, students can either use the questions as a review, take a scored test, or the teacher can have the computer prepare a printed test or worksheet with answer key. Two programs and complete manual, for Level II 16K, $11.95. T Y C™ Software, 40 Stuyvesant Manor, Geneseo, NY 14454 (716) 243-3005

NEW DOS RELEASE FOR MOD I

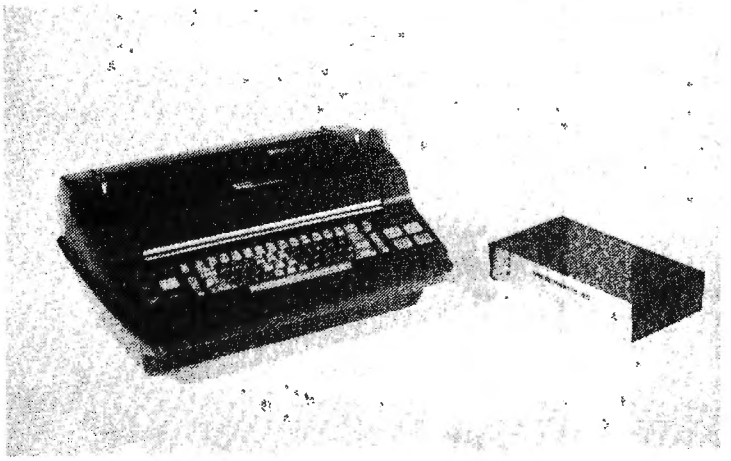
Micro Systems Software Inc has released a new DOS Version 3.1. The new DOS is said to allow setting of the PROT level of a Basic program to EXEC, so it cannot be loaded or listed without the password, and provides Run-only protection for business software. This DOS has the debounce lengthened slightly and will work with drives of up to 80 tracks. Many other features are included. Price is $99.95, from Micro Systems Software Inc 5846 Funston St., Hollywood, FL 33023 (305) 983-3390

SQUADRON LEADER GAMES

Discovery Games has released several games designed to test the mind instead of the reflexes. Among them are: RAF: The Battle of Britain, MIGs and Messerschmidts, Jagdstaffel, Winged Samurai and more. Each game includes an audiotape cassette for 16K Level II, loading instructions, a tactics reference card and a player's manual, all in an attractive bookshelf box. Price is $19.95 from Discovery Games, 936 W Highway 36, St Paul, MN 55113

MICROMATIC 80

The newly formed MICROMATIC Corporation is proud to introduce the MICROMATIC 80, a TTL based interface designed to integrate the TRS-80 and many other small computers at a very affordable price. It consists of an IBM Selectric computer printer, which is thoroughly cleaned and functionally checked due to previous use, combined with a sleek compact interface. The printer has a speed of 8 to 9 cps, with high quality typewriter print suitable for word processing. The MICROMATIC 80 simply connects to the keyboard interface port, or to the expansion interface. All code conversion and timing software are contained within the unit, and no special software is required. The interface is warranteed for 90 days. The MICROMATIC 80 is priced at a low $795., and can be ordered by mail or telephone from The MICROMATIC CORP 5747 West 85th St., Indianapolis, IN 46278 (317) 299-8614



MAXI-DISK

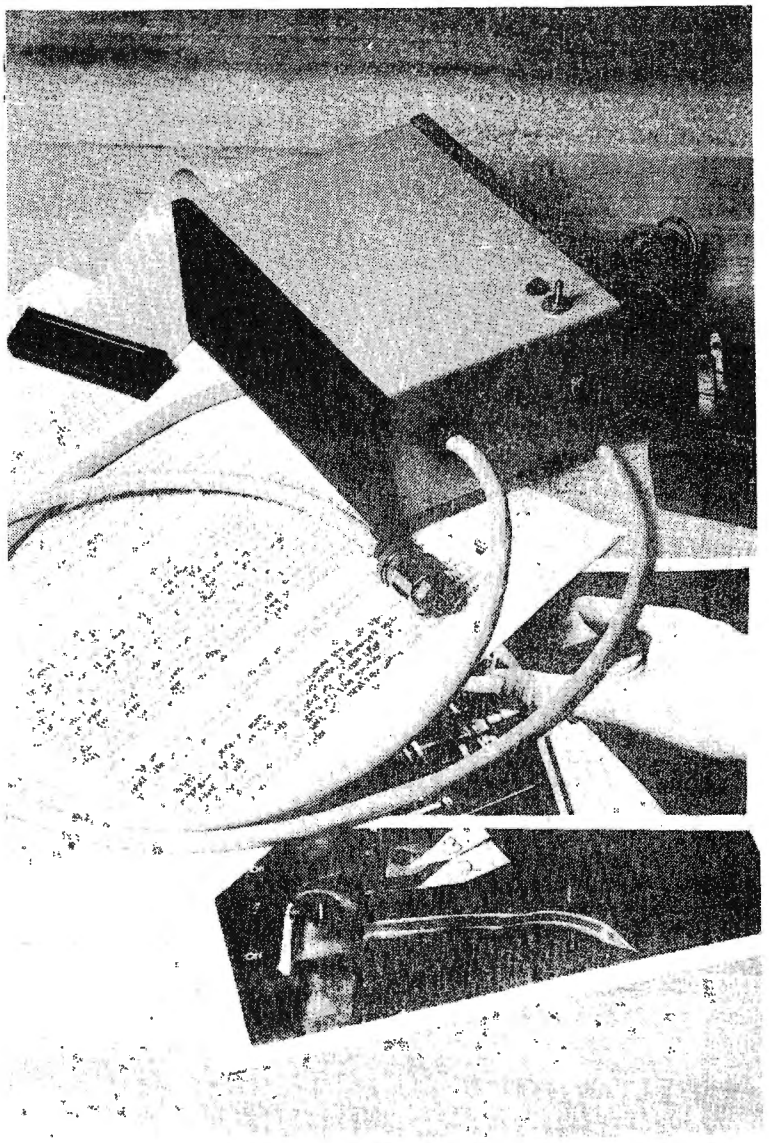
Parasitic Engineering has announced that its Maxi-Disk 8" floppy disk drives are now compatable with the TRS-80 Model II. When used with the Model II, Maxi-Disk drives are functionally identical to the Radio Shack expansion drives, but provide additional features at a competitive price, with no hardware or software changes needed. Each Maxi-Drive is completely self-contained in its own attractive cabinet. Additional drives are simply plugged in. The Radio Shack three drive box is not needed. Maxi-Drives for the Model II cost $845. A three drive cable (one needed) is $60. Delivery is approximately 60 days after receipt of order. Parasitic Engineering Inc., 1101 Ninth Ave. Oakland, CA 94606 (415) 839-2636

EDUCATIONAL SOFTWARE

RITE 80 Software, an enterprising new software firm announces their field tested educational series of programs. Among them: Math Series, Spelling Series, Topics Series, Block Letters, Rollbook and Earth. Earth is an animated globe which can rotate either direction or stop and go at 15 degree intervals with the land shapes changing in three dimensional perspective. All for TRS-80 Level II, cassettes priced at $19.95 each, 10% off for a series, 20% for all three series'. For a descriptive catalog write to RITE 80 Software, 4660 Willens Ave., Woodland Hills, CA 91364

PIGSKIN

Acorn Software Products Inc announces the release of PIGSKIN, a football strategy game for the Model I Level II TRS-80. Two players can compete against each other, or one player can challenge the program in one of five levels of difficulty. Any game in progress may be saved. There is also a spectator mode for demonstrations. PIGSKIN's graphic display of the field shows ball movement and statistics as players employ their strategic skills. Strategy involves the use of 10 offensive plays and 6 defensive positions. PIGSKIN is priced at $9.95 on cassette, or $15.95 on disk. Both are on protected media. Dealers may direct their inquiries to Acorn Software Inc., 634 North Carolina Ave SE, Washington, DC 20003 (202) 544-4259



MOD II UTILITY PACKAGE

RACET Computes announces the availability of a Utility Package for the Model II TRS-80. The package provides the user with eight new and powerful DOS commands. The entire package is written in machine language and is fully documented in a 124 page manual. This further extends the power of the Model II with capabilities such as: Recovery of blown diskettes, Copy multiple files, Examine/Change diskette contents, Catalog diskette directories and Change Disk names and create files. The Utility Package is available from Racet Computes, 702 Palmdale, Orange, CA 92665 (714)637-5016

MEDIAMIX 50/80 INTERFACE

Mediamix of Universal City, CA has extended their product line to include new hardware and software for the TRS-80 Model I, II and III. Their original 50/80 Interface for connecting an IBM Electronic Typewriter Model 50, 60 or 75 has been improved in that it now gets feedback from the typewriter. The DRIVER program that supports this interface uses the feedback to control timing. Benefits include faster typing speed and more control over all of the typewriter's automatic funtions. Mediamix is primarily a mail order company, however dealerships are being set up. Contact Mediamix, PO Box 8775, Universal City, CA 91608

(More New Products on Page 14)

VARKEEP

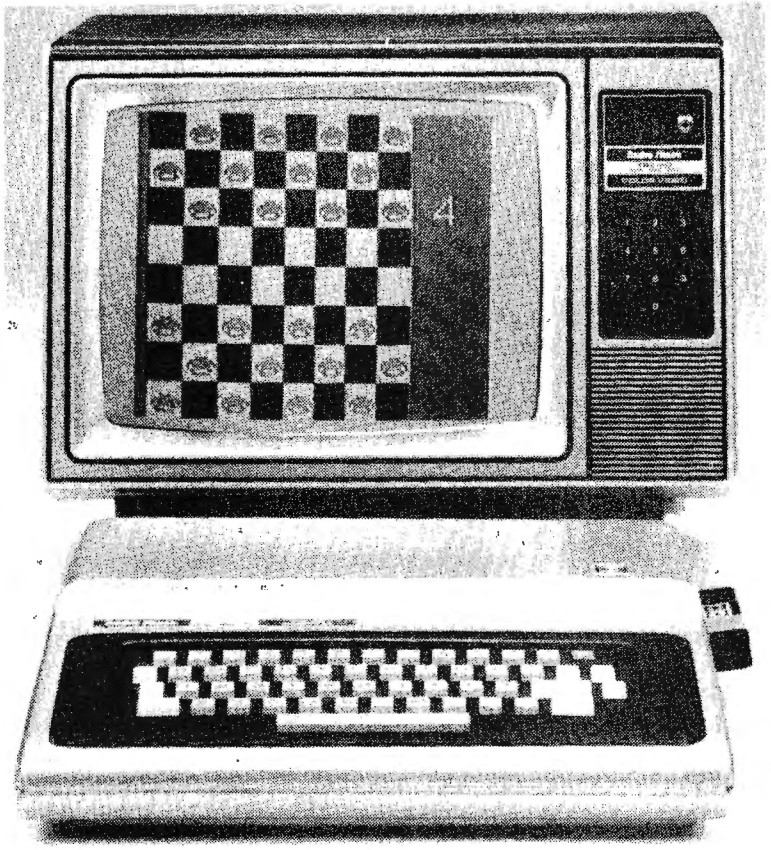
The Alternate Source announces the release of a new memory management utility for TRS-80 Level II and Disk Basic programmers. VARKEEP will add the following powerful commands to Basic: NAME SAVE, NAME RESTORE, NAME DELETE and NAME CLEAR. Using these commands, the programmer can accomplish the following: NAME SAVE allows the programmer to protect the values of all variables from erasure by LOAD, RUN, NEW and/or CLEAR. NAME RESTORE allows the programmer to restore to a program all variables used by a previous program. This provides the ability to easily simulate the powerful CHAIN command found in other Basics. NAME DELETE allows the deletion of variables no longer needed in order to reclaim valuable memory space. This feature allows arrays to be redimensioned. NAME CLEAR will change the amount of string space available to a program while it is running, without losing any variables or a single string. VARKEEP is written in Z-80 and requires about 720 bytes of user RAM. The program is available for $14.95 on cassette, or $16.95 on diskette from The Alternate Source, 1806 Ada Street, Lansing, MI 48910 (517) 485-0344 or (517) 487-3358

SURVEYOR PROGRAMS

Four new low-cost land surveying programs for the TRS-80, developed by Disco-Tech, bring the benefits of computer technology within reach of every surveyor's office. The four programs are Field Note Data Reduction, Coordinate Geometry, Stadia Reduction and Horizontal Curve Staking. They are part of Disco-Tech's SURVEY 80 package, which will also include Vertical Curve Design and Subdivision Earthwork by the end of 1980. To order or request free information on SURVEY 80 or other Disco-Tech programs, write or call DISCO-TECH, Morton Technologies Inc., PO Box 11129, Santa Rosa, CA 95406 (707) 532-1600

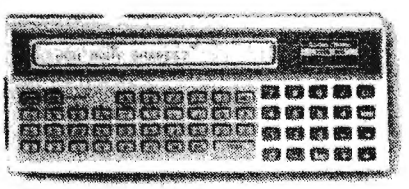
MASS/MAIL SYSTEM

Galactic Software Ltd has introduced the MASS/MAIL System for the Model II TRS-80. All of the features of the Galactic Mail/File system are included in the MASS/MAIL system with the following added features: Drive spanning capabilities now allow the user a total of 10,500 entries on a complete 4-drive system (3500 entries per expansion drive). Two standard label formats and two standard directory formats are supported, plus a unique user formatted output that allows the user to set up a custom output format. The system allows for multiple across printing of labels in standard format. Four auto-entry keys are supported to allow easy input of repetitious entries by the user. All processing is done in a "batch mode". This feature speeds up file maintenance and item addition/deletion, as the user need not be present when processing is taking place. Data retrieval is by alphabetical or ZIP order, plus any of 6 other criteria, plus up to 19 codes at once. Access of information can be done by control number as well. Access time using control number is instantaneous. Access by a key field will always be less than 10 seconds, even with 10,000 names in the system. Files created using the Model II Mail/File System are compatible with the MASS/MAIL System. For additional information contact Galactic Software Ltd., 11520 N Port Washington Road, Mequon, WI 53092 (414) 241-8030



TRS-80 COLOR COMPUTER

Radio Shack's new TRS-80 Color Computer provides color graphics and features instant-load Program Pak™ software that enables the user to instantly program the computer for a variety of educational and recreational purposes. See your local Radio Shack dealer.



TRS-80 POCKET COMPUTER

Considered a breakthrough in computer technology, the TRS-80 Pocket Computer weighs only 6 ounces and is less than 7" long. Yet, it is said to be able to do almost any of the smaller jobs the popular TRS-80 Model I computer can do. See it at your local Radio Shack.

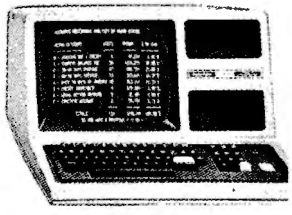
AUTHORIZED
TRS-80® DEALER A301

COMPUTER SPECIALISTS

MODEL II

26-4002
64K 1 Drive
$3499.00

MODEL III

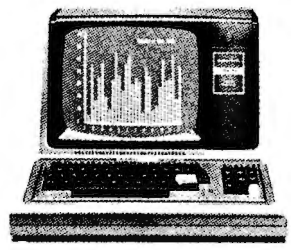


26-1061 4K I.....$630.00
26-1062 16K III.....900.00
26-1063 32K III
2-Drives, RS232.....2246.00

26-1056 16K Level II System with Keypad. $670.00
26-1145 RS-232 Board.....84.00
26-1140 "O" K Interface.....249.00
26-1141 "16" K Interface.....365.00
26-1142 "32" K Interface.....476.00
26-1160 Mini Disk - Drive O.....424.00
26-1161 Mini Disk - Additional.....424.00
26-1154 Lineprinter II.....720.00
26-1156 Lineprinter III.....1799.00
26-1180 Voice Synthesiser.....339.00
26-1181 VOXBOX.....145.00
26-1104 Factory Upper/Lower Case Modifcation Installed.....70.00
26-1506 Scripsit - Tape.....60.00
26-1563 Scripsit - Disk.....85.00

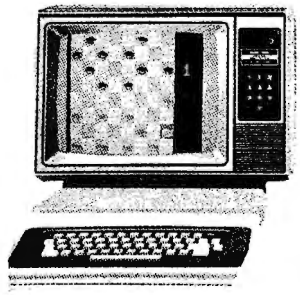
NOTE: Call for availability of VIDEO TEX, Model III, Color, and other new products.

MODEL I

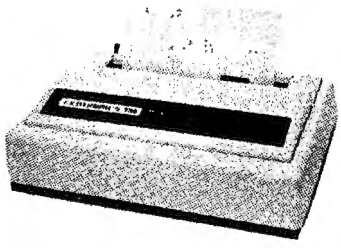


26-1054
4K Level II
$552.00

COLOR



26-3001 4K.....$360.00
26-3002 16K.....540.00
26-3010 Color Video.....360.00
26-1206 Recorder.....54.00
26-3008 Joysticks.....22.50



CENTRONICS

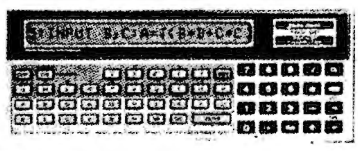
Fast 100 CPS Centronics
730 Printer.....$675.00
Text Quality Centronics
737 Printer.....$850.00

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

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

Novation Cat Modem..$149.00
CCA Data Management System.....72.00
Adventure Games Games 1-9 each.....14.00

Pocket Computer



26-3501 1.9K P.C.....$225.00
26-3503 Cassette I/F.....45.00
14-812 Recorder.....72.00

Acorn Software Products, Inc.

GAMES:
Alien Invasion.....$9.00
Stock Market.....9.00
Star Trek.....9.00
Block 'Em.....9.00
Ting-Tong.....9.00
UTILITIES:
System Savers.....14.00
EDUCATION:
Language Teacher.....18.00

FREE: PRICE LIST UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

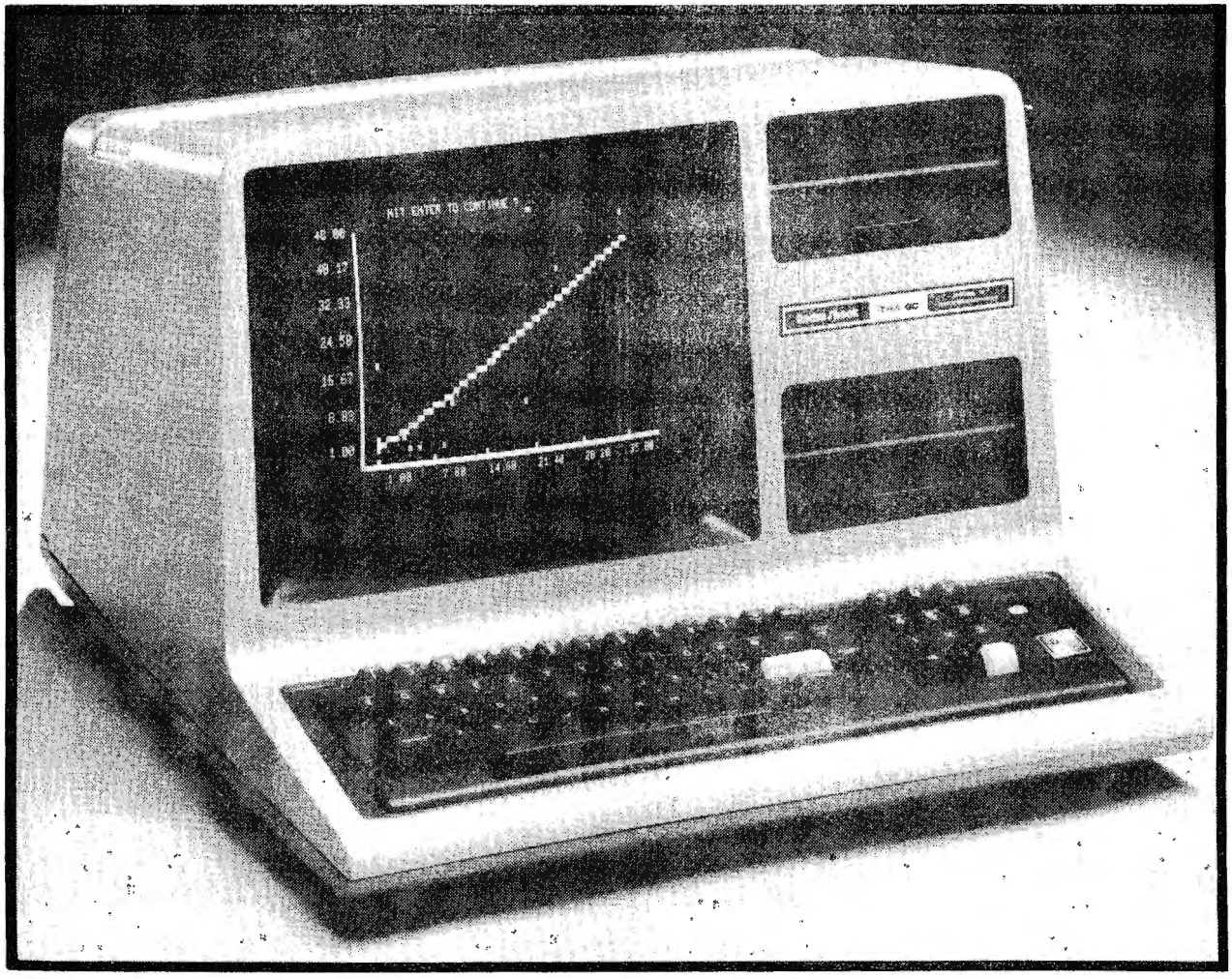
Immediate Shipment From Stock on Most Items

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.



The New Model III from Radio Shack

A couple of months ago, Radio Shack announced their new Model III TRS-80 Microcomputer. In spite of rumors about a TRS-90, color and other assorted features, this one turns out to be our old friend - the Model I, with some new features.

But what features! For openers, the entire computer is contained in one case, including the keyboard. Further, there is only one power cord to plug in (hooray!). And there is space for two 5 inch disk drives to the right of the screen.

Those drives, by the way, are dual density, 40 track, and can contain up to 175K per data disk. Most programs are upward compatible with Model I, except for programs using "non-standard" programming techniques.

The Model III comes in several versions. It is available in a 4K Level I version ($699.), or a 16K Level II version ($999.). The two-disk, 32K version is very nicely priced at $2495. Memory is expandable to 48K, although there is no listing in the Radio Shack literature for a 48K version, only a 32K with add-on memory.

The keyboard on the Model III is complete with keypad, and the cassette is built-in. The cassette loads at 1500 baud, about three times faster than Model I.

Software already available for the Model III includes general ledger, payroll, inventory, payables, receivables, advanced statistics and others. Because of the compatibility of this machine with Model I software, it shouldn't be long until most of what is available for Model I will be running on Model III.

DIFFERENCES

Level I Basic

Model III Level I Basic is compatible with the Model I Level I. All programs written for the 4K Level I should load from tape and run without revision on the Model III. There may be minor differences in the characters displayed on the screen. Model III Level I Basic includes the commands LLIST and LPRINT.

Level II Basic

Model III Level II Basic is compatible with the Model I Level II. It requires 258 more bytes of memory for internal use than did Model I. Except for this memory restriction, all programs written for the 4 to 48K Model I in Level II Basic should load from tape and execute without revision on the Model III. Again, there may be some differences in the characters actually displayed on the screen.

Model III contains lowercase as a standard feature. The default condition for the keyboard is "CAPS" mode. In the Model I the default was non-shift for capitals, and shift characters returned the lower case ASCII code. The SHIFT-0 key in the Model III is a toggle which alternates between CAPS mode and lowercase mode, where non-shifted characters are lowercase and shifted characters are upper case.

The use of non-documented programming techniques or memory locations in Model I software may cause it to function improperly on the Model III. Some examples:

1) Use of POKE/PEEK to examine or alter memory which was not documented or reserved for system use.

2) Use of POKE to perform line printer output.

3) Use of POKE/PEEK or OUT to manipulate the cassette.

4) USR routines which reside in restricted areas or call non-documented routines in the ROM.

New Level II Features

The keyboard has an automatic repeat key, CAPS key, and can be routed to other devices.

Video

The video displays standard upper/lower ASCII characters. It also contains two additional character sets which can be swapped with space compression codes under software control. Further, it has a scroll protect function and a user definable cursor character. There is a software cursor blink function, a screen print function and the screen contents can be routed to other devices.

Cassette

The cassette features user (or software) selectable speeds of 500 or 1500 baud. Basic data tapes operate at 500 baud only (and automatically). During cassette operation the BREAK key can terminate the operation at any time. The cassette can be routed to other devices.

Line Printer

The line printer has user definable line width and lines per page. Printer operations can be terminated at any time by use of the BREAK key, and the printer output can be routed to other devices.

Serial Communication (RS 232)

Serial communication can be routed to other devices, and are supported by ROM routines.

Clock/Date

The Model III contains a real-time, displayable clock, which can be set and turned on or off from Level II Basic using the POKE function. Current date is also supported.

If your software needs to determine which Model (I or III) it is operating in, you can examine memory location 0125H. In Model I this memory address contains 20H, while in Model III it contains 49H.

The TRS-80 Model III uses a double density recording technique to achieve approximately twice the storage capacity per disk than did the Model I. Because of this difference, diskettes produced on Model I will need to be converted to the Model III format. A program called "CONVERT" is provided on the Model III TRSDOS diskette to perform this function.

A Model III diskette is organized in the following format:

40 tracks
240 granules (6 grans per track)
720 sectors (3 sectors per granule)
184,320 bytes (256 bytes per sector)

User space on the system diskette (required in drive 0), is approximately 29.5 tracks, 177 granules, 531 sectors or 135,936 bytes. User space on data diskettes (drives 1, 2 or 3) is 39 tracks, 234 granules, 702 sectors or 179,712 bytes.

Model III Disk Basic

Model III Disk Basic is compatible with Model I Disk Basic. Programs written in Disk Basic for Model I should execute without revision on the Model III. There may be some minor differences in user programmed input routines which control the cursor. After the Model I disks are converted to the Model III format, you may want to reorganize the software to take advantage of the increased space.

Model III DOS (TRSDOS)

The Model III TRSDOS is significantly different from the Model I TRSDOS in terms of internal operation. The command syntax should be familiar to the Model I user, though. A "HELP" command is provided to explain the function and syntax of most library commands. In addition to most of the commands found in Model I DOS, there are several new commands in Model III. These are:

BUILD creates a command file to be executed with "DO".
CLEAR clears user memory and resets memory size.
CLS clears the video screen from DOS READY.
COPY copies single (or multiple) disk files.
CREATE creates a disk file and sets size and configuration.
DO executes a command file (created by BUILD).
DUAL causes all output to go to both video and printer.
FORMS sets line printer paging or executes top of form.
HELP displays a library command's syntax and function.
MASTER sets a drive to be the default READ/WRITE drive.
PATCH makes changes to disk files.
PAUSE pauses execution (usually in a DO file).
PURGE allows multiple file deletion.
RELO changes the loading address of a disk file.
ROUTE changes input/output to a different device.
SETCOM initializes the RS 232 drivers.
TAPE writes a SYSTEM tape from disk or memory or reads it.
WP is software write protect for a disk drive.

Summary

The Model II uses a Z-80 microprocessor which runs at 2.0275 Mhz. The video is 16 lines of 64 characters, and is memory mapped to the same locations as is the Model I. The keyboard is identical to the Model I, and the ten-keypad is standard. SHIFT(down arrow) is used as a control key.

The compatibility, style and features of the Model III are sure to make this an excellent choice for home and small business use. With a good printer and Scripsit, it should be a very neat word processing package, and - best of all, the price is right. ●

from

The Programmers' Guild

IRV

IRV, one of the most powerful utility programs available, turns your keyboard into a *SUPERKEYBOARD!*

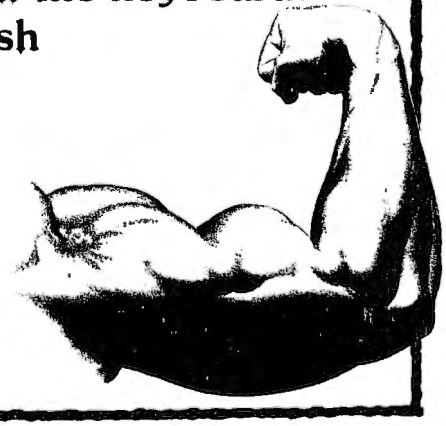
Now you can have single key programming. IRV comes complete with its own keyboard definitions, or up to 255 characters can be assigned to every key, including (ENTER) and (BREAK). You can enter often used BASIC words, variable names or even entire lines. Even functions, such as RUN, LIST, or EDIT can be entered with a *single keystroke.*

The relocate feature of IRV is unique, in that it allows single line relocation and renumbering. You can merge lines using the EDIT function and a single keystroke.

As a video editor, IRV is so powerful, you'll wonder how you got along without it. Full cursor control, blinking cursor, block movement and special erase functions are just the beginning. Frequently used video graphics blocks can be saved and used again and again.

Even IRV's minor virtues are impressive. You can have auto repeat with any key, including programmed functions. You won't have to pull plugs or fiddle with a control box to rewind or fast-forward a tape. The cassette recorder can be controlled from the keyboard.

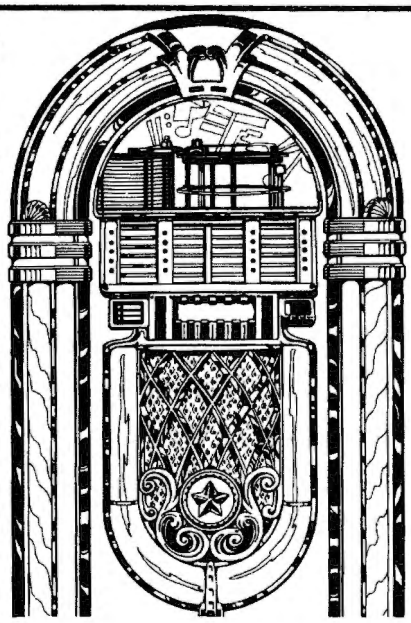
If you are a creative programmer (or wish to be), you *need* the power and convenience of IRV! (DOS compatible).



The Programmers' Guild
P.O. Box 66 Dept V
Peterborough, NH 03458
(603) 924-6065

$24.95 Tape
$29.95 Disk

Send Check or M/O (Shipment within Five Days of Receipt)



The Sound Idea

by Tim R Lantz
Canoga Park, CA

Now you can add the dynamics of real sound and music to your programs. Unlike other methods that gobble up computer time sending streams of data to a reproducer, the SOUND IDEA is a self contained generator requiring attention from your computer only when you want to change the sound *it* is producing. And just to keep life really interesting, there are no less than three separately programmable audio channels available.

Architecture

The Sound Idea is built around the General Instruments GI AI-3-8912 chip. The chip is controlled by first opening one of 14 registers (0-13), and then writing a command into that register. Basically, there are three audio channels, each of which may produce a tone or noise, or both. Further, the volume of each of the channels can be controlled by you, or placed under the control of an envelope generator. If this seems a little overwhelming at first glance, it won't when we break it down.

Communication

The Sound Idea is a ported device, which means that rather than taking up memory space, it is treated as an I/O device and uses OUT commands. The cassette recorder you most likely have is the same type of device. The cassette is assigned port 255, and receives data by way of the command OUT 255, D, where D is the data to be recorded.

The Sound Idea is assigned to two ports. Since the GI chip has 14 registers, the first step is to open the register into which you will send your command. To open a register, use the command OUT 252, N, where N is the number of the register you want to command. This register will remain open until you open a different register with another OUT 252,N.

The second step is to place the decimal value of the command you desire into the currently open register. To enter the value, use the command OUT 253, V, where V is the decimal value of the data you want to place in the currently open register. For example:

```
100 OUT 252,0 : REM OPEN REGISTER 0
110 OUT 253,213 : REM FILL OPEN REGISTER WITH 213
```

This two line routine opened register 0 and then filled it with the value 213. What will be handy to remember is that register 0 is still open. If, for instance, you had just established the pitch of a tone with these two lines, all that would be needed to change the pitch is another value:

```
120 OUT 253,100 : REM FILL OPEN REGISTER WITH 100
```

Now, register 0 contains the value 100, which raised the pitch.

Construction

Referring to the schematic, R1, C2 and three of the inverter gates of Z2 form the oscillator which feeds pin 15 of Z4 (the AY-3-8912) sound generator chip. Altering the value of C2 will "tweek" the pitch of the tone generators.

The other three gates of Z2, along with Z1 and Z3, decode the commands OUT 252,N and OUT 253,V and change the bus control inside Z4 to direct the data lines to the register select circuits, or to fill the currently open register.

The prototype was built on Vector board and wirewrapped. The only physical layout consideration was to insure that the oscillator (Z2), C2 and R1 were kept close together and as close as practical to pin 15 of Z4.

Flat ribbon cable was used to run to the edge connector J1 which plugs directly into the expansion interface bus extension card edge, or the rear of the keyboard. Radio Shack does not specify the amount of current available from their +5 volt source, so no guarantee can be made that the Sound Idea can be run from this source. In some versions of the Level II keyboard, the +5 volt source has been removed altogether. In any case, it worked just fine on my 16K Level II with a fully loaded expansion box. If an external supply is used, try Radio Shack's 270-155, set to the 4.5 volt position. Due to the high level of ripple, the value of C1 is 2200 mfd. Also, the power line from pin 39 of J1 must *not* be connected to the external supply!

The audio output can easily drive the finest pair of headphones (which pleased my wife no end during the experiments). If you want an inexpensive self-contained audio amplifier, I had good luck with Radio Shack's Archer 277-1008A mini amplifier-speaker for about $11.00 plus a 9 volt battery.

For the super technical individual, the unused pins of Z4 are an external port for use as your imagination decides. Also, with modification to the bus control decode logic, the chip can input to the computer the status and contents of any of the registers. If you acquire GI's information on their chip, you will find you have a 61 page manual, loaded with data enough to blow your mind.

In any event, this Sound Idea will provide completely new vistas to those creative programmers frustrated with a silent screen. You can even unstrap the three output channels 1, 4 and 5 of Z4, use three amplifiers and have not just stereo, but Trireo!

Register Functions

Now let's look at the functions of the 14 registers. Rather than take them in numerical order, let's examine them in a kind of logical order that makes it easier to understand.

Register 7 - Enable control - OUT 252,7. This register establishes what kind of sound or mixture each of the three channels (A, B or C) will produce. Each channel can produce a tone or noise, or both. The decimal value to be used to control this mixture is provided in Table 1. Rather than list all 64 possible combinations, the table is split into noise and tone values. Add the two values together and use the total.

Registers 8, 9 and 10 - Volume Control - OUT 252,8 (9) or (10). Each of the three channels has its own volume control. Channel A is controlled by the value output into Register 8, Channel B by register 9 and C by 10. The value 0 is effectively off, and the volume increases as the value used gets larger, up to 15, the loudest volume. The value of 16, instead of establishing a volume level, transfers control of the volume for that channel into the hands of the envelope generator. (See Registers 11, 12 & 13).

Register 13 - Envelope Shape - OUT 252,13. This register establishes the shape of the envelope which is used to control the volume of any or all three channels whose volume control registers were sent a value of 16. There are 8 distinct shapes of

TABLE 1

REGISTER #7 - ENABLE CONTROL

DECIMAL VALUE				
ø	A	B	C	
1	-	B	C	
2	A	-	C	
3	-	-	C	TONE ENABLED CHANNELS
4	A	B	-	
5	-	B	-	
6	A	-	-	
7	-	-	-	
ø	A	B	C	
8	-	B	C	
16	A	-	C	
24	-	-	C	NOISE ENABLED CHANNELS
32	A	B	-	
4ø	-	B	-	
48	A	-	-	
56	-	-	-	

OUTPUT THE TOTAL INTO OPENED REGISTER #7

EXAMPLE:

```
1ØØ OUT 252,7  :  OUT 253,Ø  : REM ALL 3 CHANNELS HAVE
                                  BOTH NOISE AND TONE ON
```

EXAMPLE:

```
1ØØ OUT 252,7  :  OUT 253,62 : REM ONLY CHANNEL A IS
                                  ENABLED FOR TONE
```

TABLE 2

ENVELOPE SHAPE

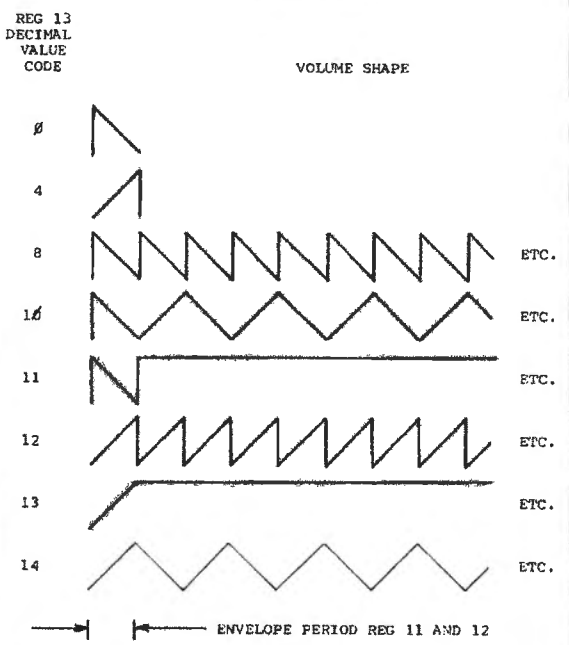


TABLE 3

CHROMATIC SCALE

OCTAVE	NOTE	DECIMAL VALUE COARSE	DECIMAL VALUE FINE	OCTAVE	NOTE	DECIMAL VALUE COARSE	DECIMAL VALUE FINE	OCTAVE	NOTE	DECIMAL VALUE COARSE	DECIMAL VALUE FINE
1	C	13	93	3	G#	2	27	6	E	ø	85
1	C#	12	156	3	A	1	252	6	F	ø	8ø
1	D	11	231	3	A#	1	224	6	F#	ø	76
1	D#	11	6ø	3	B	1	197	6	G	ø	71
1	E	1ø	155	4	C	1	172	6	G#	ø	67
1	F	1ø	2	4	C#	1	148	6	A	ø	64
1	F#	9	115	4	D	1	125	6	A#	ø	6ø
1	G	8	235	4	D#	1	1ø4	6	B	ø	57
1	G#	8	1ø7	4	E	1	83	7	C	ø	53
1	A	7	242	4	F	1	64	7	C#	ø	5ø
1	A#	7	128	4	F#	1	46	7	D	ø	48
1	B	7	2ø	4	G	1	29	7	D#	ø	45
2	C	6	174	4	G#	1	13	7	E	ø	42
2	C#	6	78	4	A	ø	254	7	F	ø	4ø
2	D	5	244	4	A#	ø	24ø	7	F#	ø	38
2	D#	5	158	4	B	ø	226	7	G	ø	36
2	E	5	77	5	C	ø	214	7	G#	ø	34
2	F	5	1	5	C#	ø	2ø2	7	A	ø	32
2	F#	4	185	5	D	ø	19ø	7	A	ø	3ø
2	G	4	117	5	D#	ø	18ø	7	B	ø	28
2	G#	4	53	5	E	ø	17ø	8	C	ø	27
2	A	3	249	5	F	ø	16ø	8	C#	ø	25
2	A#	3	192	5	F#	ø	151	8	D	ø	24
2	B	3	138	5	G	ø	143	8	D#	ø	22
3	C	3	87	5	G#	ø	135	8	E	ø	21
3	C#	3	39	5	A	ø	127	8	F	ø	2ø
3	D	2	25ø	5	A#	ø	12ø	8	F#	ø	19
3	D#	2	2ø7	5	B	ø	113	8	G	ø	18
3	E	2	167	6	C	ø	1ø7	8	G#	ø	17
3	F	2	129	6	C#	ø	1ø1	8	A	ø	16
3	F#	2	93	6	D	ø	95	8	A#	ø	15
3	G	2	59	6	D#	ø	9ø	8	B	ø	14

TABLE 4

REGISTER FUNCTIONS SUMMARY

REG #	FUNCTION	COMMAND VALUES	COMMENTS
ø	CHANNEL A TONE - FINE	ø-255	LOW VALUES PRODUCE THE HIGH PITCH
1	CHANNEL A TONE - COARSE	ø-15	
2	CHANNEL B TONE - FINE	ø-255	
3	CHANNEL B TONE - COARSE	ø-15	
4	CHANNEL C TONE - FINE	ø-255	
5	CHANNEL C TONE - COARSE	ø-15	
6	NOISE GENERATOR	1-31	
7	ENABLE CONTROL	CODED	SEE TABLE 1
8	CHANNEL A VOLUME	ø-15 OR 16	ø IS OFF THRU 15 THE LOUDEST. 16 IS ENVELOPE CONT.
9	CHANNEL B VOLUME	ø-15 OR 16	
1ø	CHANNEL C VOLUME	ø-15 OR 16	
11	ENVELOPE PERIOD - FINE	ø-255	LOW VALUE IS FASTEST TIME
12	ENVELOPE PERIOD - COARSE	ø-255	
13	ENVELOPE SHAPE	CODED	SEE TABLE 2

the volume pattern envelope. The decimal value to be loaded into Register 13 and the shape produced, are shown in Table 2.

The start of the volume shape begins immediately when Register 13 is filled with the desired value. Therefore, it is usually the last register to be programmed when used. Also note that while codes 0 and 4 will cause a single burst of volume and then go silent, the balance of the codes will sustain sound indefinitely until the volume control register is loaded with a value of 0. Even then, of course, the envelope generator is still producing the selected volume shape.

Registers 11 and 12 - Envelope Period - OUT 252,11 for fine, and OUT 252,12 for coarse. There are two registers that together establish the time that the envelope generator takes to produce the slope portion of the volume waveshape. Register 12 is the coarse or main time period control, while Register 11 is the fine time period control. The legal values that both registers will accept range from 0 through 255 only. The smaller the number, the shorter the time period, in other words, the faster the slope of the waveform. To help you establish the relationship between the coarse and fine control registers, consider the value in the coarse register as a whole number, and the value in the fine register as a decimal addition to that whole number. The only caution is that the minimum value is 000.001 where the whole number 0 is output to the coarse register, and the number 1 is output to the fine register.

The reason for this is that it must take some time for the slope to ramp, so a value of 0 in both registers is not legal. To put some sort of time equivalent to the values, a value of 20 in register 12 and 0 in register 11 will produce a duration of approximately one half second.

If the fine register has been incremented to its max value of 255, and you still want to increase the slope time by one more least significant amount, the new values will be 21 in register 12 and 0 in register 11. In other words, the amount has been increased by 1/255 of the value in the coarse register.

So far we have looked at the enable register, which establishes the mix of tone and/or noise for each of the three channels and the volume which can be fixed or placed under the control of the envelope generator. Now let us examine the kind of noise and pitch of the tones.

Register 6 - NOISE GENERATOR - OUT 252,6. The noise that can be passed to any or all of the three channels has a pitch, or basic tone range, from a low grumble to a high hiss sound. The legal decimal values are 1 to 31, where 1 is the highest hiss and 31 is the lowest grumble.

Registers 0 through 5 - Tone Control. Each of the three tone generators has a coarse and fine adjustment register pair quite similar to the envelope period. Registers 0 and 1 are the fine and coarse pitch controls for generator A. Registers 2 and 3 are the fine and coarse for generator B and 4 and 5 are the fine and coarse for C. The legal decimal values are 0 to 255 for fine, and 0 through 15 for coarse. As with the envelope period, the pairs of registers determine the total value which must be no less than 00.001. The lower the total value, the higher the pitch. You will find that the range will not only exceed your hearing range, but also the range of the best audio amplifier you might want to use. The use of the sub and super audible tones can be used, however, to modulate the mixed tones you can hear. For instance, a low audible tone mixed with a sub-audible produces a motorboat sound.

Table 3 is provided for the music buff. Assuming that the oscillator is tweeked so that the fifth octave A is close to 880 Hz, the decimal values are provided in the table for the coarse and fine register pairs to assist in making quite recognizable tunes. Or, you can create your own unrecognizable tunes.

Sample Program

Now, let's have some fun with the Sound Idea:

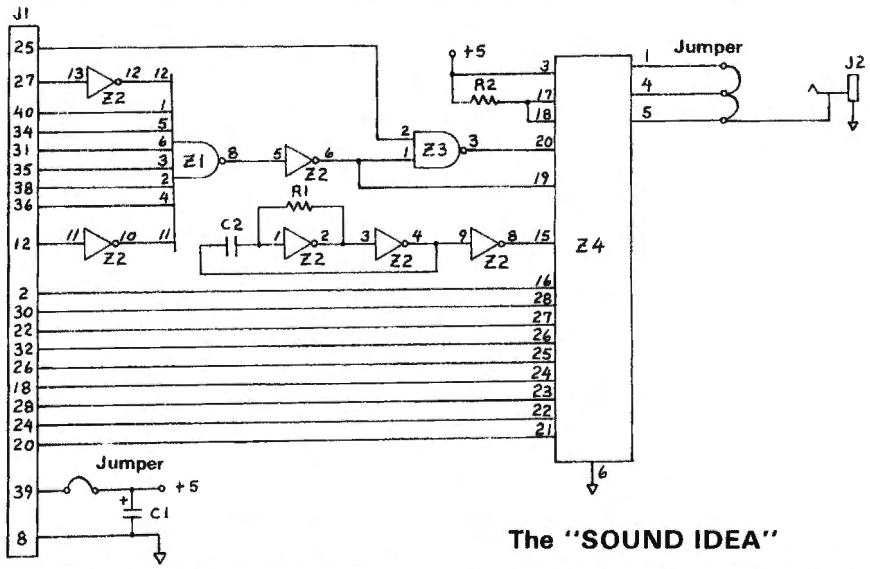
```
100 FOR R=0TO13:REM HOUSEKEEP ALL REGISTERS TO
      A VALUE OF ZERO
110 OUT 252,R : OUT 253,0
120 NEXT R
200 OUT 252,7 : OUT 253,62 : REM ENABLE TONE ON CH
      A
210 OUT 252,8 : OUT 253,10 : REM FIX VOL FOR CH A TO
      LEVEL 10
220 OUT 252,0 : REM OPEN FINE TONE FOR CH A
230 FOR V=48TO192 : REM LOOP VAL V FROM 48TO192
240 OUT 253,V : REM FILL REG 0 WITH VAL V
250 NEXT V
300 OUT 252,6 : OUT 253,25 : REM SET NOISE PITCH
310 OUT 252,7 : OUT 253,15 : REM ENABLE ONLY NOISE
      ON CH B AND C
320 OUT 252,9 : OUT 253,16 : REM PLACE CH B VOL
      UNDER CONTROL OF ENVELOPE
330 OUT 252,10 : OUT 253,16 : REM PLACE CH C VOL
      UNDER CONTROL OF ENVELOPE
340 OUT 252,12 : OUT 253,150 : REM SET ENVELOPE
      PERIOD TO APPROX 4 SECONDS
350 OUT 252,13 : OUT 253,0 : REM START ENVELOPE
      PATTERN 0
```

Just for laughs, after running the sample program, enter from the keyboard without any line number, the following:

OUT 253,0

You will remember that everything has been set up for the explosion, and that register 13 is still open. So, when you output the value 0, you reinitiated the one-shot volume pattern 0. Take this opportunity to play with some of the other volume shapes. Table 4 is a summary of the registers and their functions. Have a real blast! ●

Kits and assembled units are available in the following configurations. Printed Circuit Kit - $15.00 (includes PC board, instructions and programming information.) Mini-Kit $52.00 (includes PC board, AY-3-8912 integrated circuit, ribbon cable and connector, instructions and programming information). Full Kit - $72.00 (Includes all components, chassis, cable, instructions and programming information) Unit, Assembled and Tested - $99.00 (includes programming information). Send check or money order to: Lantz & Youngren Enterprises, PO Box 1283, Canoga Park, CA 91304 (CA residents add 6% sales tax to price Inquires include self addressed stamped envelope or call evenings or weekends (213) 882-7872)



The "SOUND IDEA"

PARTS LIST

C1 2200 MFD
C2 820 PFD
R1 200 ohm 1/4 watt
R2 5.1K 1/4 watt
Z1 74LS30
Z2 7404
Z3 74LS00
Z4 AY-3-8912
J1 AMP 88103
J2 Miniature Phone Jack

Transform your programming language into a real data base management system

*SPECIAL NOTE! HDBS interfaces with CBASIC under CP/M, with TRS-80 (Models I and II), and with APPLE DOS.

NEW!

A major breakthrough in data handling for micros! At best, existing programming languages provide only rudimentary file handling capabilities. With HDBS (Hierarchical Data Base System), you can now *transform your programming language* from an ordinary file handling system into a flexible, powerful, and productive data base management system.

HDBS provides the two components any genuine data base system must have:

1. The Data Description Language (DDL), an easy-to-use, stand-alone language for specifying blueprints or schemas of data base organization.
2. The Data Manipulation Language (DML) for manipulating (storing, accessing, modifying, deleting) data organized according to a data base blueprint. The DML extends your data handling capabilities by allowing you to embed powerful data manipulation commands in the programs you write. The commands are stated through the "CALL" facilities of your programming language.

HDBS users are...

- free of the need to design files.
- free of the need to merge files; there is no scattering of data over several files. Records of many different types are automatically maintained in a single, integrated organization...one that can be spread over numerous disk drives.
- free of the need to be concerned with disk I/O because it is handled automatically by HDBS.

HDBS offers...

- hierarchical schema designs.
- data base schemas of up to 254 record types; each record type may contain up to 255 fields. The size of a field may be up to several thousand bytes in length.
- data bases spread over one to eight disk drives. HDBS is independent of the sizes and types of drives.
- user-defined names for fields, record types, and sets.
- records maintainable in several sorted orders...and in other orders as well.
- *written in machine language for maximum execution efficiency and minimal memory usage.*
- available versions: Z80 (requires approx. 18K), 6502 (approx. 26K), 8080 (approx. 22K). Total memory requirement must allow for buffer areas.

For more information on the HDBS, write or call us today!

HDBS can be used to extend any of the following programming languages under the indicated operating systems:

CP/M with CBASIC; Microsoft BASICs, FORTRAN or COBOL: InterSystem PASCAL/Z; Sorcim PASCAL/M; Micro Focus CIS COBOL; Digital Research PL/I
MVT/FAMOS with BASIC
OASIS with BASIC
TRSDOS and NEWDOS (Models I and II) with *Disk BASIC*
North Star DOS with North Star BASIC
Apple DOS and Applesoft BASIC
Machine Language interface available on all above systems.

Note: Because HDBS can be integrated with a wide range of languages and operating systems, it provides *uniform methods of data handling* across those many languages and systems.

HDBS/QRS. An interactive Report-Writer/Query-System with these features:

- may be customized for non-technical users.
- complex retrieval conditions may be specified.
- detailed reports can be quickly generated.
- wildcard and "match-one" string specifications included.

HDBS/SRS. This Schema Redesign System permits...

- renaming fields, record types, sets.
- adding new fields to existing record types.
- allocating additional pages to an existing data base.

Ordering information (applicable to Z80, 8080, and 6502 versions):

HDBS	$300.00
HDBS/QRS	300.00
HDBS/SRS	150.00
HDBS/QRS/SRS Package	675.00
HDBS expansion to MDBS*	650.00
HDBS Manual	35.00
QRS Manual	5.00
SRS Manual	5.00
System Specific Manuals (each)	5.00
Guide to Data Base Management	10.00

*HDBS may be expanded at any time to full network data base management system: MDBS.DMS/MDBS.DDL

Within a given operating system, add $125.00 for each additional language selected.

For prices outside the U.S. and Canada, please ask for price lists.

Add $2.50 handling fee for non-cash order ($5.00 outside U.S.).

When ordering, specify intended use with...

1. North Star DOS and BASIC
2. CP/M - CBASIC
3. CP/M - Microsoft BASIC 4.XX
4. CP/M - Microsoft BASIC 5.XX
5. CP/M - Microsoft BASIC or FORTRAN Compiler
6. CP/M - Microsoft COBOL-80
7. CP/M - InterSystem PASCAL/Z
8. CP/M - Sorcim PASCAL/M
9. CP/M - Digital Research PL/I
10. CP/M - Micro Focus CIS COBOL
11. TRSDOS/NEWDOS and TRS Disk BASIC (Models I and II)
12. Apple DOS and Applesoft BASIC
13. MVT/FAMOS and BASIC
14. OASIS and OASIS BASIC
15. Machine Language Programs (Specify operating system.)

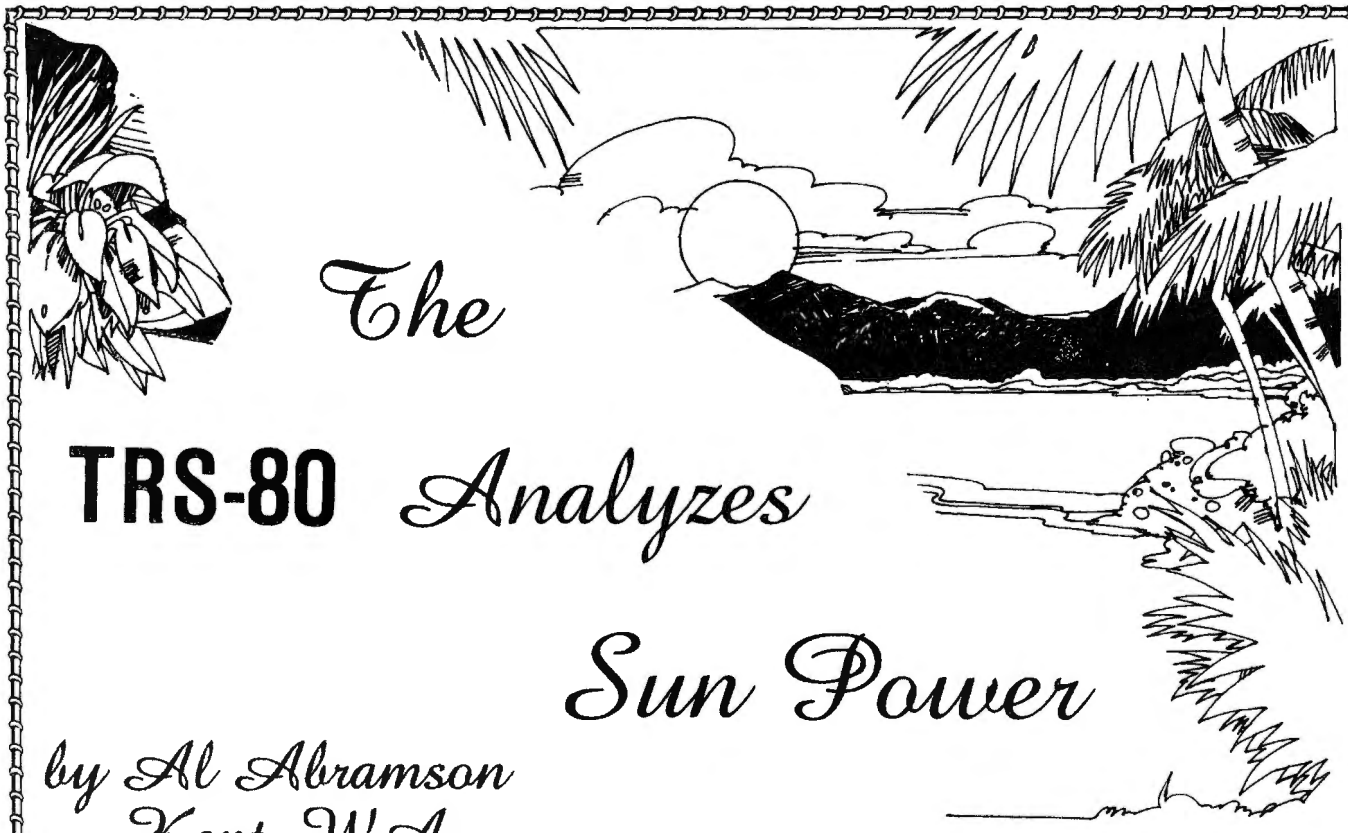
Indiana residents add 4%.

We accept Visa and Master Charge.

Micro Data Base Systems, Inc.

Box 248, Lafayette, Indiana 47902
317-742-7388 or 317-448-1616

Micro Data Base Systems, Inc.
Setting standards of excellence for data base software...worldwide.



The TRS-80 Analyzes Sun Power

by Al Abramson
Kent, WA

Everybody knows the TRS-80 offers fun and games, home budgets, small business programming, and in general, an escape from the hum-drum existence and monotony of repetitious TV programming. My own interest was small business (solar heating) analysis, and engineering without taking on the national debt in order to utilize the advantages of computer programming.

I soon learned that the 4K Level I I had bought would not work a mailing list, but it was too late for disappointment because I was caught up in the excitement of being able to do in one hour on the TRS-80 that which had formerly taken 14 hours with pencil and paper for engineering calculations. I had never "programmed" before. I moved up to a Level II 16K with screen printer.

There was a lot of talk about solar heating and "pay-back" periods. However, as we got assignments for three different large bank buildings to be solar heated, I noticed that the bankers were talking about "Return on Investment". Solar heating was generally thought to be the province of the "back to the good old days" people. But, sophisticated bankers were not "wood choppers".

"I found I could do in one hour that which had formerly taken 14 hours!"

I got to wondering, what is the difference between "pay-back period" and "return on investment"? I asked some of the bankers I had worked with. It turned out that the "pay-back period" was the length of time it took for the return on investment to equal the initial investment. I was told that it was calculated in exactly the same manner as any other real estate investment, but that it had the added advantage of returning a good "after-tax" savings in fuel costs that was growing at the rate of some 30% per anum. In addition, Uncle Sam began offering up to $4,000 income tax cash rebate to those who would put a renewable energy resource gathering facility (again, solar heat) on their dwellings.

I decided to see what all of this would add up to. I borrowed program segments from all over; laboriously worked up some segments that seemed to me to be "original"; used the return on investment factors that had been given to me by the bankers and "voilà", I had a simple solar heating investment analysis. It can be modified to analyze any investment under consideration.

But then solar heating is rapidly becoming a subject of common interest, so - why change the format? It might not make a great programmer out of you, but it might be fun to show your friends that you are an "expert" on Solar Heating Systems Investment Analysis! ●

For Model I, Level II, 16K and up.

```
10 CLEAR1000:CLS:GOTO40
20 IFINKEY$=""THEN20ELSECLS:RETURN
30 FORD=0TO127:SET(D,0):SET(D,47):NEXT:FORD=0TO47:SET(127,D):SE
   T(0,D):NEXT:RETURN
40 GOSUB30
50 PRINT@526,"SOLAR INVESTMENT VALUE ANALYSIS";:PRINT@720,"--PR
   ESS ENTER TO CONTINUE--";:GOSUB20
60 PRINT@133,"THIS PROGRAM WILL HELP ANALYZE THE VALUE OF AN IN
   VESTMENT
IN A SOLAR SYSTEM FOR A PRIVATE DWELLING.  IT WILL GO AT YOUR
SPEED.  PRESS ENTER TO CONTINUE.":PRINT@320,STRING$(63,131):GOS
   UB20
70 PRINT@198,"PLEASE DO NOT USE DOLLAR SIGNS ($), COMMAS (,), O
   R PERCENTSYMBOLS (%) IN ANSWER TO QUESTIONS.  FOR PERCENTAG
   E, PLEASE USE
THE DECIMAL EQUIVALENT (.XXXX).":GOSUB20
80 INPUT"WHAT IS THE PRICE OF THE SOLAR SYSTEM";AA
90 AB=AA*.4:IFAB=)4000AB=4000
100 PRINT"YOU SHOULD BE ENTITLED TO A TAX CREDIT OF $";AB;" LEA
    VING YOU
 WITH A NET INVESTMENT OF $";AA-AB
110 PRINT"THIS AMOUNTS TO A ";(AB/AA)*100;"% RETURN ON INVESTME
    NT THE 1ST YEAR.":GOSUB20:CLS
120 INPUT"WILL YOU (1) BE BORROWING THE MONEY, OR (2) INVESTING
     FROM
YOUR OWN HOLDINGS";Z
130 IFZ=1GOTO390
140 INPUT"IF YOU INVESTED THIS MONEY ELSEWHERE, WHAT RATE OF RE
    TURN WOULD YOU EXPECT (.XXXX)";A
150 INPUT"HOW MANY YEARS ARE WE LOOKING AT";Y
160 REM * A = ANNUAL RATE OF INCREASE *
170 REM * Y = YEARS OF PROJECTION PERIOD *
180 REM * A1 = VALUE OF 1 INCREASED BY 'A' RATE FOR 'Y' PERIOD
    *
190 REM * A2 = VALUE OF AMOUNT PER PERIOD *
200 A1=(1+A)↑Y:A2=(A1-1)/A
210 PRINT"IF YOU INVESTED THE MONEY ELSEWHERE, IN";Y;"YEARS IT
    WOULD BE WORTH $";A1*(AA-AB)
220 INPUT"WHAT IS THE PRESENT RATE OF INFLATION";I
230 BB=(1+A-I)↑Y
240 PRINT"THAT WOULD REDUCE YOUR NET RETURN TO";A-I;"IF YOU INV
    ESTED ELSEWHERE.  THE VALUE OF THE INVESTMENT AFTER";Y;"YEA
    RS WOULD BE $";BB*(AA-AB);"IN TODAY'S DOLLARS.
250 GOSUB20:CLS:A$=STRING$(64,92):PRINTA$:INPUT"AT WHAT RATE HA
    S REAL ESTATE BEEN INCREASING IN VALUE/YEAR OVER THE PAST F
    IVE YEARS (.XXXX)";RE
260 RR=RE-I:PRINT"IF WE SUBTRACT THE RATE OF INFLATION FROM THA
    T WE GET A NET RATE OF INCREASE OF";RR:PRINT"WHICH MEANS TH
    AT YOUR SOLAR INVESTMENT WOULD INCREASE TO $";AA*(1+RR)↑Y;"
    IN TODAY'S DOLLARS IN";Y;"YEARS.
270 PRINT"IF YOUR FIGURES ARE ACCURATE, AND THINGS CONTINUE AS
    THEY ARE:  "
280 PRINT"IN DOLLARS AT THAT TIME, YOUR SOLAR INVESTMENT WOULD
    BE WORTH";AA*(1+RE)↑Y
290 GOSUB20:CLS:B$=STRING$(64,"?"):PRINTB$:PRINT:INPUT"AT TODAY
    'S FUEL RATE, HOW MUCH WOULD YOU EXPECT TO SAVE EACH YEAR B
    Y INSTALLING THE SOLAR SYSTEM";S
300 INPUT"AT WHAT RATE DO YOU EXPECT FUEL COSTS TO INCREASE PER
     YEAR (.XXXX)";SR
310 TS=(1+SR)↑Y:TT=(TS-1)/SR:PRINT"THAT SAVINGS AT THE EXPECTED
     FUEL COST INCREASE RATE WOULD
 AMOUNT TO $";S*TT;"OVER";Y;"YEARS.
320 J=(AA*(1+RE)↑Y)-(AA-AB)+(S*TT):PRINT"IF YOU SOLD AT THE END
     OF";Y;"YEARS, YOU COULD EXPECT A PROFIT OF $";J;".":K=J/(A
    A-AB):PRINT"YOU WOULD BE GETTING BACK $";K;"FOR EACH DOLLAR
     YOU INVESTED.
330 PRINT"AND $";S*TT;"OF THAT WOULD AMOUNT TO COMPLETELY TAX-F
    REE DIVIDENDS.
340 PRINT"DO YOU KNOW ANY OTHER PLACE YOU CAN GET THAT KIND OF
    A RETURN
 WITH THE SECURITY OF REAL ESTATE?":GOSUB20
350 GOSUB30
360 PRINT@281,"SOLAR HEATING";:PRINT@599,"MAKES GOOD SENSE!";:G
    OSUB20
370 GOSUB30:PRINT@281,"THE DECISION-";:PRINT@602,"IS YOURS!!!";
    :GOSUB20
380 END
390 INPUT"WHAT INTEREST RATE DO YOU EXPECT TO PAY (.XXXX)";M
400 INPUT"WHAT IS THE PRESENT RATE OF INFLATION (.XXXX)";I
410 PRINT"YOUR NET INTEREST WILL BE ";M-I
420 INPUT"HOW MANY YEARS WILL YOU TAKE TO PAY THE LOAN";Y
430 PRINT"YOUR ACTUAL COST IN TODAY'S DOLLARS WILL BE $";(AA-AB
    )*(M-I)/2*Y+(AA-AB)
440 GOTO250
```

THE PROGRAM STORE (202) 337-4691

4200 Wisconsin Ave.NW PO Box 9609 Washington D.C. 20016

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

NEW!

microcomputer games®
from The Avalon Hill Game Company

Avalon Hill Game Company has just introduced their first five war and strategy games for the home computer. You play against the program. Each package includes instructions and software for the TRS-80, APPLE II and PET computers having 16k of memory.

Midway Compaign
The battle of Midway is recreated with you in control of the outnumbered and outranged U.S. Navy. The Japanese need air superiority to win. $14.95

B-1 Nuclear Bomber
Pilot your advanced bomber towards the target city in the Soviet Union. Avoid the MiG fighters and the surface-to-air missles. $14.95

North Atlantic Convoy Raider
This simulates the Bismark convoy raid of 1941. You control the British Home Fleet. $14.95

Nuke War
This nuclear confrontation pits you against the computer. You choose either massive espionage or military build-up. $14.95

Planet Miners
One to four players compete against the computer staking claims in the solar system. Watch out for sabotage and claim jumping. $14.95

INTERACTIVE FICTION

by Robert Lafore from Adventure Inter.
In these TRS-80 disk programs you influence the story by speaking with the characters. Each program sets a fictional scene. Then you start the dialogue with the other characters.

Six Micro Stories offers a good introduction. $14.95

Local Call For Death is a detective story in the style of Lord Peter Whimsey. $19.95

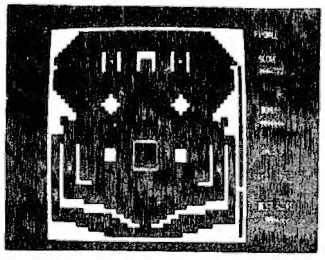
Two Heads of the Coin is a psychological mystery set in the London of Sherlock Holmes. The most challenging. $19.95

Electronic Handicapper BASKETBALL

by Sothen, Laurence & Gavenda
from Acorn Software Products
Basketball is the first in the Electronic Handicapper series from Acorn. It will introduce you to the benefits of predicting the winners of this season's basketball games. This two-tape package gives you power ratings to get you started. You keep the data tape informed of the current week's wins, losses and points. The program then calculates a winner and point spread for you to use. Last season it was used to predict 85% of the winners with a 64% accuracy with the point spread. 16k required. $99 [illegible] tape.

STOCK TRADER

from Galactic Software
This system is designed for the active "trader" and not the long term investor because the system is technically oriented. It tracks issues you select and reflects their performance against the overall market. There is also a comparison of the issue against itself to allow spotting "unusual" activity.
The initial data are from either the Standard and Poor Stock Guide or Value Line. The daily data of high, low, close and volume are input from the newspaper.
The program is intended to be a guide to indications and not as a sole recommendation.
Tape $89.00 Disk $99.00
Manual only $20.00



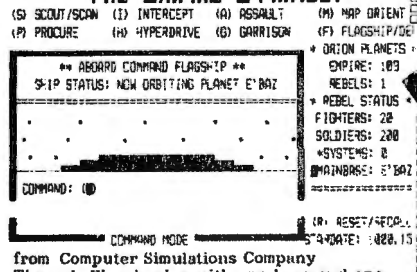
PINBALL

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

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 parameters including the speed, if you think it's too fast. Only a few heroes will be able to save earth.
Protected tape $14.95
Protected disk $14.95

THE EMPIRE STRIKES!



from Computer Simulations Company
The rebellion begins with one base and one warship. You take on fighters, conduct ground operations and secure planets, adding to the number of Rebel bases. Don't let any Empire Scoutcraft escape! You are the last hope. $14.95

SUPERSCRIPT

by Richard Wilkes from Acorn
"Scripsit" from Radio Shack is a great program, but it lacks some features. SuperScript adds features to your disk version of "Scripsit." Some of these features are:
You can get a directory or kill files from within SuperScript. TRSDOS or NEWDOS can read SuperScript files. You can insert text into unjustified lines during printout. For example, inserting a name after "Dear" and before the colon. For this purpose a lowercase driver is included.
On printers that can backspace, underlining and slashed zeroes (ø) are options. On Diablo and NEC printers one can superscript, subscript, underline, boldface and select 10/12 pitch.
The keyboard driver is changed to allow a correct key repeat which is faster than tapping on a key and which does not destroy the video display. The initial character sent to the printer is changed from a linefeed to a carriage return to empty the buffer. A required space may be specified when it is undesirable to place spaces between parts of text when justifying. From the keyboard you can also enter special characters such as brackets, braces and carets.
Serial and parallel drivers are included on the disk. You can customise these drivers for use with other types of letter quality printers. The serial drivers are included which use the ETX/ACK protocol for 1200 baud communications. Furthermore, printer drivers can be protected in high memory.
The "L" command used to load a file now requires a filespec to avoid destroying text buffer if the question mark is omitted from the "?L" command.
On disk for $29.95

STUCTURED BASIC TRANSLATOR

by Gene Bellinger from Acorn
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 efficient 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

COMING SOON to BALTIMORE

PROGRAM STORE (202) 337-4691

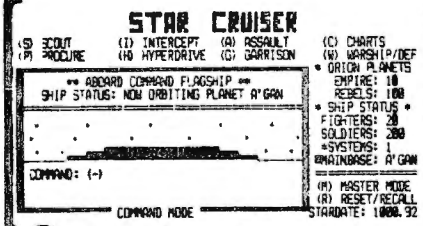
4200 Wisconsin Ave.NW PO Box 9609 Washington DC 20016

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

SUPER NOVA

by Bill Hogue from Big Five

This arcade game of shooting asteroids and alien ships is written in machine language and is the best we have seen on any computer. There are five different types of alien ships including the very deadly Flagship. You shoot from your ship, rotate it, use your thrust key to move, and in emergencies you go into hyperspace. Level 1 or 2 - $14.95



Star Cruiser

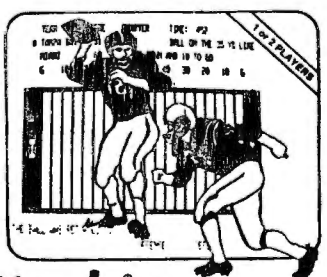
from Computer Simulations

You are on the side of the bad guys, the Empire. Program includes fighter combat, ground action, hyperdrive, garrisoning, retaking the 10 system capitals and avoiding the Rebel Star Fighter groups. The Star Cruiser is your last hope. $14.95

Checker King

by Michael Marks from Personal

Play checkers against the computer in one of eight levels of skill. Features include saving board positions, switching sides, going back a move and letting it play against itself. Also includes three checker puzzles. Level 1 or 2 - $19.95

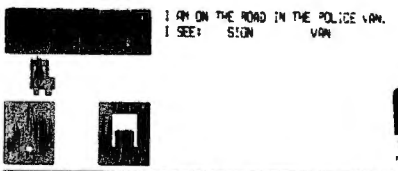


Pigskin

by John Laurence, Rick Sothen and Walter Gavenda from Acorn

In this football game you call the plays, watch the thirty-second clock, and get called for penalties if you aren't careful. There are eleven offensive and seven defensive plays. Features graphic display of field, the ball, and statistics on the scoreboard. You can play against the program, against a friend, or watch the play in the spectator mode.

Protected cassette $14.95
Protected diskette $20.95



SIGN SAYS 'NO PARKING'. Scene from QUEST

Adventures

with GRAPHICS & SOUND

by Robert Nicholas from Mad Hatter

These two adventures are similar to most others where you use two word sentences to explore. But these also show you the 'rooms' and have sound effects!

In QUEST you travel in search of fame, fortune, treasures, monsters and more. Three levels of play with a random set up of each game.

As the detective in SLEUTH you try to solve a murder. The victim, weapon and murderer will be different each game.

On cassette $14.95 each.
On disk $24.95 for both.

COMPUTER Bismark

from Strategic Simulations

The historic wargame that accurately simulates the epic battle between the German battleship and the British Home Fleet. Features a computer controlled mapboard of the North Atlantic, hidden movement, ship vrs. ship combat and shadowing, firepower and damage, historic setup of the ships, and both two-player and solitaire scenarios.

Cassette $49.95 Disk $59.95
Coming soon Computer Ambush

Microcosm I&II

from Basics & Beyond

Conbined these two packages contain fifty solid programs.

MICROCOSM I contains 16k-Memory Test, Atlantis, Biorhythms, Calories Food, Calories Ingredients, Challenge Match-em, Checkbook Balancing, Country Guess, Day Of The Week, Divisor Game, Driver, Eliminate, Factorials, Fakeout, Flowering Houseplants, Home Insurance, Instruction Test, Magic Squares, Match-em, Math Tables Drill, Metric Conversion, Morse Code, Mortgage, Music Transposition, News Photographer, Remainder, South Pole, Spelling Drill, Spelling Review, Wipe-out.

MICROCOSM II contains Algebraic Factoring, Base Conversion, Blank Slate, Decipher, Doodle Art 1, Electrical Use, Explore, Fish, Foliage Houseplants, Isolation, Perspective Racer, Planet Finder, Prefix Study and Quiz, Roman Numerals, Shooting Gallery, Spelling Bee, Stains, State Guess, Suffix Study and Quiz, Trivia Quiz.

Each available on two cassettes for $19.95

DISK SCOPE

from Instant

This diskette contains three programs to check out your disks. You can find the tracks and sectors where a file is stored with FILELOCK. With CDISK you can read the ASCII and hex representation of any sector. PASSWORD will give you the password for any file or any disk. $19.95

DISK EDITOR

from Instant

Access to any byte on a diskette is possible with the machine language utility. You just give either the track and sector or the filename in order to alter, add, delete, or view information.

You can also search a diskette for string of characters. And the data can be dumped to a printer. Both the 35 and 40 track versions are on the same diskette. $39.95

SuperSTEP

by Allen Gelder

Single-step/TRACE/Disassembler for TBUG. Variable speed TRACE mode lets you run any Z80 machine language program under total control. Includes intelligent RAM window, foreground background breakpointing, and more. TBUG required. $19.95

FORTRANslator

by Peter Charlton

You can literally translate BASIC programs into Fortran with this program which assumes that you know Disk BASIC and Fortran. It will not do a complete translation because of the differences between the languages. For example, subroutines, 'THEN,' string handling and disk operations are different. But for a programmer this utility is a good start. Requires Microsoft FORTRAN. $24.95 on disk.

FORTRAN available for $95.00

THE PROGRAM STORE
4200 Wisconsin Ave NW
P.O. Box 9609 Dept. KR4
Washington DC 20016

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.

Matrix Manipulation

For Model I, Level II, 16K and up.

In science, mathematics, business and many other fields that use number systems, it becomes necessary to be able to solve systems of linear equations. One system might take on the form of.

$$3x + 4y = 10$$
$$7x - 5y = 22$$

In the above example, there are two equations in two unknowns. (Unknowns refer to the variables x & y.) These kinds of equations are relatively simple to solve, and there are many methods to solve them by.

But what if there is a need to solve 35 equations in 35 unknowns?! It would take a human 3 to 4 hours to solve these.

In trying to maximize profits, or determine the total pressure needed to inflate 20 different balloons of 20 different sizes, or how much voltage is needed in order for 25 different circuits to carry a certain amount of current and (the list can go on forever), - systems of linear equations must be solved. Instead of paying an employee to sit down and solve these by pencil and paper, why not let a computer do it in 3 to 15 seconds??!

In order to solve these systems, it becomes necessary to work with matrices. And it becomes necessary to multiply matrices, and mainly to find the inverse of an m X n matrix. (m-rows and n-columns are usually designated by m X n.) Well, the TRS-80 has no matrix manipulation commands whatsoever. I admit that the other fine commands it has makes up for the difference, but why not simulate these missing commands with subroutines, like those in the following program?

The program will do the following things:

1. Add two matrices
2. Subtract two matrices
3. Transpose a matrix
4. Multiply a matrix by a constant
5. Multiply two matrices
6. Find the inverse of a matrix and/or solve for a solution of equations.

The first 4 items are straightforward, and probably will seldom be used except for large matrices. But the key to solving equations is in the last two. The program as it stands will fit into 32K of memory. If you have 16K, change the Dimension statements to 10 by 10, and if you have 48K these can be increased to 42 by 42. This means that if you have enough memory, you can solve a 42 by 42 matrix system, and it does this in less than 20 seconds!

The program will first display the menu, giving the above commands. Enter the number of your choice, and the computer will come back asking you the order (number of rows and columns) of your matrix. Some matrix algebra requires square matrices, but others can have a different number of rows than columns.

Then it will allow you to see the matrix that it just completed, giving you a chance to make sure that it was entered correctly. If you have a printer I would suggest that you output all matrices onto it. (The program will ask if you want output to go to the printer or not.) If your matrices are of a high order, then the output will not be very neat on the screen, since there are only 64 positions.

If you are solving for inverses the input works the same way, but on output you will notice that all of the matrices are double-precision. This is to allow for round-off errors inside the computer on the calculations needed for finding inverses. There is almost always a need for fractions in inverses, and using double-precision variables will take care of most of this. They are, by the way, returned to normal precision as soon as the inverse has been computed and printed.

Once the computer has solved for the inverse, it will prompt you with a question asking if you want to use that inverse to find the solutions to the equations. If you do, answer positively and enter the solution matrix when requested to. (Solution matrices are on the order of 1 by n.) It will then calculate all of the variables in the solution system and come back with output. (Here again you can request this to go to the printer.)

Here is a sample use:

if you had the equations 2x - y = 5 & x - 3y = 5 you would ask for the inverse function and enter 2, -1, 1, and -3 as the matrix for which you want the inverse. It will compute it and then enter the solution set of 5,5. It will come back with the solutions to the variables in the matrix form something like 2,-1. We know that x = 2 and y = -1. (2 * 2) - (-1) = 5 & 2 - (3 * -1) = 5.

Solving for x & y and 20 other variables could be done just as easily.

In the program run, the first matrix entered is called 'A', and the second if needed is called 'B'. This is standard notation in matrix algebra.

Of course every programmer will not have a use for the program. (It's an easy way for algebra students to get their homework done, so they can play Star Trek!) But those people in fields that use systems of equations will find this program invaluable, especially the inverse and multiplication functions, since the TRS-80 is not equipped with the matrix functions some computers have.

I hope that time and money can be saved by putting the program to good use.

PROGRAM PARTS

LINES:

10	initialize variables
20 - 130	menu
170 - 400	inverse function
400 - 560	solutions to equations if desired
570 - 670	add 2 matrices
680 - 830	subtract 2 matrices
840 - 930	transpose a matrix
940 - 1030	multiply by a constant
1040 - 1120	input of matrices
1130 - 1160	output of matrices
1170 - 1420	multiplication of two matrices

```
10 DIM A(20,20),B(20,20),T(20,20),I(20)
20 CLS
30 PRINT:PRINT          M A T R I X   M A N I P U L A T I O N"
40 PRINT"
50 PRINT:PRINT"WHAT WOULD YOU LIKE:"
60 PRINT"   (1) ADD MATRICES A & B
70 PRINT"   (2) SUBTRACT MATRICES A & B
80 PRINT"   (3) TRANSPOSE MATRIX A
90 PRINT"   (4) MULTIPLY MATRIX 'A' BY A CONSTANT
100 PRINT"  (5) MULTIPLY MATRICES A & B
110 PRINT"  (6) COMPUTE THE INVERSE OF MATRIX A
120 PRINT"      &/OR DETERMINE THE SOLUTION TO SYSTEM EQUATIONS
130 PRINT:PRINT"ENTER YOUR CHOICE:";
140 INPUT CH
150 ON CH GOSUB 570  ,680  ,840  ,940  ,1170 ,180
160 GOTO20
170 REM ** INVERSE OF A MATRIX
180 CLS:PRINT:PRINT"YOU HAVE ELECTED TO RECEIVE THE INVERSE OF
    A MATRIX":PRINT:INPUT"ENTER TO NUMBER OF ROWS AND COLUMNS I
    N A";N
190 DEFDBL A-Q,T,X
200 C=N:R=N
210 A$="A"
220 GOSUB 1040 :FORI=1TON : FORJ=1TON : A(I,J)=T(I,J):NEXTJ,I
230 FORC=1TON
240 FORI=1TO(N-1)
250 Q(I)=A(I,I+1)/A(I,I)
260 NEXTI
270 Q(N)=1.00000001/A(1,1)
280 FOR I=1TO (N-1)
290 FORJ=1TO(N-1)
300 A(I,J)=A(I+1,J+1) - A(I+1,1) * Q(J)
310 NEXT J
320 A(I,N) = -(A(I+1,1)) * Q(N)
330 NEXT I
340 FOR J =1TON
350 A(I,J) =Q(J)
360 NEXTJ
370 NEXTC
380 PRINT:PRINT:PRINT"THE INVERSE HAS BEEN COMPUTED.":FORI=1TON
    :FORJ=1TON:T(I,J)=A(I,J):NEXTJ,I:PRINT"THE RESULTS ARE READ
    Y TO BE PRINTED.":C=N:R=N:GOSUB1140
390 PRINT:PRINT"THE ABOVE IS THE INVERED MATRIX 'A'. WOULD YOU
    LIKE TO
1) RETURN TO THE MENU
2) ENTER A SOLUTION MATRIX TO SOLVE THE SYSTEM"
400 PRINT:INPUT"WHAT IS YOUR CHOICE";AN$:IFLEFT$(AN$,1)<>"1"THEN
    RETURN ELSEIFLEFT$(AN$,1)<>"2"THENGOTO390
410 PRINT"INPUT THE SOLUTION MATRIX"
420 X=0.00000001
430 FORI=1TON
440 PRINT "SOL. #";I;:INPUTQ(I)
450 NEXTI
460 FORI=1TON
470 FORJ=1TON
480 X=X+A(I,J)*Q(J)
490 NEXTJ
```

```
500 Q1(I)=X
510 X=0.00000001:NEXTI
520 PRINT:PRINT"THIS IS THE SOLUTION SET:"
530 FORI=1TON:PRINTQ1(I):NEXTI
540 DEFSNG A,T,Q,X
550 PRINT:INPUT"PRESS ENTER TO CONTINUE";A$
560 RETURN
570 REM****** REM ADD A & B  ***
580 CLS:PRINT:PRINT"YOU HAVE ELECTED TO ADD TWO MATRICES"
590 PRINT:PRINT"PLEASE BEGIN ENTERING THEM AS FOLLOWS:":PRINT:P
    RINT"HOW MANY ROWS IN YOUR MATRICES ":INPUT" ('A' & 'B' MUS
    T BE OF THE SAME ORDER)";R
600 PRINT:INPUT"HOW MANY COLUMNS IN 'A' & 'B'";C
610 A$="A":GOSUB1040
620 FORI=1TOR:FORJ=1TOC:A(I,J)=T(I,J):NEXTJ,I
630 A$="B":GOSUB1040
640 FORI=1TOR:FORJ=1TOC:B(I,J)=T(I,J):NEXTJ:NEXTI
650 FORI=1TOR:FORJ=1TOC:T(I,J)=A(I,J)+B(I,J):NEXTJ,I
660 PRINT:PRINT:PRINT:PRINT"'A' + 'B' HAS BEEN COMPUTED AND IS
    READY TO BE PRINTED":GOSUB1140 :PRINT:INPUT"PRESS ENTER TO
    CONTINUE";A$
670 RETURN
680 REM***** SUBTRACT A AND B
690 CLS:PRINT:PRINT"YOU HAVE ELECTED TO SUBTRACT TWO MATRICES"
700 PRINT:PRINT"PLEASE BEGIN ENTERING THEM AS FOLLOWS:":PRINT:P
    RINT"HOW MANY ROWS IN YOUR MATRICES ":INPUT" ('A' & 'B' MUS
    T BE OF THE SAME ORDER)";R
710 PRINT:INPUT"HOW MANY COLUMNS IN 'A' & 'B'";C
720 A$="A":GOSUB1040
730 FORI=1TOR:FORJ=1TOC:A(I,J)=T(I,J):NEXTJ,I
740 A$="B":GOSUB1040
750 FORI=1TOR:FORJ=1TOC:B(I,J)=T(I,J):NEXTJ,I
760 PRINT:PRINT"YOU JUST ENTERED 'A' & 'B' RESPECTIVELY":PRINT"
    WOULD YOU LIKE TO:
     1) SUBTRACT 'A' FROM 'B' OR
     2) SUBTRACT 'B' FROM 'A'"
770 INPUT"ENTER YOUR CHOICE 1 OR 2";AN$:IFAN$="2"THENGOTO810
780 IFAN$()"1"THEN PRINT:GOTO770
790 FORI=1TOR:FORJ=1TOC:T(I,J)=A(I,J)-B(I,J):NEXTJ,I
800 GOTO820
810 FORI=1TOR:FORJ=1TOC:T(I,J)=B(I,J)-A(I,J):NEXTJ,I
820 PRINT:PRINT"THE SUBTRACTION HAS BEEN DONE. ":PRINT"THE RESU
    LTS ARE READY TO BE PRINTED"
830 GOSUB1140 :PRINT:INPUT"PRESS ENTER TO CONTINUE";A$:RETURN
840 REM**** TRANSPOSE
850 CLS
860 PRINT"YOU HAVE ELECTED TO RECEIVE THE TRANSPOSE OF 'A'"
870 A$="A":INPUT"ENTER THE NUMBER OF ROWS IN 'A'";R
880 PRINT:INPUT"NOW ENTER THE NUMBER OF COLUMNS";C:PRINT
890 GOSUB1040
900 FORI=1TOR:FORJ=1TOC:A(I,J)=T(I,J):NEXTJ,I
910 FORI=1TOC:FORJ=1TOR:T(I,J)=A(J,I):NEXTJ:NEXTI
920 PRINT"THE TRANSPOSE OF 'A' HAS BEEN COMPUTED":PRINT"THE RES
    ULTS ARE READY TO BE PRINTED":PRINT
930 TE=C:C=R:R=TE:GOSUB1140 :PRINT:INPUT"PRESS ENTER TO CONTINU
    E";A$:RETURN
940 REM**REM ***** MULTIPLY BY A CONSTANT    **
950 CLS
960 PRINT:PRINT"YOU HAVE ELECTED TO MULTIPLY 'A' BY A CONSTANT"
    :A$="A"
970 PRINT:INPUT"ENTER THE NUMBER OF ROWS IN 'A'";R
980 INPUT"NOW, ENTER THE NUMBER OF COLUMNS";C:PRINT
990 GOSUB1040
1000 PRINT:INPUT"WHAT IS THE MULTIPLICATIVE CONSTANT";CN
1010 FORI=1TOR:FORJ=1TOC:T(I,J)=CN*T(I,J):NEXTJ,I
1020 PRINT:PRINT"THE MULTIPLICATION HAS BEEN DONE":PRINT"THE RE
    SULTS ARE READY TO BE PRINTED":PRINT
1030 GOSUB1140 :PRINT:INPUT"PRESS ENTER TO CONTINUE";A$:RETURN
1040 REM ** INPUT ROUTINE
1050 PRINT:PRINT"START ENTERING THE MATRIX '";A$;"' :"
1060 FORI=1TOR
1070 FORJ=1TOC
1080 PRINTI;",";J;
1090 INPUTT(I,J)
1100 NEXTJ
1110 NEXTI
1120 PRINT:INPUT"WOULD YOU LIKE TO SEE THE MATRIX THAT YOU JUST
     ENTERED";AN$:IFLEFT$(AN$,1)="N"THENRETURN
1130 REM ** OUTPUT ROUTINE **
1140 PRINT:INPUT"WOULD YOU LIKE THE OUTPUT TO GO TO THE PRINTER
    ";AN$:IFLEFT$(AN$,1)="Y"THEN P=1
1150 FORI=1TOR:FORJ=1TOC:IFP=1THENLPRINTT(I,J);"  ";:NEXTJ:PRIN
    T:NEXTI:ELSEPRINTT(I,J);"  ";:NEXTJ:PRINT:NEXTI
1160 RETURN
1170 REM *** MULTIPLICATION OF TWO MATRICES
1180 CLS
1190 PRINT:PRINT"YOU HAVE ELECTED TO MULTIPLY TWO MATRICES 'A'
    & 'B'"
1200 PRINT:INPUT"ENTER THE NUMBER OF ROWS IN 'A'";R
1210 PRINT:INPUT"NOW ENTER THE NUMBER OF COLUMNS IN 'A'";C
1220 A$="A"
1230 GOSUB1040
1240 FORI=1TOR:FORJ=1TOC:A(I,J)=T(I,J):NEXTJ,I
1250 INPUT"ENTER THE NUMBER OF ROWS IN 'B'";RB
1260 INPUT"NOW ENTER THE NUMBER OF COLUMNS IN 'B'";CB
1270 IFC=RBTHENGOTO1320
1280 PRINT"** ERROR-THE NUMBER OF COLUMNS IN 'A' MUST EQUAL":PR
    INT"THE NUMBER OF ROWS IN 'B' BEFORE THEY CAN BE MULTIPLIED
    .":PRINT"WOULD YOU LIKE TO
1) END FUNCTION AND RETURN TO MENU
2) RE-ENTER ROW & COLUMN LENGTHS FOR 'B'";
1290 PRINT:PRINT"ENTER YOUR CHOICE OF 1, OR 2";
1300 INPUTAN$:IFAN$="1"THEN RETURN ELSEIFAN$()"2"THEN1290
1310 GOTO 1250
1320 TR=R:TC=C:C=CB:R=RB:A$="B":GOSUB1040
1330 FORI=1TOR:FORJ=1TOC:B(I,J)=T(I,J):NEXTJ,I
1340 R=TR:C=TC
1350 FORI=1TOR
1360 FORJ=1TOCB
1370 T(I,J)=0
1380 FORK=1TOC
1390 T(I,J)=T(I,J)+A(I,K)*B(K,J)
1400 NEXTK,J,I
1410 C=CB:PRINT:PRINT"'A' * 'B' HAS BEEN COMPUTED AND IS READY
    TO BE PRINTED":PRINT
1420 GOSUB1140 :PRINT:INPUT"PRESS ENTER TO CONTINUE";A$:RETURN
```

by T R Dettmann
Associate Editor

THREE NEW DISK OPERATING SYSTEMS

One of the most asked questions when people move up to a disk system (aside from: What is a DOS?), is: "Which DOS should I use?". Why should there be a question - you get what you need with the computer when you buy it - right? Not quite.

TRS80 DOS provided with the system has improved over the revisions up to the current 2 3 version. Many of the complaints about TRSDOS have been corrected, but some people still find it somewhat restricting.

For true business applications, the end user should *never* see DOS. He should be able to boot right into the system. TRSDOS also lacks some of the capability to provide the kind of flexibility needed to get a variety of jobs done easily, without large amounts of machine language programming.

The fundamental purpose of a DOS is to handle all the funny little problems associated with talking to a real computer, and to make it possible for the user to get things done without tricky programming. TRSDOS gets you started in the right direction, but some of the other DOS's carry you further.

Since many of us have some pet capabilities we would like to see, nearly all of us have asked the question. "What's in the *other* DOS's?" Unfortunately, you can spend much money buying the DOS's before you find out that you don't need (or don't want) them all. Further, some of the DOS's now on the market have features which the average user will not be able to use efficiently Before you go out and spend up to $350 for all of them, let us run down their features for you

Let us clear this before we start: We talked to the various people who have DOS's for sale and were able to get a look at what they have But, we have only just received them, so what we will do is give a quick look. This will compare the features of the systems and give you an idea of what they offer based on our first look.

We will cover DOSPLUS 3.1 in more detail here because we have had it longer than the others. We will follow up with NEWDOS80 and VTOS 4.0 in the next issue. Remember that this is just a first look at NEWDOS80 and VTOS 4 0. We have hardly had time to digest them.

Lastly, you should be aware that for many people, TRSDOS is perfectly adequate It can do the job they want without the features some of the other systems have. Since it is free with your disk purchase, there is no reason to buy another unless (1) you *need* a special capability or, (2) you collect software instead of stamps. Either way, choose intelligently.

Now let's look at the DOS's one by one

NEWDOS80

NEWDOS80 by Apparat, Inc. of Denver, is the second operating system put together by them for the TRS-80. The first,

(Please turn the page)

What is a DOS?

DOS is computer jargon for *Disk Operating System*. This is a set of machine language programs which make it convenient for you to talk to your computer and especially to work with your disks.

Without a Disk Operating System, you would find that a diskette is just a flat, magnetic platter, and that you can write anything on it within the hardware limitations of your drives.

Sound good? Well, it wouldn't if you stopped to think that you would have to keep track of where everything is on the diskette. If you want to load some data, you would have to write a machine language program that would start the disk, find the data you wanted by track and location on the track, and then read it from the diskette.

All of this can be incredibly complicated if you want to do it yourself. For a single limited application it is rather easy, but to do it in general is really difficult. The Disk Operating System does it for you. It lets you find files simply by giving a name. Who cares where it is on the diskette? We let the DOS worry about that, we're more interested in our data.

It would take a whole book to explain the things a DOS does. In fact, in Computer Science departments, courses in how to write DOS's are graduate level instruction. What's really important to know is that a DOS insulates you from the realities of computer hardware, and makes it possible for you to worry about the problems you are trying to solve, rather than the mechanics of housekeeping.

VTOS 4.0	DOSPLUS	NEWDOS80	
●			ALLOC
●	●	●	APPEND
●	●	●	ATTRIB
●	●	●	AUTO
		●	BASIC2
●	●	●	BOOT
		●	BREAK
●	●		BUILD
●		●	CHAIN
	●		CLEAR
●	●	●	CLOCK
●	●	●	COPY
	●		CREATE
●	●	●	DATE
●	●	●	DEBUG
●	●		DEVICE
●	●	●	DIR
	●		DO
	●	●	DUMP
		●	FORMAT
	●		FORMS
	●	●	FREE
		●	HIMEM
		●	JKL
	●	●	KILL
●	●	●	LIB
●			LINK
●	●	●	LIST
●	●	●	LOAD
		●	MDBORT
		●	MDCOPY
		●	MDRET
●			MEMORY
	●		PAUSE
		●	PDRIVE
●		●	PRINT
●	●	●	PROT
●		●	PURGE
●	●	●	RENAME
●			RESET
●			ROUTE
	●		RS232
●			RUN
●			SET
●			SPOOL
●		●	SYSTEM
●	●	●	TIME
●	●	●	TRACE
●	●	●	VERIFY
●			XFER

A chart of Library Functions

This is a comparison of library functions. The presence or absence of a function in a particular system does not necessarily mean that function is missing. Some of the systems may include the function within a system utility. Or better yet, may remove the need for a function entirely by clever use of existing modules within the DOS.

commonly known as NEWDOS, has been widely hailed as the best all-around operating system for the TRS-80. I won't try to get into the middle of *that* argument, but I am already on record as being more than mildly impressed with the capabilities of NEWDOS.

As a general rule, NEWDOS set a standard for ease of use and flexibility that continues to influence the direction of software in this industry. NEWDOS80 pushes beyond this to offer more and better.

The biggest, most apparent improvement in NEWDOS80 is the introduction of a manual that can be read by someone other than the original programmer. NEWDOS80 comes with a three-ring binder manual which is a significant improvement over the old manual. It still has the provision that you *must* have a TRSDOS manual to get full benefit from it. In other words, they are not providing a complete reference manual, but they come much closer than before.

Let's look at some of the new library commands and utilities NEWDOS80 provides. We will cover them in more detail in the next issue, but for now, let's try and get some idea of what they accomplish.

Some of the new commands are:

BREAK - Enables/disables the BREAK key.

BOOT - Resets the computer.

CHAIN - Allows the user to create files of commands.

HIMEM - Allows the user to get the high limit of memory to protect drivers, etc.

JKL - prints the screen to the printer (as a command, i.e., CMD "JKL"

PDRIVE - allows the system to operate with a mixture of drives.

PURGE - selectively kills files from a disk.

SYSTEM - allows the user to change system options such as the inclusion of passwords, etc.

TRACE - displays the Z80's execution location every 1/2 second.

All the standard features of TRSDOS are included through some are modified to add new capabilities.

NEWDOS80 has the same utilities that have beome a standard with the NEWDOS system. These are:

DIRCHECK - does a check and listing of the directory.

EDTASM - a disk based version of the Radio Shack Editor-Assembler.

DISASSEM - a Z80 machine code disassembler.

LMOFFSET - allows you to load machine language programs with offsets in memory.

SUPERZAP - The disk inspection/modification program (now in machine language).

LEVEL1 - The Level I Basic system for Disk Operation.

LV1DSKSL - Lets you load and save Level I programs to disk.

As well as a few new ones:

LCDVR - A lower case driver for those with a lower case mod.

ASPOOL - An automatic spooling program to route printed output via disk (a free program not fully supported under NEWDOS80).

VTOS 4.0

Randy Cook (he's the person who wrote the original TRSDOS) formed his own company, called Virtual Technology Inc., to market his version of TRSDOS with the enhancements he felt *should* have been there in the first place.

People who have had some experience on large computers such as the IBM 360 series have been heard to compare VTOS 4.0 to IBM's JCL (that's *Job Control Language*). To some, Randy's system has always been as complex and hard to understand as JCL. It is also, without a doubt, the most flexible system around.

VTOS 4.0, like VTOS 3.0 before it, has the ability to route outputs to and from various devices simply by using a few simple

commands. You can freely re-direct output from screen to printer or to both. You can spool files to disk which will then print as time is available. Cataloging all of the "can's" with this system is a near impossibility. It is so flexible, that its limits have hardly been touched.

The problem with flexibility is that it introduces complexity. You can do so many things in so many ways that you lose sight of the problem you are trying to solve and get wrapped up in the complexities of the system. For the average user this may cause a problem.

Programmers putting together systems will find the flexibility useful. Some, who haven't discovered it yet, may find it absolutely necessary. Let's look at the extensions to TRSDOS:

ALLOC - Pre-allocates space on the disk for a file.
BOOT - causes the system to reboot from drive 0.
BUILD - allows you to create an ASCII file for use in Chaining, Patching, or with the special keyboard driver.
CHAIN - allows you to execute a file of commands automatically.
DEVICE - lists the logical devices and their assignments in the system.
FILTER - to establish an I/O 'filter' which can modify data on the way to an I/O device.
LINK - links I/O together from two devices, so they operate together (such as output to the screen and printer at the same time).
MEMORY - sets the high address of memory.
RESET - returns logical device assignments to normal.
ROUTE - re-routes an I/O device's output to another device (send everything that should go to the printer to a disk file or to the screen).
RUN - lets you load and execute a program.
SET - creates new logical I/O devices.
SPOOL - routes output to a device through a spooler.
SYSTEM - allows you to change the operating system configuration.
TRACE - displays the Z80 program counter.
XFER - transfers files from one disk to another on a single drive system, even if there is no system on either.

The system also comes with the following utilities and drivers:

VTCOMM - an advanced communications package that can use any of six logical I/O devices.
PR/DVR - a printer driver that allows the use of special printers with the system.
RS232/DVR - a general purpose driver for use of the RS232 connection in the expansion interface.
KSM/DVR - a special 'keyboard multiplication' feature which allows up to 26 phrases to be substituted for keys on the keyboard.

DOSPLUS 3.1

DOSPLUS, from Micro Systems Software in Hollywood, Florida, is still another entry in the DOS sweepstakes.

The user manual for DOSPLUS claims that it is the "most powerful, the easiest to use operating Disk Operating System on the market at present". Let's see what it does to live up to that claim.

The manual indicates that the following design features were incorporated in DOSPLUS:

1. Error Free System - it dosen't have a lot of errors to cause frustration in use and will not hang up the disk drives.

2.Increased speed - it is advertised as being the fastest system available.

3. No drivers are put in high memory, everything is in low memory. CMD can execute DOS commands from Basic and return to Basic.

4. Automatically knows when a lower case modification has been made and activates the lower case driver.

5. Adds repeating keyboard and keyboard debounce in the system.

6. The screen can be printed by pressing (SHIFT-CLEAR). This won't hang up, since if the printer is not ready, the command is ignored.

7. Variable length records for disk I/O.

The commands added to the system as extensions are:

BOOT - allows loading the system without affecting memory above 7000 Hex.
BUILD - allows creation of a file of DOS commands for later execution.
CLEAR - zeros user memory above 7000 Hex.
CREATE - pre-allocates file space on disk.
DEVICE - displays I/O devices and their driver addresses.
DO - executes a file created with BUILD.
FORMS - controls printer driver parameters.
PAUSE - stops program execution for operator input (during the execution of a file executed with DO).
RS232 - checks status of RS232 board and displays the switch settings.
TRACE - displays the Z80 program counter

It also adds the following utilities.

COPY1 - a single disk drive copy utility.
TRANSFER - copies all user files to another disk.
PURGE - kills unwanted files.
RESTORE - restores a file that was accidentally killed.
CLRFILE - zeros a file without returning the file space to the system.

Why bother to get DOSPLUS when you already have TRSDOS? Well, to answer that, I thought we would look into a few of the claims and see just what could be done to check them.

The most interesting claim to us was that this was the fastest DOS. I wrote a small benchmark program to check them out. The program looped to wait for the *ENTER* key, then opened a file and wrote out the numbers to 1000. It then closed the file, opened it again and read the file into memory, after which it printed "STOP" on the screen. After testing each DOS the file was killed, so that each had to create the file in the first place. Timing was done with a stopwatch, since the system clock during disk I/O is not reliable. TRSDOS 2.3 took 1 minute, 0 seconds; VTOS 4.0 took 1 minute 10 seconds; DOSPLUS took 1 minute, 0 seconds and NEWDOS80 took 1 minute and 5 seconds

While this may not have been what was intended, I didn't feel that the DOSPLUS really had any speed advantage. By the way, all of the DOS's were in an "as delivered" state, no modifications were made to any of them.

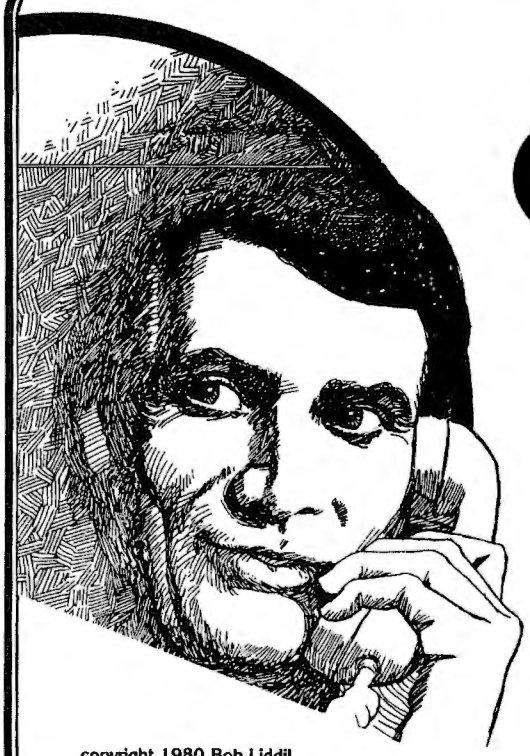
Trying out the DOS commands though, was much more satisfactory. Each of the commands listed earlier was tried, and they all worked as advertised The system in use booted twice, once into DOS and once into Level II Basic during the exercise, but I can't definitely pin that on either a hardware or software error.

What did impress me about the added commands was their similarity to the Model II DOS commands It looks as though the best of the Model II features have been implemented in DOSPLUS. This gets around many of the problems we have with multiple commands, etc.

I wasn't able to force any errors with the standard DOS commands This supports the claim to error free operation. I couldn't get the lower case driver to work with my keyboard, but once again, I should point out that my keyboard lower case modification is non-standard (non-Radio Shack), but both Scripsit and Pencil work with it.

I found the utilities more useful; PURGE and RESTORE appeared to be really useful and definite assets for real programming.

In the final analysis, my only real point about DOSPLUS is that the "PLUS" portion has some interesting new commands and utilities. In particular, some commands from the Model II appear to be implemented, including the "Free Space Map" If any of these commands fill a need for you, then this system would surely be worth considering ●



copyright 1980 Bob Liddil

CAPTAIN 80™

Bob Liddil

Here's Captain 80 in his Software Secret Agent disguise, boarding a 747, westbound for sunny California. Dressed in a natty Scotland Yard trenchcoat and teardrop mirror shades, I blend well with the business men and hijackers who make up the passenger list for this afternoon's flight.

While we are in flight would be a good time to reduce the stack of software that had piled up on my desk since my recent transition from the other magazine.

To market a program to the Boys from Fort Worth (TANDY), is a dream come true for a TRS-80 programmer. In the case of Leo Christopherson, whose talent and scope seem limitless, the big score came in the form of DANCING DEMON, a fine and entertaining little $9.95 number now available in every Radio Shack in the country. The animation techniques that earned Leo the respect and admiration of programmers everywhere, have never been more entertaining than in DEMON.

The end user is first treated to a light and lively little tapdance, complete with music, in two routines that demonstrate what the Dancing Demon can do. Then, when bows are done and the menu returns, we are invited to compose a song and dance ditty of our own. THAT'S RIGHT! Control of the graphics and sound routines are given over to the consumer.

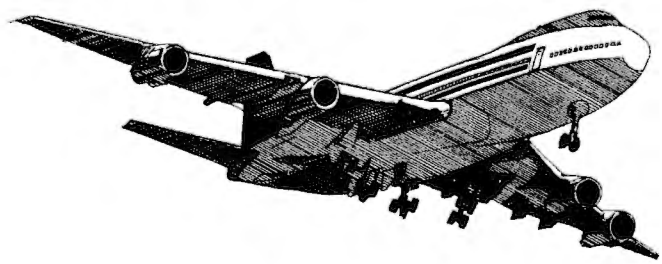
the arrival of the Master Pirate and his four hundred sixty ships lead to a thirty minute cliff hanger.....

Who can find fault with Christopherson's magic? Not me. Radio Shack, if you keep bringing out stuff like this, you'll have me back collecting my free battery and checking out what's new on the software shelf.

The next program has been around for a while, but, like watermelon wine, seems to get sweeter with each partaking. I am refering to TAIPAN, by Cybernautics, and promoted by The Software Exchange, Milford, NH.

Any economics simulation runs the risk of either being or becoming dry. Not possible with TAIPAN. Set in the 1800s as a China trade simulator, TAIPAN is startlingly realistic. It is at once easy and impossible to play.

You begin the game with a small amount of cash and a huge debt to Elder Brother Wu, the moneylender,



plus an extortion demand, (er, excuse me), donation request to a Master Pirate, on behalf of the Temple of the Sea Goddess. Once the financial formalities are over, the new China Trader may use his meager reserve to buy one of four stock commodities, General Cargo, Arms, Silk or Opium. Needless to say, many small transactions and no small amount of luck are required to move from Arms and General Cargo to the more lucrative Silk and Opium trade.

In my opinion, there is no better way to test "educational programs", than to turn them over to a kid and stand back. Scotty, a thirteen year old Little League catcher, Shannon, an eleven year old frisbee tosser, and Jason, his seven year old sidekick served as reviewers for TAIPAN. None of these boys had ever played computer before.

Starting with only a few hundred dollars, Scotty played for four hours straight, had amassed around eighty million and would likely have been there until the cock crowed if the Master Pirate hadn't arrived with four hundred sixty ships. Scotty, never one to count the odds, battled the pirates down to four ships, a thirty minute cliff hanger that ended with him being dispatched to Davy Jones' Locker, along with about two million dollars worth of Silk he'd bought for a fire sale in Shanghai.

Shannon and Jason played as a team, with Shannon, (who reads better), keypunching and Jason interjecting advice. This team's financial effort amounted to almost three million bucks before Elder Brother Wu dispatched five Chinese muggers to collect a much overdue debt. Undaunted, they bought Hong Kong silk and set out to regain their fortune. Scotty could have warned them about those pirates.

Listening to these youngsters happily using Math, business logic and deduction, and just generally enjoying the exercise of their minds, prompts me to recommend TAIPAN to anyone who has growing youngsters over seven or anybody who tires quickly of slushy programs that do the same thing over and over. If the inclusion of Opium as a commodity bothers you, change it to Jade or Morphine, etc. Watch out, though, TAIPAN's as addicting as ADVENTURE. Anybody need three kids? Please?

Another interesting program from The Software Exchange, one that is shared with Adventure International, Instant Software, plus several dozen smaller retailers, is called IRV.

IRV is a utility which intercepts the signals from the keyboard and reprograms then into a powerful set of programmer helpers. A TRS-80 shorthand is built in, with all the most used basic commands handily located in the (shift) key position. What sets IRV apart from the mundane is programability. The keyboard accepts reprogramming, up to 255 characters per key, every key on the board including BREAK, ENTER and the SPACE BAR. A cursor controlled video editor allows IRV to dip into a basic program to affect quick changes or block move and merge whole lines. Individual line renumbering is possible without changing the original line. Graphics on the screen may be assigned line numbers and saved to memory as packed strings. IRV occupies high memory and does not conflict with the loading or execution of any program which can exist without the 600 bytes or so that it uses. IRV retails for $25 tape and $29.95 disk.

"I put aside my reviewer's pen and prepare to do what Software Secret Agents do best....."

As the Jumbo Jet lands in Los Angeles I put aside my reviewer's pen and prepare to do what Software Secret Agents do best, that is, sneak around to computer shops and check out what they're doing.

I visited Hobby World's new Northridge California store. Posing as a salesman, I got a good look at the behind the scenes Hobby World. They had one whole, huge room devoted to computer classes. The classes feature the Atari, but my guide left the impression that TRS-80 is spoken energetically at HWC. The retail software rack was well stocked and there was a system set up and running. The interior was pleasant and the staff friendly. What caught my eye was the number of kidniks scattered across various computers, engaged in gaming or other computer activities. This situation seems to be patiently tolerated by the HWC retail staff, some of whom I observed taking time to answer the youngster's questions and drop words of encouragement. These kids are the computer generation and the cash customers of tomorrow. Only one other place that I have visited displays that kind of positive attitude toward unescorted kids. That was Computer City's Charlestown (Boston) store. And they have BOY'S CLUB right up the street. A tip of the Secret Agent's hat to these computer stores.

COMPUTER MART of El Toro California, who plans to open a SOFTWARE PLUS division, dealing in popular and not so well known lines of programs for the TRS-80 is inviting vendors large and small to submit packaged products for resale. There was no TRS-80 on hand to test software on the selling floor, but they're new and that could change.

Computer Components of Westminster California, who has always maintained a healthy supply of TRS-80 stuff, is called COMPUTER WORLD now, and is also looking to upgrade their supply of good software. Addresses and phone numbers for these guys can be obtained from an Orange County (California) phone book.

Other California shops visited seemed too busy to talk about the 80 or were not inclined .to deal in software for the 300,000 plus computerists who own 80s. But that's OK, we'll patronize those shops who treat us like something other than orphan stepchildren. And those merchants will prosper.

Almost before it began, my trip to California is over and I once more board a 747, this time eastbound. Now my secret agent disguise is no longer needed. I can relax in my Captain 80 uniform. The Hostess for my section serves me a Coke and comments on the color contrast between the green cape and pink tights of my Software Super Hero suit.

We take off and I head back for New Hampshire, enjoying the movie the airplane has provided. I like this 747. It sure beats flying. ●

Add POWER to your TRS-80™

SOFTWARE by MiProg

XEDIT, a high powered compact disk based editor designed for the TRS-80™ Model I or II. Whether it is BASIC, ASSEMBLY, or FORTRAN, XEDIT is packed full of commands needed by programmers who are serious about their work. Here are just a few features:

- Edits most file formats
- Block text copy command
- Locate, Delete, and Change with windows
- Inserts and maps up to five input files
- Upper/lower case compatible
- Operates with or without line numbers
- Rapid access disk cache
- Recovers from most DOS errors
- Fast file entry point map
- Change text command for any number of occurrences
- DOS Directory and Kill commands
- Line printer paging with adjustable forms
- Sophisticated reprinting line editor, handles line feeds
- Disk BASIC, Disk EDTASM, and EDIT-80 format compatible
- Display status command, includes free memory, current pointer printer forms, number of input files, output filename and format.

XEDIT will handle files of any size up to 2.7 Megabytes or 10K lines in length. Comes complete with instructions covering operation, externals, and file formats.

Model I (32K single disk system)
- Formatted diskette $44.95
- Cassette tape. $39.95

Model II
- Formatted diskette $89.95

Model III (32K single disk system)
- Formatted diskette $79.95
- Cassette tape. $75.95

ASM/CMD, a disk based assembler which generates object code to disk or tape (disk only on Model II). 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 operating manual.

Model I (16K single disk system)
- Formatted diskette $34.95
- Cassette tape. $29.95

Model II
- Formatted diskette $59.95

Model III (32K single disk system)
- Formatted diskette $49.95
- Cassette tape. $45.95

PACK/CMD removes spaces from text files generated by XEDIT, and EDIT-80 to reduce file lengths by 5 to 40 percent. PACK will also strip comment fields and line numbers for additional space savings. Text can be masked for upper case only. Does not destroy compatibility of assembly and FORTRAN source files. Comes complete with instructions.

Model I (16K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

Model II
- Formatted diskette $19.95

Model III (32K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

Special package, XEDIT, ASM, and PACK.

Model I
- Formatted diskette $79.95
- Cassette tape. $74.95

Model II
- Formatted diskette $149.95

Model III
- Formatted diskette $129.95
- Cassette tape. $125.95

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.

Model I (16K disk system)
- Formatted diskette $19.95
- Cassette tape. $15.95

CALL/CMD extends and improves the TRSDOS™ AUTO function. Can be enabled and disabled by prompts, and through keyboard, resident program, or the call file.

Model I (16K single disk system)
- Formatted diskette $19.95
- Cassette tape. $15.95

TANDON/CMD improves TRSDOS™ by allowing higher step rate, extending access to 40 tracks for the new Tandon disk drives. Also fixes the break key problem.

Model I (16K single disk system)
- Formatted diskette $14.95
- Cassette tape. . . . . $ 9.95

DEXER/CMD, a disk exerciser emulator program designed to speed repair of any TRS-80™ compatible disk drive. DEXER eliminates the need for the Shugart SA809 test fixture and decreases repair time with easy to use commands and on screen display of required set up data. DEXER was written specifically for the repair technician and Shugart or Tandon disk drives. Shugart alignment diskette or equivalent and a 30Mhz oscilloscope required. One key commands allow easier adjustments necessary for Shugart alignment. DEXER is not for general disk testing and is recommended only for service personnel who have previous experience in disk drive repair.

Model I (16K single disk system)
- Formatted diskette $24.95
- Cassette tape. $19.95

Dip shunts for conversion and upgrades for the TRS-80™. Comes complete with instructions for A, D, E, and G level boards and new 2 chip level II.

Two dip shunts and instructions $1.00

Please send check or money order to:

MiProg
P. O. Box 27014
Minneapolis, MN
55427
612-574-1711

Minnesota residents add 4% sales tax. Outside continential U.S.A., add $3.50 postage and handling.

Larry S Panattoni

Panattoni's Panacea

Inside the Radio Shack Expansion Interface

The expansion interface provided by Radio Shack provides a variety of functions. While many of these functions may not be needed by several present-day TRS-80 owners (until they make further investments); some are desired and needed right now.

Figure 1 shows a block diagram of the Radio Shack expansion interface. The major sections are:

Memory Expansion - This is the single function most '80 owners associate with the expansion interface. This circuitry contains enough integrated circuit sockets to expand RAM to 48K.

Real Time Clock - This section provides a 25 millisecond pulse, used by the software to count seconds, minutes, hours, etc. Not truly a real time clock as one associates with the term - but still very useful.

Dual Cassette Outlets - Enables the use of either one of two cassettes at any moment. Truly improving a cassette based '80 setup. The more recent use of this circuitry is to provide stereo sound output.

Disk Controller - This portion would not be needed by '80 owners until they purchase at least one disk drive. This circuitry performs several control functions associated with the disk drives (up to four).

Parallel Line Printer - Here data is routed to an output port to be sent to the line printer. Sensing circuitry is also provided to tell if the printer is busy.

Screen Printer Bus - This bus is provided for driving a screen printer, as well as a multitude of other external devices.

Address Decoding - This portion determines with which section of the interface the computer is attempting to communicate. Hence, this decoding section is utilized in most all sections within the interface.

Power Supply - Finally, this section provides three voltages for use within the interface: +5 volts, - 5 volts and +12 volts.

Starting with this issue, I will provide you with an inside look at the Radio Shack interface. Taking one section at a time, we will give an in-depth circuit description, enabling you to understand the reason for each component. Then we will present a construction project, enabling you to build a similar stand-alone unit of each section, including added features where appropriate.

Even though this series may be interrupted occasionally in the future to provide you with 'Special Construction Projects' who's time have come; continuation of this interface series will be promptly resumed in the succeeding issue.

Line Printer Interface

The first section I will take apart in discussion, will be the 'Line Printer Interface' section, which is detailed in **Figure #2**. Here an ASCII parallel line printer output port is achieved. This is a widely used industry standard, and has proven to be quite reliable. It is, of course, directly compatible with most Centronics printers, as well as all R.S. printers and several other makes. It is best, before purchasing other line printers, to inquire as to whether the printers are directly compatible with the S-80 or need hardware or even software modifications.

Status Lines

When the S-80 is to send data to the line printer it first reads the status of the four input lines (D4 thru D7), via the four buffers of Z46 in **Figure #2**. A High on D7 indicates the printer is busy, informing the computer to wait until the printer completes its present operation; at which time D7 will be changed to a Low . Depressing the front panel 'Print' switch to the off position on a Centronics Model 779 line printer will also place a High on D7, creating a busy signal.

Status line D6, when High, indicates the printer is out of paper and unable to receive data from the computer. Upon refilling the paper supply, this D6 signal is .eturned to its normally Low condition.

Two more status lines are available. D5, Unit Select, and D4, Fault, both would normally indicate a printer busy condition with a Low, which is the reverse of D7 & D6. However D4 and D5 are not used in the R.S. line printers. Therefore resistors R33 & R34 are used in **Figure #2** to retain a High signal, indicating a Ready-to-Print status.

Data Output

Z44 and Z45 are latches which are used to hold each byte being sent to the line printer. Upon determining the printer is not busy and not out of paper, the computer places a byte to be sent to the printer on the data lines D0 thru D7. A low pulse is then sent to the 'WRITE' lead, which is connected to the 'Clock' pins of Z44 and Z45. This pulse loads the byte into the latches for output to the printer, at which time the S-80 is free to go about other business.

The address necessary to produce a Low pulse on the 'Write' or 'Read' lines of the line printer circuitry is -- 37E8 (HEX). The address decoding circuitry of **Figure #1** monitors all addresses on the address bus and activates the proper section within the Expansion Interface who's address has been indicated.

The 'Read' and 'Write' lines of **Figure #2** originate in the decoding circuitry of **Figure #1**. A 'Read' operation to address 37E8 (HEX) reads the status bits D4 thru D7 (D0 thru D3 are not used). And a 'Write' operation to address 37E8 (HEX) loads the latches with the data to be sent to the printer.

Printer Synchronization

The 'Write' pulse mentioned above is also routed to the B input of Z29. Z29 is a 'Monostable Multivibrator', which produces a 1.5 Microsecond Low output pulse on pin 4. This 1.5 Microsecond pulse is determined by R16 & C64. The other pins of Z29, which are designated with the letter 'X' are not used in this line printer circuitry, but are used in the Disk Control section and will be discussed at a later time.

Data Strobe, as this 1.5 Microsecond pulse is called, provides the synchronization between the computer and the line printer. On the rising edge of this pulse, data is transfered from the output latches (Z44 & Z45) to the data

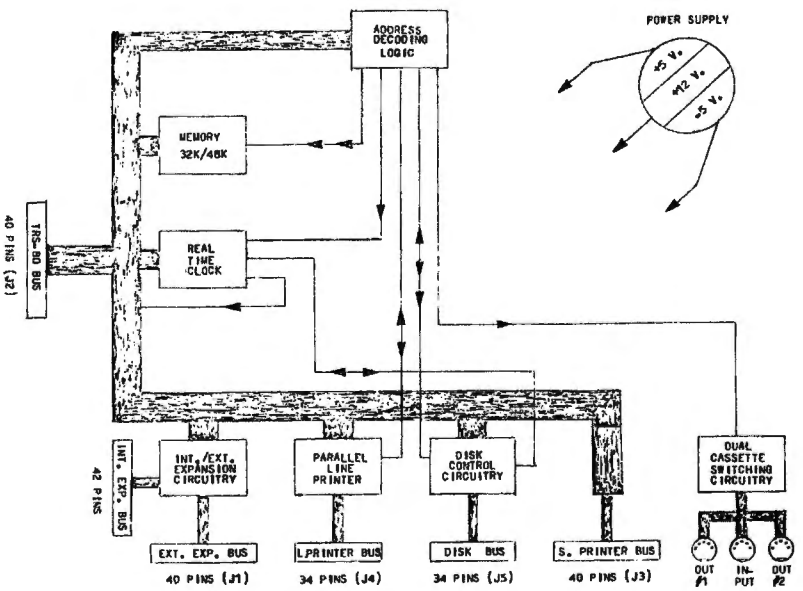


Figure 1
Block Diagram of the Radio Shack expansion interface.

buffers (RAM) within the printer itself.

The R.S. line printers are capable of storing up to 132 characters in their internal data buffers. Upon receiving either the 132 characters or a carriage return, the line printer places a High on the busy status line (D7), informing the computer to wait until it has printed the contents of its buffer, so as to enable it to use the buffer again for the next 132 characters.

Hand-Shaking

The R.S. line printer control circuitry utilizes a form of Hand-Shaking. Hand-Shaking means symbolically that the computer raises its hand when it has a byte ready for the printer to accept. And the computer maintains this byte until the printer returns a hand of its own, acknowledging receipt of this byte, and indicating it's ready for the next byte. Hence the term 'Hand-Shaking'.

The data strobe of Z29 pin 1 in **Figure #2**, is the computer's indication a byte is ready for the printer to accept. However, instead of waiting for a signal from the printer that it's ready for the next byte, the computer waits only for the 1.5 Microsecond duration of the data strobe pulse; then the ROM software (which I have not yet deciphered), similar to the source code above, monitors the condition of the status lines in **Figure #2**. If they do not indicate a 'Busy' or an 'Out of Paper' condition, then the next byte is output to the printer, along with another data strobe pulse.

Software

The R.S. printers, such as the Centronics, have the status signals D6 (out of paper) & D7 (busy) OR'd together, which will cause a busy signal if either goes High. So the software necessary to check the status of the line printers need only check D7. The source code listed below, can be used as a subroutine in your machine language programs, for outputting data to the line printer.

(before entering this subroutine, Register C must contain the Character to be printed.)

```
TSTPTR LD  A,(E7E8H) ;get line printer status
       AND   128     ;test D7
       JP  NZ,TSTPTR ;if not Low, go back & recheck
                      printer status
       LD  A,C       ;get character in Register A
       LD  (37E8H),A ;send character to printer
       RET           ;return back to main program
```

R.S. Cable Mod. -- Buffered Cable

Figure #2 shows +5 Volts on pin 37 of the Expansion Interface bus (J2), which connects to the keyboard through a cable. On earlier units this was a straight cable, and pin 37, along with pins 8, 29, and 39 were grounded. However, due to feedback glitches causing undesired resets, while using the disk drives, the cable was replaced with a buffered cable. And at that time the ground on pin 37 of the Interface bus (J2) was removed and replaced with +5 Volts, which is used to provide VCC to the IC's in the buffered cable.

Warning stickers were at that time placed on the Interface near connector J2, stating the Interface is to be used only with the buffered cable. The reason being that pin 37 will now short out the power supply, should it be connected using the old "Unbuffered" cable.

Next Issue

Next issue I will present a modified version of **Figure #2**, enabling you to construct a stand-alone line printer interfacing cable; which will drive a printer without the need of purchasing an Expansion Interface unit. Also included will be a couple of extra features, which you may find useful. ●

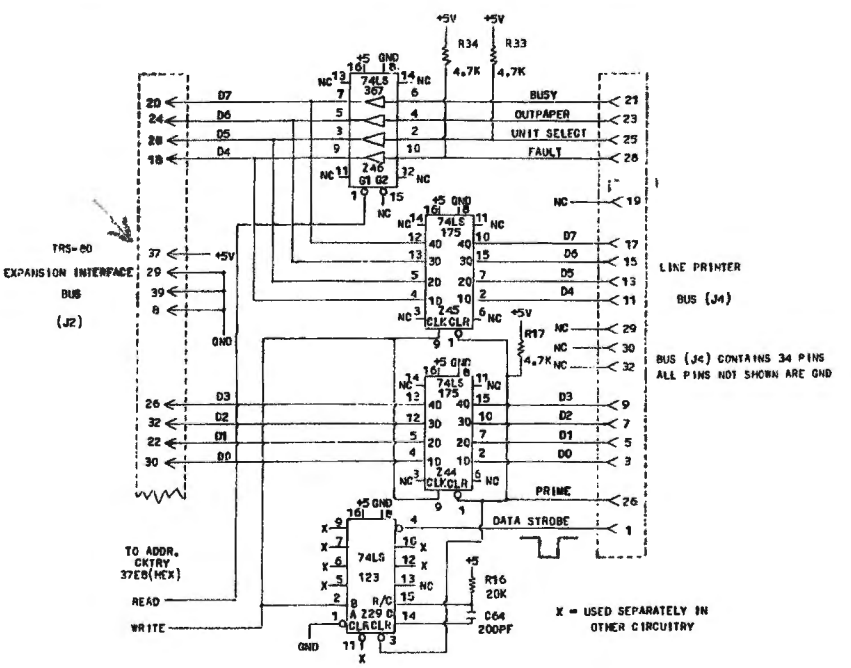


Figure 2
Line Printer circuitry within the Radio Shack expansion interface.

DOSPLUS 3.0

FREE

with purchase of **Dosplus** or **Master Diskette Directory.**

One of the below:

STAR TREK
BREAKOUT
BLOCKADE

Machine Language Games With Live Action Sound

MICRO SYSTEMS SOFTWARE INC.
5846 Funston Street, Hollywood, FL 33023
(305) 983-3390

NAME _____
ADDRESS _____
STATE _____ ZIP _____
PHONE _____
ATTENTION! _____
ACCOUNT # _____
MC □ VISA □ EXP. DATE _____
MASTER DISKETTE DIRECTORY
QUANTITY _____
DOSPLUS 3.0
QUANTITY _____

Faster And More Efficient Than:

- TRSDOS
- NEWDOS +
- VTOS

DOSPLUS Uses less memory and still offers all the commands of the other major operating systems, AND some innovative and useful features of our own design.
DOSPLUS is now available for the TRS - 80* Model I, II, and III

LIBRARY OF COMMANDS:

APPEND	ATTRIB	AUTO	BOOT	BUILD
CLEAR	CLOCK	COPY	CREATE	DATE
DEBUG	DEVICE	DIR	DO	DUMP
FORMS	FREE	KILL	LIB	LIST
LOAD	PAUSE	PROT	RENAME	RS232
TIME	TRACE	VERIFY		

BUILT - IN FEATURES:

AUTOMATIC LOWERCASE
SCREEN PRINTER
KEYBOARD DEBOUNCE
REPEATING KEYBOARD

UTILITIES:

COPY 1
RESTORE (DEAD FILES)
DISK DUMP/CMD
FORMAT
TRANSFER
PURGE
CLEARALL
BACK-UP

NEW FROM MICRO-SYSTEMS!!!

Master Diskette Director Ver 1.0 -

This program will read the directories of up to 320 diskettes or 5000 files! Insert your diskette in the drive, press enter, and it works automatically to read, store, and categorize your files. No more looking through box after box of diskettes looking for those favorite programs that are "Around Here Somewhere". You can get a listing of all the files on all the diskettes. In addition, you can list them by file extension, diskette number, or program category. With master directory, you need never again worry about the frustration of not being able to locate a particular program. Master directory will search for a particular file name and give you every occurence of that file, its size, and the diskette number that contains it.

A totally self contained, machine language program on a self-booting diskette; master directory will work equally well on single or multiple drive systems.

Dosplus - $99.95 Master Directory - $29.95

Please Contact:
Micro Systems Software Inc.
5846 Funston Street
Hollywood, Florida 33023
1 (305) 983-3390

Coming Soon: Dosplus 4.0 for 10 - Megabyte TRS - 80s*

*TRS - 80 is a trademark of Tandy Corp

TRS-80 TRICKS

by Ronald H Bobo
St Louis, MO

At the Amateur Radio and Computer Hobbyist convention held in St. Louis in May 1980, one of the more interesting features was a talk by John Knoderer on programming tricks with the TRS-80.

John is the owner of COMPUTERVAN, a mobile programming and consulting service, as well as the COMP-U-TRS software store, both operating out of a shopping center in Florissant, MO.

With John's concurrence, I'd like to pass along some of the ideas which were discussed.

Some methods for speeding up program execution include:

DEFINT - This alone can speed up a program as much as 20% if you have many integer variables. As a bonus, there is the possibility of being able to write longer programs, since two bytes will be saved for each variable defined as an integer.

DEFSTR - Defining variables as strings will afford some increase in speed, since less information has to be checked by BASIC. In addition, a little memory space is gained by not having to type the "$" on each variable.

A memory-saving trick not documented in your Level II book concerns printing strings. In some situations, it is possible to omit about 75 to 80% of semicolons. When switching from a literal to a variable and back, semicolons are not necessary. (A literal is anything enclosed in quotes.) If printing two strings together, as A$ followed by B$, the "$" acts as a delimiter, making the semicolon unnecessary. Semicolons may also be dispensed within PRINT TAB operations.

Another increase in speed of execution may be achieved by packing program lines. Use multiple statements on a line whenever possible, and remove all REMark statements. There are utility programs on the market which will do this for you after your program is written. In the event that explanations must be present in the listing, try putting all REMarks at the end of the program. This way, the program is not slowed down reading these lines. Never put a REMark inside a loop!

Believe it or not, putting commonly used subroutines at the beginning of a program, rather than near the end, will save time. The reason? It takes less time to find a line number near the beginning of a program.

When putting graphics on the screen, the fastest method is to print strings of graphic characters. Contrary to popular belief, this is much faster than POKE graphics. Try this one:

```
10 A$=STRING$(255,191):
   B$=LEFT$(A$,3)
20 PRINT A$A$A$A$B$:POKE
   16383,191
```

In this example, one A$ will give almost four lines of solid white. Four of these will white out all but the last four spaces on the screen. B$ takes care of three of these and, since the last space cannot be printed to, a 191 is POKEd into 16383 to fill up the corner. This was timed with a stopwatch and painted the screen white in .63 seconds! Contrast that with SET graphics.

During discussion of program protection, John brought out that one of the best protection methods is to make documentation too expensive to copy. This is only feasible, of course, for a very complicated program which cannot be used without the written instructions.

Several methods may be used to bog down potential pirates, however. Methods not recommended are those which make it impossible to copy a tape or disk. When a person has shelled out good money for a program he should be able to either back it up or get a free replacement in case of failure of the tape or disk.

Security disks are particularly reprehensible. Not only is it aggravating to have to use one, but it seems that in a good percentage of cases the disk read fails, and the operation must be started over from the beginning.

All the following methods allow backups to be made to both tape and disk.

One idea, and it has been explained elsewhere, so I will not go into it too deeply, is to hide your name or birthday or some other personal info by manipulating variables buried within the program. If a case ever gets into court where someone is claiming your work as his own, it is a simple matter to RUN the program, print out a variable with this info in it, then sit back and listen to the defendant's lawyer try to explain how it happened.

In Radio Shack Disk Basic 2.2 (the latest version, with DOS 2.3) the BREAK key may be disabled by POKEing 16396,23. Hitting the BREAK will have absolutely no effect - the program keeps right on running. In Level II BASIC, this POKE has a little different effect. Hitting the BREAK key stops the program, but the keyboard locks up. With NEWDOS, hitting BREAK will reboot the system. Of course, the big drawback here is that the program may still be LISTed if this is done before the RUN.

One way around this limitation, which may be satisfactory in some situations, is to make the whole program unlistable. In Level II BASIC, this is done by POKEing 17131,254 and 17132,255. Do this in the Command Mode, not in the program itself. In Disk BASIC the numbers are POKE 27174,254 and POKE 27175,255. For those using NEWDOS, try POKE 26812,254 and POKE 26813,255. Now type LIST and watch what happens.

Nothing happens, that's what, except that you get a READY prompt. RUN the program, and you'll never know the difference. It will run perfectly, and may be saved to tape or disk. Hope you saved an unmodified copy for yourself, in case you someday want to make a change in the program.

What is occurring here is that you're POKEing in a line number of 65534 for the first line - but 65529 is the largest line number allowed. Never mind - the

(Please turn to Page 44)

TRS-80 LEVEL II

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

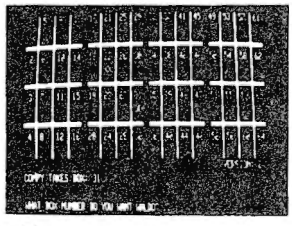
Are you an Othello freak? Flip Disc is a program which will turn your computer into an excellent opponent Three different skill levels, (good, expert, and genius), provide an introduction for the novice and continuing interest for the experienced player

• 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

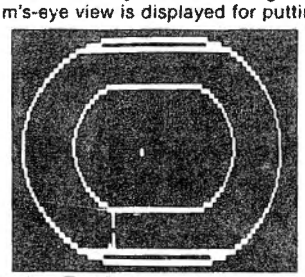
In this real time game, you are pursued around the game board by an evil-looking snake Variations of play include two different speeds and hyper-jumps which randomly relocate you on the board Looking for an escape? Try Evasion

• 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

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!

WRITE FOR...

FREE

SOFTWARE CATALOG

creative computing

computer thinks it's OK, except it can't list a line number like this. Everything added to the program from this point on will, as far as the computer is concerned, goes before the first line number, so nothing gets listed.

Like some other subterfuges, there is a catch here. This trick will only work in a program with no backward GOTO's. In other words, all your jumps must be forward in the program, or it won't run right. If you have a program, though, where everything proceeds in a straight line, this could be helpful in protecting your programming secrets.

The following is useful not so much for protection as to make it hard to solve a problem. John Knoderer has used it to good advantage in his version of "Petals Around the Rose" to keep players from discovering the Secret of the Rose unfairly by looking at the listing. What you do here is to write long, multiple statement lines in your program, taking up almost four lines of video, and put the statement you want to hide at the end of the line. Once this is done, go into the EDIT mode and insert PRINTs into the line, using the "?" for PRINT. You want enough so that the "?'s" are expanded into PRINT's. Part of your fourth line will disappear off the right side. You have just made your line 255 characters long. When the program is LISTed, the last part of the line will be invisible and if an attempt is made to EDIT the line, part of it will be lost.

One nifty trick that may be useful if you have some private programming trick you want to keep to yourself involves making program lines unlistable. Of course, you have the poke routine which will cause the whole program not to list, but with this trick you can unlist selected lines. Imagine someone going through your program line by line trying to find how you did something, and the key lines are just not there! Much hairpulling will result, you may be assured.

This is very simple to execute, but may be a little difficult to visualize when reading about it. Briefly, what you want to do is to type in the line to be hidden in the usual manner. Put a REM statement at the end of the line, preceded by a colon. Now get into the EDIT mode, hit the "X" key to get to the end of the line in a hurry, and type one or more spaces. Now type a shifted up-arrow to get out of the insert mode and backspace one or more spaces and type "C" for Change. Next, type a shifted up-arrow again. This will shift the cursor up one line and, when you hit ENTER, the line will vanish. If your program line takes up more than one line on the video display, you will have to change enough backspaces to shifted up-arrows to blank out the whole line.

Now, when you LIST the program, the doctored lines will flash briefly on the screen, so quick as to be unreadable before the next undoctored line appears.

There are two small drawbacks to this method. One, the lines may be edited. This requires knowing the line number, however, and you can make this difficult to learn. Two, the lines will show up if the program is LLISTed on a printer, but then everyone doesn't have a printer.

To hide it from the person who does have a printer or access to one, do this. When you add the spaces to the end of the line, make sure you add at least three. Get out of the insert mode as before, and backspace three times. Now type 3C to change three characters, and hit the shifted up-arrow as before. Of course, if your program line is more than one line on the video, you will have to put in extra spaces to compensate, so the extra upward line feeds may be inserted. Now, type a shift down-arrow and while still holding both keys down, type "L". This is the fiendish part. The shifted down-arrow will give you an ordinary line feed, but typing "L" while these two keys are down gives, not a capital "L", but ASCII 12 which is a control code for Top of Form when a lineprinter is connected. Now, when he tries to LLIST, he may or may not get a line of program or a partial line every now and then, but one thing is certain: He'll have one very high pile of paper if you've done this on every line!

Not perfect security by a long shot, but it can sure slow someone down. Also, if your key lines are scattered judiciously through the program, it may not even be suspected that this method has been used. Due to the very short time that a doctored line is on the screen, a single line may not even be noticed as it flashes.

Happy programming, and may these hints be of some assistance to you ●

View

(from the Top of the Stack)

The 13th in a series of tutorials on Machine Language programming.....

James W Crocker
Technical Editor

In our last installment, we were getting around to loading and storing data between our registers and memory. I mentioned that I would be devoting a special section to the stack pointer register, SP. Here we are.

The SP is one of the 4 genuine 16 bit registers. This means that we can load it with any number from 0 to FFFFH. This is an amazing coincidence, since this is the limit of the memory that can be accessed by the Z-80 directly. The SP register is another of those registers that is designed for indirect memory access only. Unlike the index registers, it is not accessed using a displacement. Instead, access is via two very special instructions: PUSH and POP.

The stack pointer (not too surprisingly) POINTS at a STACK. There are two kinds of stack: The last in first out, or LIFO; and the first in first out, or FIFO. The Z80 uses the LIFO version. The easiest way to visualize a LIFO stack is to go down to your local cafeteria or buffet style restaraunt, and take a good look at the devices they keep the plates in (if someone looks at you a little strangely, just explain that you work with computers. They will understand). These devices are basicly a hole with a plate on the bottom, held up with a spring. The spring keeps the top plate on the stack even with the top of the table, no matter how many plates are added or removed (within limits, of course). As each plate is removed from the stack, the one below moves up to be taken. Also, no matter how many plates added to the stack, the *LAST* plate IN is always the *FIRST* plate OUT.

Now we obviously don't have plates in our computers. We are dealing with computer data. Nor do we have memory moving up and down as we add or remove data from the stack. Instead, the SP simulates this moving up and down by AUTOMATICALLY incrementing or decrementing itself. Automatically is the key word here.

Let's say that you have need of the BC register pair, but it already holds the result of a previous computation which you also need. This is a job for the stack pointer! First, you must set aside some memory for that use (and no other). Then you must load the SP register with the highest address of that space, (this is usually done at the beginning of a program, and is only done once). Then, to save your BC pair, simply execute a PUSH BC instruction. The SP is decremented, the B register is loaded into the address pointed to by the SP. The SP is decremented again, and the C register is loaded into the new address pointed to by the SP. To clarify all this, assume that your program has reserved 100 bytes of memory starting at address F000H. Assume also that the SP has been loaded with the highest address of the reserved area, or F100H. Furthermore, assume that the BC pair holds the value 1234H. After the execution of a PUSH BC instruction, address F0FF will have 12 in it, address F0FE will have 34 in it, the SP will have F0FE in it, and (this is important) *the BC will still have 1234 in it.*

To get your value back, simply execute a POP BC instruction, this works exactly opposite of the PUSH instruction. First, the byte pointed to by the SP is loaded into the C register and the SP is incremented (it now points to address F0FFH). Then, the byte the SP is now pointing to is loaded into the B register and the SP is incremented again. Now the SP points to F100H, and the BC again hold 1234H. Note that unlike the PUSH instruction, the POP instruction destroys the previous contents of the register in question. But since that is the reason for a POP, don't let it bother you.

PUSH and POP are allowed for any register pair (except SP and PC), and for these two instructions the A and F are considered as a pair. There is also a special instruction to allow the contents of the HL pair to be exchanged with the word pointed to by the SP. In essence what happens is that a POP instruction is executed with the data going into internal storage, the HL is PUSHed, then the word POPed is put into the HL. This instruction is very useful in determining where a subroutine was called from, and makes possible the use of a special technique of multiple returns. This is possible because of another Z80 instruction, the CALL.

The CALL (and it's cousin the RST) instruction is a way to get to a subroutine, not unlike basic's GOSUB command. Whenever a CALL is executed, the CPU gets very busy. First, the current contents of the PC are PUSHed onto the stack, with all the associated incrementing of the SP. Then the next two bytes are read into the computer and are placed into the PC. In simpler terms, a CALL causes the computer to save it's place and then jump somewhere. To return from a subroutine that was CALLed, execute a RET instruction. This POPs the top of the stack into the PC, which in effect puts the computer back where it was before CALL was executed. Notice that I didn't say that the computer POPs the return point off the stack. That's because the computer hasn't got the foggiest notion where it's return point is. *The computer is dependent upon the programmer to assure that the return point is on top of the stack before a RET is executed!* Just about the only way of assuring this is to make sure that *for every PUSH in a subroutine, there is a corresponding POP.* Let's look at how the computer can be fooled into going somewhere that it doesn't belong. Suppose that our BC still contains 1234H. Let's also suppose that the SP contains F100H as before. Our program is toolin' along at warp 7, when it encounters a CALL to 6000H at address 4000H. The CPU reads address 4000H and increments the PC (incrementing the PC is always the first thing a computer does after reading an instruction). The instruction decode portion determines that a CALL is to be executed, so the next two bytes are read into the CPU, with the PC being incremented each time. Now the PC contains 4003H. The next thing the CPU does is to PUSH this value onto the stack, decrementing the SP before each byte, so that addresses F0FF and F0FE

contain 40 and 03 respectively. Then (finally) the data that was read before is installed in the PC. The next instruction read comes from the new address, 6000H.

(All of the above occurances are invisible to the programmer. All you really need to know is that the address of the instruction following the CALL is PUSHed onto the stack and a jump is done).

Now we are at address 6000H, and are toolin' along in our subroutine, when we decide that we need the use of our BC again. PUSH BC, and presto we can use it without loosing the value in it previously. We proceed along merrily, and when we are done, execute a RET. That's where we fall into the pits of OOPS. The last PUSH we executed was the BC, with the value 1234. The return POPs that value into the PC and, not knowing any better, proceeds with program execution from that point. Unfortunately, 1234 just happens to be somewhere in the ROM. What happens after that, I can only guess. One possibility might be the computer returning to your program, but with all the registers messed up. Or some kind of Level II error message might suddenly appear. Most likely, though, is the classic keyboard lockup or MEMORY SIZE?. Don't get me wrong. There are valid uses for techniques that "confuse" the computer. One of these is the use of "multiple return paths". This is a method used when you want program flow to return to one of several paths dependent upon the conditions of the return. This is a quite advanced technique, however, so we had better wait a while for that one.

There are a few other instructions that can deal with the SP. Among these are the INC SP and DEC SP. These two instructions can be used in a variety of ways, not the least of which is control of what gets POPed. If you had a value that was PUSHed onto the stack a while back, and several other items have been PUSHed on top of it, you can use the INC SP(twice for each POP you want to bypass) to get to it without disturbing the data above. Similarly, you can use the DEC SP to reset the SP to where it was before you went searching. Another use of these instructions is when you only want to save one byte. The Basic commands GOSUB and FOR do this. The GOSUB command PUSHes the current line number, the address within the line being executed, and the special code number 91H onto the stack. Although there are 3 PUSHes, only 5 bytes are needed to be saved, so after PUSHing the 91H, an INC SP is executed. When the command RETURN is encountered, the top of the stack is POPed off, a DEC SP is executed, and the byte received is compared with 91H. If it is not a 91H, an RG error is generated.

There are also the (LD SP,HL), (LD SP,IX), (LD SP,IY), and (ADD HL,SP) instructions. These are all ways of manipulating the stack to give you better control. But for our purposes, we won't be using them much.

There is something else to take into consideration; basic keeps it's own stack. It is imperative that it's integrity be untarnished if you expect to be able to interact with BASIC. If the routine you are writing is to be accessed via a USR call or something similar, you MUST be absolutely certain that for each and every PUSH, there is a subsequent POP. You must remember that your routine is essentially a subroutine of basic, and it demands that you keep things straight.

Included in this issue is a copy of LLIST, the program that we use here to get our line listings for the magazine. It demonstrates another valid place to put the stack, immediately above our program. Since the stack grows down (towards zero), and the PC usually goes up (towards FFFFH), there should seldom (if ever) be any conflicts.

This program includes examples of PUSH, POP, CALL, RET, and several of the assembler directives we have been discussing. Study it and see if you can make any sense of those instructions that haven't been discussed yet. Fear not, for we shall get to them, but it's a good exercise to try to figure out an instruction with just the manual and a program to work with . Meanwhile, I hope you'll try this program for yourself. ●

LLIST/CMD Formatter for Disk Systems

James W Crocker
Technical Editor

Many people have asked us to publish the program we use to format our line listings for the magazine. In case you haven't noticed, we use a program that automatically outputs a CR/LF and a specified number of spaces This results in much neater looking listings for the magazine. Well, here it is.

The program itself is quite simple. It's the initialization routine that takes a little explaining. I chose 8000H for the ORG for this routine because it was out of the way.

Lines 110-160 are the standard linkages to the normal lineprinter routine. Line 170 simply assures me that the program won't get overwritten by the stack (DOS automatically resets the stack pointer as soon as it gets control back).

Lines 180-200 set up to print MSG1, calls the print routine, and calls the keyboard input routine. The keyboard routine sets the carry flag if an up-arrow is hit, indicating that the user wants to use the default values of 64 characters/line and 5 spaces after the forced CR/LF. Line 210 checks for this condition and exits the initialization routine if found.

"Select the width of your Printed Listings!"

If the up-arrow is not found, then the buffer pointed to by HL contains the ASCII representation of the decimal number of characters/line desired. A CALL to 1E5AH converts this to a HEX number and returns this value in DE. Since we are not interested in values over 256, we concern ourselves only with the E register, which is stored in MAXLEN+1 by lines 230-240.

Lines 250-340 perform a quite similar operation to determine the number of spaces to print after a forced CR/LF, except that lines 300-320 verify that you don't want more spaces than the maximum number of characters. Again the up-arrow (carry flag set) indicates that you want to use the default value. Line 350 (QUIT) simply returns control to DOS.

Lines 360-410 are a very simple print to screen routine. I chose not to use any more ROM calls than necessary, since some don't work as expected with DOS in

memory. The routine expects the HL to point to the character string to be printed, and that that string terminates in a zero byte. Lines 360-380 get the character, check for a zero byte, and return if it is found. Lines 390-410 actually print the character, increment the HL to point to the next character, and loop around to do it again. Very simple.

Lines 420-630 constitute the keyboard input routine. Since there are more contingencies to deal with, the program must be longer than the video print routine.

Line 420 loads the HL with the address of the input buffer. Line 430 actually calls the ROM keyscan routine, and provides a looppoint for continuing input. The next two lines simply check to see if a key was hit, and loops back to LP2 if not. Lines 460-490 check for the up-arrow key, skips to CONT if not, and sets the carry flag and returns if so.

CONT, lines 500-560 checks for and deals with the (ENTER) and backspace keys. TERM, lines 570-590 store the terminating character *(ENTER)* and return to the calling program. BCKSPC decrements the HL and prints a "backspace and erase" character to the screen. Embedded in this routine is PRTIT, which prints the key hit (Level II does not automatically print on keyboard scan) and loops back around to LP2 again.

Lines 640-690 define the buffer space and the two messages to be printed. Note that all of the program up to line 700 is expendable, and therefore does not require protection. It will, however, do a very nice job of destroying any BASIC program already in memory, so don't try to pull a 'CMD "LLIST"' from Basic.

Line 700 actually begins the print formatter program. In fact you can live without lines 170-340 and 360-690 if you don't want to be able to specify the number of characters/line and number of spaces after a forced CR/LF.

Line 700 specifies the ORG for a 48K system. Use OBFAC if you have 32K.

LLIST, line 710, is the address that will be linked into the line printer drive routine address. The calling routine places the character in the C register, so we must first get it into the A so we can work with it. The first thing we do is check for a Carriage Return or "Top Of Form" character. Level II and DOS handle these the same, so I do also, with a relative jump to OUTCR.

If neither of these characters is found, we must bump our counter (TEMP) by one. Then we check to see if we equal the maximum character count (the default is supplied as 64. Changing this number will change the default maximum characters/line). If we have, we must output our CR and spaces, otherwise we simply continue on as normal (JR ENDIT).

OUTCR (840-880) handles a normal (not forced) CR by zeroing the character count and jumping to ENDIT.

CRLF (890-990) saves the character to be sent by PUSHing the BC onto the stack, then prints a carriage return to the line printer, after changing the character count to one (to compensate for the character on the stack). The B is then loaded with the number of spaces to send, the call is made, and a Decrement and Jump if Not Zero loop is entered to send them, incrementing the character count each time. Finally the BC and HL are POPed to keep the stack straight and to get the original character back in C, and a Jump is made to the normal printer driver.

Last but not least, line 1010 defines the location and initial value of TEMP, and 1020 simply tells me how much memory I have used. Line 1030 directs the EDTASM to END the assembly, and instructs it to autostart at START (line 110).

That's how the routine works. I sincerely hope that you find it as useful as we do. ●

```
8000            00100           ORG     8000H           ;A CONVIENIENT PLACE
8000 2A2640     00110 START     LD      HL,(4026H)      ;INITIALIZATION
8003 22D1FF     00120           LD      (DRIVER+1),HL   ;       "
8006 22D8FF     00130           LD      (DRIV2+1),HL    ;       "
8009 22E0FF     00140           LD      (ENDIT+1),HL    ;       "
800C 21ACFF     00150           LD      HL,LLIST        ;       "
800F 222640     00160           LD      (4026H),HL      ;       "
8012 310080     00170           LD      SP,START        ;STORE STACK POINTER
8015 218780     00180           LD      HL,MSG1         ;PRINT MESSAGE ONE
8018 CD4280     00190           CALL    PRINT           ;       "
801B CD4B80     00200           CALL    INPUT           ;GET KEYBOARD INPUT
801E 381F       00210           JR      C,QUIT          ;NO FURTHER ACTION NEEDED
8020 CD5A1E     00220           CALL    1E5AH           ;CONVERT TO HEX
8023 7B         00230           LD      A,E             ;HEX IN DE, WE NEED E
8024 32BCFF     00240           LD      (MAXLEN+1),A    ;STORE IT
8027 219E80     00250 LP1       LD      HL,MSG2         ;PRINT MESSAGE TWO
802A CD4280     00260           CALL    PRINT           ;       "
802D CD4B80     00270           CALL    INPUT           ;GET KEYBOARD INPUT
8030 380D       00280           JR      C,QUIT          ;IN CASE OF UP-ARROW
8032 CD5A1E     00290           CALL    1E5AH           ;CONVERT TO HEX
8035 3ABCFF     00300           LD      A,(MAXLEN+1)    ;GET LINE LENGTH
8038 BB         00310           CP      E               ;COMPARE TO # OF SPACES
8039 38EC       00320           JR      C,LP1           ;TO MANY SPACES
803B 7B         00330           LD      A,E             ;GET HEX IN A
803C 32D4FF     00340           LD      (NUMSPC+1),A    ;STORE IT
803F C32D40     00350 QUIT      JP      402DH           ;DONE, RETURN TO DOS
8042 7E         00360 PRINT     LD      A,(HL)          ;GET CHARACTER
8043 B7         00370           OR      A               ;ZERO?
8044 C8         00380           RET     Z               ;YES, DONE
8045 CD3300     00390           CALL    33H             ;PRINT IT
8048 23         00400           INC     HL              ;BUMP POINTER
8049 18F7       00410           JR      PRINT           ;NEXT CHARACTER
804B 217380     00420 INPUT     LD      HL,BUFF         ;HL=)BUFFER
804E CD2B00     00430 LP2       CALL    2BH             ;GET KEY PRESSED
8051 B7         00440           OR      A               ;WAS THERE ONE?
8052 28FA       00450           JR      Z,LP2           ;NO, TRY AGAIN
8054 FE5B       00460           CP      5BH             ;UP-ARROW?
8056 2002       00470           JR      NZ,CONT         ;NO
8058 37         00480           SCF                     ;SET CARRY FLAG
8059 C9         00490           RET
805A FE0D       00500 CONT      CP      0DH             ;CR?
805C 2808       00510           JR      Z,TERM          ;YES, TERMINATE
805E FE5F       00520           CP      5FH             ;BACKSPACE?
8060 2809       00530           JR      Z,BCKSPC        ;YES, HANDLE IT
8062 77         00540           LD      (HL),A          ;ELSE STORE IT
8063 23         00550           INC     HL              ;BUMP POINTER
8064 1808       00560           JR      PRTIT           ;PRINT CHAR AND CONTINUE
8066 77         00570 TERM      LD      (HL),A          ;TERMINATOR
8067 217380     00580           LD      HL,BUFF         ;HL=)BUFFER AGAIN
806A C9         00590           RET
806B 2B         00600 BCKSPC    DEC     HL              ;DEC POINTER
806C 3E08       00610           LD      A,08            ;BACKSPACE CHARACTER
806E CD3300     00620 PRTIT     CALL    33H             ;PRINT CHARACTER
8071 18DB       00630           JR      LP2             ;GET NEXT ONE
0014            00640 BUFF      DEFS    20              ;SHOULD BE ENOUGH ROOM
8087 45         00650 MSG1      DEFM    'ENTER MAX LINE LENGTH '
809D 00         00660           DEFB    0
809E 0D         00670 MSG2      DEFB    0DH
809F 45         00680           DEFM    'ENTER NUMBER OF SPACES '
80B6 00         00690           DEFB    0
FFAC            00700           ORG     0FFACH          ;FOR 48K
FFAC 79         00710 LLIST     LD      A,C             ;CHARACTER IN C
```

```
FFAD FE0D      00720          CP      0DH          ;CARRIAGE RETURN?
FFAF 2811      00730          JR      Z,OUTCR      ;YES, HANDLE IT
FFB1 FE0C      00740          CP      0CH          ;TOP-OF-FORM?
FFB3 280D      00750          JR      Z,OUTCR      ;YES, SAME AS CR
FFB5 E5        00760          PUSH    HL           ;SAVE HL
FFB6 21E2FF    00770          LD      HL,TEMP      ;TEMP KEEPS COUNT
FFB9 34        00780          INC     (HL)         ;OF # OF WORDS SENT
FFBA 7E        00790          LD      A,(HL)       ;GET COUNT IN A
FFBB FE40      00800 MAXLEN   CP      64           ;64 SENT?
FFBD 280C      00810          JR      Z,CRLF       ;YES, SEND A CR/LF
FFBF E1        00820          POP     HL           ;ELSE GET HL BACK
FFC0 181D      00830          JR      ENDIT        ;AND CONTINUE
FFC2 E5        00840 OUTCR    PUSH    HL           ;SAVE HL
FFC3 21E2FF    00850          LD      HL,TEMP      ;GET TEMP'S ADDRESS IN HL
FFC6 3600      00860          LD      (HL),0       ;AND RESET TO ZERO
FFC8 E1        00870          POP     HL           ;GET HL BACK
FFC9 1814      00880          JR      ENDIT        ;AND PROCEED
FFCB C5        00890 CRLF     PUSH    BC           ;MUST SAVE BC
FFCC 0E0D      00900          LD      C,0DH        ;SEND CR TO PRINTER
FFCE 3601      00910          LD      (HL),1       ;RESET COUNTER TO ONE
FFD0 CD0000    00920 DRIVER   CALL    $-$          ;SEND IT
FFD3 0605      00930 NUMSPC   LD      B,5          ;NUMBER OF SPACES TO SEND
FFD5 0E20      00940 LOOP     LD      C,20H        ;20=SPACE
FFD7 CD0000    00950 DRIV2    CALL    $-$          ;SEND IT
FFDA 34        00960          INC     (HL)         ;BUMP COUNT
FFDB 10F8      00970          DJNZ    LOOP         ;REPEAT 4 TIMES
FFDD C1        00980          POP     BC           ;GET BC BACK
FFDE E1        00990          POP     HL           ;AND HL, TOO
FFDF C30000    01000 ENDIT    JP      $-$          ;AND QUIT
FFE2 00        01010 TEMP     DEFB    0            ;COUNTER
FFE3           01020 THEEND   EQU     $
8000           01030          END     START
00000 TOTAL ERRORS
BCKSPC 806B 00600   00530
BUFF   8073 00640   00420 00580
CONT   805A 00500   00470
CRLF   FFCB 00890   00810
DRIV2  FFD7 00950   00130
DRIVER FFD0 00920   00120
ENDIT  FFDF 01000   00140 00830 00880
INPUT  804B 00420   00200 00270
LLIST  FFAC 00710   00150
LOOP   FFD5 00940   00970
LP1    8027 00250   00320
LP2    804E 00430   00450 00630
MAXLEN FFBB 00800   00240 00300
MSG1   8087 00650   00180
MSG2   809E 00670   00250
NUMSPC FFD3 00930   00340
OUTCR  FFC2 00840   00730 00750
PRINT  8042 00360   00190 00260 00410
PRTIT  806E 00620   00560
QUIT   803F 00350   00210 00280
START  8000 00110   00170 01030
TEMP   FFE2 01010   00770 00850
TERM   8066 00570   00510
THEEND FFE3 01020
```

OWN A TRS 80*?.....
SELECTRA-PRINT Puts You Just A Cable Length Away From A Word Processing System!

...just a simple hook-up with the supplied cables and your SELECTRA-PRINT is ready to run.

For Word Processing on a TRS 80, for example, just command "LPRINT" and SELECTRA-PRINT automatically outputs clear, clean, high-fidelity, hard copy....and of course you can use it to print-out any other information you might need.

SPECIAL NOTE:
SELECTRA-PRINT is a Selectric typewriter and although it has been modified for computer print-out, it may still be used as a standard office typewriter.

SELECTRA-PRINT is VERSATILE!
SELECTRA-PRINT is compatible with most micro and mini-computers including ☐ Apple ☐ Heath H8 ☐ IMSAI ☐ Cromemco ☐ Alpha Micro-systems ☐ Space Byte ☐ North Star Horizon ☐ SWTP ☐ Vector Graphic ☐ Sol ☐ Polymorphic ☐ Digital Group ☐ Ohio Scientific ☐ Altair ☐ Sorcerer ☐ Xitan ☐ Rex ☐ KIM ☐ EXORcisor

Already own a Selectric I™, II™, or III™?
We can convert it to Selectra-Print
dealer inquiries invited

MICRO COMPUTER DEVICES, INC.
3156 East La Palma Avenue, Suite H, Anaheim, California 92806 - (714) 630-8206
*TRS 80 is a Trademark of Radio Shack, a division of the Tandy Corporation

Omikron's Mapper + NEWDOS/80 8" Drives for the TRS-80

NEWDOS/80 is Apparat's latest upgrade to NEWDOS. Features include variable length records, chaining, and drivers specifically configured for Omikron's MAPPER II. $150.

MAPPER II adapts the TRS-80 to run both 5" and 8" drives. With NEWDOS/80, storage is increased to 300K per 8" drive. $99 plus $10 per cable connector.

MAPPER I adapts the TRS-80 to run the vast library of CP/M software as well as the TRS-80 software. All Lifeboat Software may be ordered for the MAPPER I. All MAPPER I CP/M software is compatible with the CP/M for the Model II. With MAPPER II and 8" drives, the Model I becomes disk compatible with the Model II. Standard features include lower case support, serial and parallel printer drivers, and an addressable cursor. MAPPER I is supplied with complete utilities including a memory test, a disk test, a copy program, and a proprietary program for converting TRS-DOS files to CP/M files. $199.

WORD PROCESSING—MAPPER I supports professional word processors like the Magic Wand and Word Star (see reviews in June 80 Kilobaud). Omikron's implementation includes a blinking cursor, auto repeat, shift lock, debouncing, and an input buffer that eliminates missed characters. Magic Wand super discount price $299.

FIELD PROVEN DESIGNS—After one year of MAPPER production, Omikron has established an impeccable reputation for reliability, integrity, and user support. Omikron's customers include the US Government, major corporations, universities, medical doctors, and professionals in all fields.

SYSTEMS—Omikron sells complete systems featuring Model II compatible Shugart disk drives. Call for prices and delivery.

FOREIGN ORDERS must include full payment in US funds plus $25 for air shipping and handling.

See reviews in July 80 and August 80 BYTE By Jerry Pournelle.

*CP/M is a TM of Digital Research TRS-80 is a TM of Tandy Corporation.

Let There Be Light!

by Robert Labenski
West Hartford, CT

Recently light pens have been made available for TRS-80s. These pens range in price from around 20 to 70 dollars. Before jumping into this new development a little education would be beneficial. What are they? What can they do? Can a simple one be home-brewed for less than five dollars?

Well with a little luck and your patience maybe we can answer a few of these questions and build a simple but effective one for dollars.

First of all, a light pen is a device which allows a programmer to sense a selected position on the video screen. This is done when the operator points a device (pen) at some location on the screen. Hopefully the selected spot has been highlighted by the programmer enough to draw the operator's attention to it and hence the selection has some significance to the program. On larger systems (IBM 3270) the programmer can sensitize portions of the screen so hardware can assist in the selection or highlighting of the screen. Once the spot has been selected, the program receives feedback as to where (what field) it was and then through programming takes some appropriate action.

Not as complex as it might appear, is it? Now that the program can sense something being pointed to, what can be done with it? The normal use is to provide menu selection. This simple application is intended to reduce keystrokes and the education requirements of the operator. On mini's and micro's the joy of drawing pictures might be added if desired.

Conceptually the idea of what a light pen is and what it is used for is not too difficult. A little creativity will and has produced some unusual and dramatic effects when used wisely.

Are you still interested? Good, let's try to build one. The one I have designed is not too sophisticated for the average builder, nor does it require parts which are difficult to get. As a ham (amatuer radio operator) all the parts came from my junk box. I would expect most non-junk collectors to have little trouble in locating these common parts. The design is straightforward. The objective is to sense light regardless of the source: The sun, a light bulb or any source. This approach leaves verification of source to the program and extends the pen (photo cell) to any other application not related to the video.

Referencing **Figure 1**, let's look at the basic design. First of all, the parts layout and construction techniques are not critical for performance. (That will make it easier for most of us.) The design is based on an operational amplifier used as a differential amplifier. This application uses a versatile circuit which will recognize and amplify a voltage difference. The reference voltage is determined by resistors R1 and R2, the differential voltage is created by R3 and the Photoelectric Cell. The resistence of the one I am using is about 2K in light and 100K in the dark. This network creates a base for the OP AMP to work with. When light is sensed a pulse is generated. This pulse is fed into the cassette input port which detects it and sets the input latch.

We have a pulse, a latch and a port. The cassette port is accessed using the INP and OUT instructions. The port address is 255. When the latch has been triggered the data presented to the INP instruction is 255 decimal, and 127 if not triggered. The latch must be reset each time it is triggered, using the OUT instruction. (Out 255,0 will do it.) Because the data is latched and can be read at any time, the push button can be checked by the program and reset at will. This will give you another "BREAK" key to add to your programs.

The pen is constructed with a little creativity. The photo cell I used fit very nicely in the shell of a test probe. The leads are not critical. Zip cord, a twisted pair of wires or some Hi Fi speaker wire will do just fine. The cord was terminated with a plug to allow removing the light pen. This will allow using the push button to trigger the cassette port. Programming for the push button is the same as for the photo cell.

Now the fun of programming it. I have included some examples to get you started.

The program listing in **Figure 2** has four distinct functions and examples. Program lines 10 - 80 are just a service function to allow grouping and selecting these routines.

Lines 1000 - 1060 (selection "Basic Check Out") are written to display an "X", graphic character, and an "I". Using this test, point the pen at the "X" or "I" and adjust the brightness and contrast so they are "selected". The adjustment should be such that when the pen is moved to another part of the screen "not selected" is displayed. Now point the pen at the graphic block. "Selected" should blink and then it should return to "Not Selected". This is because the persistence of the screen and the size of the character will only generate one pulse. Note this test will accept any light source so you can also test the push button.

The next test is Lines 100 - 190 (selection Basic Ball Bounce). This test displays a graphic character which moves down the left side of the screen. By placing the pen in the path of the character it can be detected. Adjust the brightness again if necessary to ensure detecting the graphic character. If you have installed the push button, remove the pen and press the button to stop the graphic. This test will give you some idea of what can be done with the push button.

The bouncing ball of the last test now can be put to use in a menu application. Lines 300 - 530 (ball select) use a graphic character which ripples through the choices. By pointing the pen at the graphic as it passes the desired entry selection is made. Lines 300 - 360 display the menu choices. The key to this routine is blinking the spot to verify it is not some other steady light source. Statements 440 - 480 blink

the spot. The "FOR" loop in lines 470 and 380 wait for the screen's persistence to fully darken the spot. Your selection can be picked off in statement 490. At that point "A" contains the number selected.

The last example is "Blink Select". Lines 700 - 840 set-up a video presentation which is a solid display until light is sensed. At that time each entry is blinked until the selected one is found. Lines 710 - 730 establish the menu. Line 750 waits for some light to be sensed. When it is, Lines 760 - 830 ripple the entries verifying what was selected. Line 800 is the point the selection is noted and "A" contains the number in the list.

I will grant these examples are not very sophisticated but they do show with a minimum of hardware and a little tricky programming you *can* make a five dollar light pen. Not only a pen for sensing screen location but a simple interface to the world (any light will do) and a programmable push button for additional capability. ●

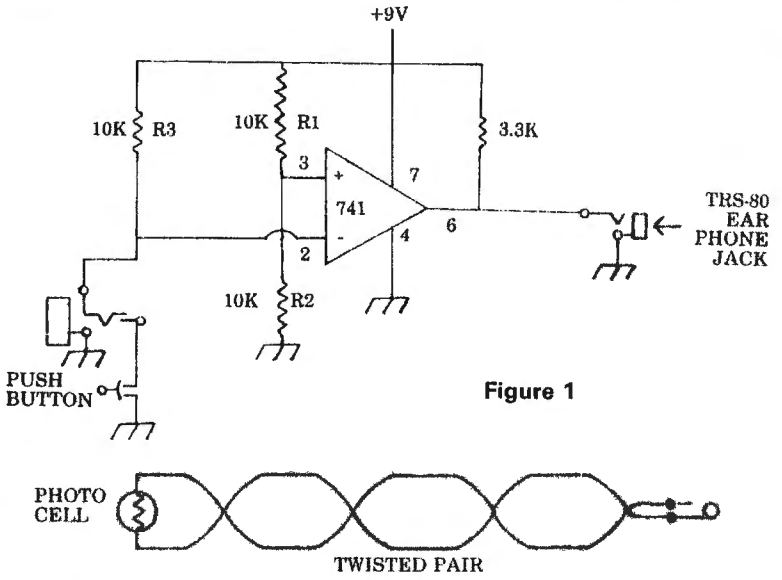


Figure 1

Figure 2

```
10 CLS
20 PRINT"1<BASIC CHECK OUT"
30 PRINT"2<BALL BOUNCE"
40 PRINT"3<BALL SELECT"
50 PRINT"4<BLINK SELECT"
60 INPUT A
70 ON A GOTO 1000,110,310,700
80 GOTO 10
100 'BASIC BALL BOUNCE
110 CLS
120 CLS:OUT 255,0
130 FOR A=0 TO 15
140 PRINT @64*A,CHR$(140)
150 FOR B=1 TO 10:NEXT
160 IF INP(255)=255:PRINT @64*A+3,"GOT
    IT":OUT 255,0:GOTO 160
170 PRINT @64*A," ";
180 NEXT A
190 GOTO 120
300 'BALL SELECT
310 CLS
320 PRINT
330 FOR A=1 TO 10
340 PRINT"MENU CHOICE";A
350 NEXT A
360 FOR A=1 TO 10
370 IF INP(255)=255 THEN OUT 255,0
380 PRINT @ 16+64*A,CHR$(140);:FOR B=1 TO 10:
    NEXT B
400 PRINT @15+64*A,"  ";
410 IF INP(255)=255 THEN 440
420 NEXT A
430 GOTO 360
440 OUT 255,0 :FOR B=1 TO 10:NEXT
450 IF INP(255)=255 THEN 420
460 PRINT @16+64*A,CHR$(140);
470 FOR B=1 TO 10:NEXT
480 IF INP(255)<>255 THEN 400
490 PRINT"CONSIDERED SELECTED";
500 FOR B=1 TO 1000:NEXT
510 PRINT @ 15+64*A,"
520 OUT 255,0
530 GOTO 360

700 'BLINK SELECT
710 CLS
720 PRINT
730 FOR A=1 TO 10:PRINT @ 30+64*A,"MENU
    SECTION";A;:NEXT
740 FOR A=1 TO 10
750 IF INP(255)<>255 THEN 820
760 PRINT @30+64*A,"          ";
770 OUT 255,0:FOR B=2 TO 10:NEXT B
780 IF INP(255)=127 PRINT @30+64*A,"ZZZZZZZZ
    ZZZZZZZZ";ELSE GOTO 820
790 FOR B=1 TO 10:NEXT:IF INP(255)<>255
    THEN 820
800 PRINT @ 10+64*A,"SELECTED";
810 FOR B=1 TO 1000:NEXT B:PRINT @ 10+64*A,
    "          ";
820 PRINT @30+64*A,"MENU SECTION";A;
830 NEXT A
840 GOTO 740


1000 'BASIC CHECK OUT
1010 CLS:OUT 255,0
1020 PRINT @ 530,CHR$(140);" X ";CHR$(191);" I "
1030 IF INP(255)=127 PRINT @ 260,"NOT
    SELECTED";
1040 IF INP(255)=255 PRINT @ 260,"SELECTED";
1050 OUT 255,0:PRINT @ 530,"      "
1060 GOTO 1020
```

Note: Leave as many spaces in line 760 as there are "Z's" in line 780.

Drawing Lines with the TRS-80

by Jeffrey C Ruble
Port Angeles, WA

Anyone who has worked with Applesoft BASIC has probably encountered that language's VLIN, HLIN and HPLOTX,Y to W,Z statements. With these the user can very easily have an Apple draw vertical, horizontal and diagonal line segments across the CRT. For example, with HPLOT all one has to do is specify the endpoints and the computer does the rest.

The attached subroutine will give TRS-80 owners the same capability as mentioned above. All one has to do is specify the endpoints (X1,Y1), (X2,Y2) of the line segment to be drawn and then GOSUB30000. The subroutine will draw the line segment. The line drawn is only as good as the TRS-80's low resolution graphics will allow, however. Also, the coordinate system is the one used by the SET function.

For a wild display try the following program:

```
10X1=RND(127) : Y1=RND(47) : X2=RND(127) :
    Y2=RND(47) : GOSUB30000 : GO TO 10

30000 '*** LINE DRAWING SUBROUTINE BY
      JEFF RUBLE **
30010 '
30020 '         (X1,Y1) = 1ST POINT
30030 '         (X2,Y2) = 2ND POINT
30040 '
30050 IF X1<0 OR X2<0 OR X1>127 OR X2>127 OR
      Y1<0 OR Y2<0 OR Y1>47 or Y2>47 THEN
      PRINT"BAD COORDINATES": RETURN
30060 '
30070  SX = SGN(X2 - X1) : SY = SGN(Y2 - Y1)
30080 IF SX = 0 THEN FOR Y = Y1 TO Y2 STEP SY :
      SET(X1,Y) : NEXT : RETURN
30090 SL = (Y2 - Y1)/(X2 - X1) : ' SL = SLOPE
30100 Y = Y1
30110 FOR X = X1 TO X2 STEP SX
30120         SET(X,Y) : Y = Y + SL*SX
30130 NEXT : RETURN
```

System/ Command

Keyboard Type-ahead

The 12th in a Series on Machine Language Applications

by Phil Pilgrim

"So I says ta J.M.,'Look, of course I do crossword puzzles while I'm workin'. I can't sit here starin'at the thing waitin' 'til it's ready for me to type somethin'. While it's busy, I like ta be busy, too. You know, sorta improvin' my mind.' Well, he just harumphs an' walks out. Jeez, Melissa, what'm I s'posed ta do? If he wanted ta complain, he oughta yell at the dumb computer. It wastes more time waitin' for me than I do it!"

"Your overwrought, Harry. Look at the bright side. It's Friday and..."

"Okay, boys and girls, coffee break's five minutes over already. Back to your desks, and don't forget to fasten your seat belts!"

"It's ol' Jowl Movement himself. Well, back ta the buttons. Oh, by the way, can you think of a five-letter word that means, 'A line of people, as in a supermarket'?"

"Try 'queue'."

"Cue's only got three letters."

"No, q-u-e-u-e, queue."

"Oh. Never heard of it, but I'll see if it fits. Thanks."

That's right, Harry, 'queue'. It's a word you ought to be more familiar with, because it might solve your problem with J.M. You see, if you could sit there typing away while the computer was busy digesting your previous input, you wouldn't have to wait for a prompt, and the computer wouldn't have to wait for an *ENTER*. Sure, those prompts are fine for beginners, but you've seen 'em all enough to key in data blindfolded. What you need is a type-ahead queuing system. While you type, the characters go into a queue, also known as a first-in first-out (FIFO) buffer. When the computer is ready for input, it doesn't need to wait for you. It just reads what's in the buffer, unless the buffer's empty, *then* it asks for the keyboard input. Most input queues are implemented as "circular" queues. These function exactly like those carousels you see in restaurants between the dining room and kitchen. The waiter puts your order under a clip on the carousel, right next to the previous order. As the cook finishes one order, he turns the carousel to the next one, takes it off, and begins work on it. The orders are processed in the same order in which they were taken. An input queue is like the carousel, with you being the waiter, the computer being the cook, and the orders being the characters typed in. If the queue becomes empty, the computer waits for you. If the queue fills up, you have to wait for the computer. Otherwise, you're both busy at the same time.

The trick in implementing an input queue is getting the computer to pay attention when you strike a key -- at least long enough to enqueue it (put it in the queue) for you. This is especially true of the TRS-80, in which the running program has to poll the keyboard just to see if a key is down. Fortunately, there are two solutions. One, in BASIC, the computer is almost always polling the keyboard while RUNning or LISTing a program, looking for a *BREAK* or *SHIFT*-@. Unfortunately, any other characters it finds, it just throws away, and this is a major problem to overcome. Second, under DOS, there is an easy way to manipulate the real-time clock to poll the keyboard every 25ms or 40 times per second. In either event, if a character is found, it will be enqueued. Then, when the keyboard is called by the regular input routines, these characters can be dequeued (taken out of the queue).

This enqueuing/dequeuing strategy holds *unless* the character keyed is a *BREAK* or *SHIFT*-@. Each needs to be brought to the computer's attention at once. And the former

should also cause the queue to empty itself. (You don't want things typed in as input before hitting *BREAK* to be interpreted as a command after hitting *BREAK*.)

The program shown here, KEYQ, does all the above, using both the BASIC and real-time clock polling techniques, without modification. In addition, the last eight characters keyed are displayed in the upper right-hand corner of the screen whenever a character is enqueued or dequeued. I tried a continuously updated display, but the screen flickered too much, so you have to be careful not to overPRINT the display or scroll it off the screen. Also note that control characters, including the left-arrow and *ENTER* are displayed as graphics characters. A control character has no effect on the queue itself - it's just enqueued with all the rest.

Taking a look at the listing now, notice that there are three major routines: PUTBUF, GETBUF, and START. PUTBUF is the keyboard polling routine. It is called every 25ms via the real-time clock and/or whenever BASIC is polling for a *BREAK*. It calls the regular keyboard routine and returns if no key is pressed. Otherwise it checks for a *BREAK* or *SHIFT*-@. If it finds either, it puts it in a special buffer, IBUF, for immediate use. If the character is a regular one, it is enqueued in BUFFER by PUTCH. BUFFER is 64 characters long and has two pointers associated with it: BUFOUT and BUFIN. BUFOUT points to the first character in the buffer (the one ready to go out). BUFIN points one character beyond the last character in the buffer (where the one coming in will go). If that character is non-zero, the buffer is full. Otherwise the new character is put there and BUFIN is advanced. If, in advancing, it passes the end of the buffer, it is moved to the beginning in INCDE. Finally, PUTBUF calls BDSPLY to display the updated buffer.

GETBUF is called in the regular keyboard calling sequence whenever the TRS-80 is looking for a character. The first thing it does is disable the real-time clock interrupt, because it's going to be fooling with the same pointers PUTBUF uses and shouldn't be interrupted. Next it calls PUTBUF in case a new key has been pressed. This is where the polling takes place in case you don't have a real-time clock or in case you've turned it off. Next it checks for a *BREAK* or *SHIFT*-@ in IBUF, returning either as top priority. In case of a *BREAK*, BUFFER is zeroed. With *SHIFT*-@, a toggle is flip-flopped. This toggle prevents characters from being taken out of the queue while BASIC is in a halt state. It also requires that another *SHIFT*-@ be issued to resume execution or listing, rather than "any key" as BASIC usually allows.

If IBUF is empty, GETOK takes over. Here the program checks to see who called GETBUF in the first place. If it was the *BREAK*-poller, we don't want to give it one of the enqueued characters, else the queue would empty out in nothing flat. So we peek at the return addresses in the stack. Both *BREAK*-pollers (one for RUN, one for LIST) reside in ROM between 1D00H and 1DFFH (at least in the three-ROM Level II. The newer ROMs might be different.), so we just check for the 1DH. Passing this test, we check for an empty queue and, barring this, get the character therefrom. Finally BUFOUT is advanced in a circular fashion by INCDE the same as BUFIN was.

START is executed once after KEYQ is loaded. It links GETBUF into the display calling sequence, then links PUTBUF into the 25ms interrupt sequence. This latter only works if you're using TRSDOS or NEWDOS. NEWDOS-80 users will want to check for different instructions on how this should be done. Level II users needn't worry about any of it, as long as no BASIC program is in memory when KEYQ is activated. Finally BUFFER is zeroed by BUFCLR, and BASIC takes over.

To use KEYQ, key it as shown into EDTASM, make whatever starting address modifications you need to, assemble it, and make a SYSTEM tape. Now initialize BASIC with a MEMORY SIZE corresponding to your ORG. Load your tape under SYSTEM and hit */ENTER* when done. KEYQ will be activated and awaiting your every keystroke. And, Harry, throw away that crossword puzzle book. You're a working man now! ●

```
7EE4                00100 MEM16K  EQU   7EE4H          ;MEM SZ=32484
BEE4                00110 MEM32K  EQU   0BEE4H         ;MEM SZ=48868
FEE4                00120 MEM48K  EQU   0FEE4H         ;MEM SZ=65252
7EE4                00130         ORG   MEM16K         ;FOR 16K. CHANGE TO SUIT.
7EE4 E67E           00140 PUTPTR  DEFW  PUTBUF         ;FOR REAL TIME CLOCK
7EE6 CD0000         00150 PUTBUF  CALL  $-$            ;GET CHAR FROM KBD.
7EE9 B7             00160         OR    A              ;IS IT ZERO?
7EEA C8             00170         RET   Z              ;  YES: RETURN
7EEB FE01           00180         CP    1              ;IS IT A BREAK?
7EED 2804           00190         JR    Z,IMM          ;  YES: IT'S IMPORTANT
7EEF FE60           00200         CP    60H            ;IS IT A SHIFT-@?
7EF1 2004           00210         JR    NZ,PUTCH       ;  NO: PUT IT IN QUEUE
7EF3 32A27F         00220 IMM     LD    (IBUF),A       ;SAVE FOR IMMEDIATE USE
7EF6 C9             00230         RET                  ;AND RETURN
7EF7 2AA47F         00240 PUTCH   LD    HL,(BUFIN)     ;POINTS BEYOND LAST CHAR
7EFA 47             00250         LD    B,A            ;SAVE CHAR
7EFB 7E             00260         LD    A,(HL)         ;PEEK INTO BUFFER
7EFC B7             00270         OR    A              ;IS IT FULL?
7EFD C0             00280         RET   NZ             ;  YES: THROW CHAR AWAY
7EFE 70             00290         LD    (HL),B         ;  NO: PUT CHAR IN BUFFER
7EFF EB             00300         EX    DE,HL          ;GET BUFIN IN DE
```

```
7F00 CD747F    00310          CALL    INCDE           ;INCREMENT AROUND QUEUE
7F03 ED53A47F  00320          LD      (BUFIN),DE      ;AND SAVE IT
7F07 CD7F7F    00330          CALL    BDSPLY          ;DISPLAY LAST 8 CHARS
7F0A C9        00340          RET                     ;AND RETURN
7F0B F3        00350 GETBUF   DI                      ;DISABLE INTERRUPTS
7F0C CDE67E    00360          CALL    PUTBUF          ;CHECK KEYBOARD
7F0F ED5BA67F  00370          LD      DE,(BUFOUT)     ;POINTS TO 1ST CHAR
7F13 3AA27F    00380          LD      A,(IBUF)        ;GET IMMEDIATE CHAR
7F16 B7        00390          OR      A               ;ANYTHING PRESSING?
7F17 2819      00400          JR      Z,GETOK         ;  NO: BRANCH AROUND
7F19 FE01      00410          CP      1               ;  YES: A BREAK?
7F1B 2006      00420          JR      NZ,GNOBRK       ;    NO: BRANCH AROUND
7F1D CD607F    00430          CALL    BUFCLR          ;    YES: CLEAR BUFFER
7F20 32A37F    00440          LD      (TOGGLE),A      ;         SET TOGGLE
7F23 F5        00450 GNOBRK   PUSH    AF              ;SAVE IMMEDIATE CHAR
7F24 3AA37F    00460          LD      A,(TOGGLE)      ;COMPLEMENT TOGGLE
7F27 EE01      00470          XOR     1               ;       .
7F29 32A37F    00480          LD      (TOGGLE),A      ;       .
7F2C AF        00490          XOR     A               ;CLEAR IMMEDIATE CHAR
7F2D 32A27F    00500          LD      (IBUF),A        ;       .
7F30 182B      00510          JR      GETOUT-1        ;AND GET OUT
7F32 3AA37F    00520 GETOK    LD      A,(TOGGLE)      ;CHECK HALT TOGGLE
7F35 EE01      00530          XOR     1               ;IN HALT STATE?
7F37 2825      00540          JR      Z,GETOUT        ;  YES: FORGET IT
7F39 ED73A07F  00550          LD      (TEMP),SP       ;PEEK INTO STACK
7F3D DDE5      00560          PUSH    IX              ;       .
7F3F DD2AA07F  00570          LD      IX,(TEMP)       ;       .
7F43 DD7E0F    00580          LD      A,(IX+15)       ;       .
7F46 DDE1      00590          POP     IX              ;       .
7F48 D61D      00600          SUB     1DH             ;CALL FROM 1DXXH?
7F4A 2812      00610          JR      Z,GETOUT        ;  YES: GIVE HIM A ZERO
7F4C 1A        00620          LD      A,(DE)          ;GET NEXT CHAR IN BUFFER
7F4D B7        00630          OR      A               ;A ZERO?
7F4E 280E      00640          JR      Z,GETOUT        ;  YES: FORGET IT
7F50 F5        00650 GETCH    PUSH    AF              ;  NO: SAVE IT
7F51 AF        00660          XOR     A               ;ZERO THAT BUFFER POS
7F52 12        00670          LD      (DE),A          ;       .
7F53 CD747F    00680          CALL    INCDE           ;INCREMENT POINTER AROUND
7F56 ED53A67F  00690          LD      (BUFOUT),DE     ;AND SAVE IT BACK
7F5A CD7F7F    00700          CALL    BDSPLY          ;DISPLAY LAST 8 CHARS
7F5D F1        00710          POP     AF              ;RESTORE BUFFER CHAR
7F5E FB        00720 GETOUT   EI                      ;REENABLE INTERRUPTS
7F5F C9        00730          RET                     ;ALL DONE
7F60 21E77F    00740 BUFCLR   LD      HL,BUFFER+63    ;START FROM END OF BUFFER
7F63 3600      00750          LD      (HL),0          ;PUT A ZERO THERE
7F65 54        00760          LD      D,H             ;DE=HL
7F66 5D        00770          LD      E,L             ;  .
7F67 1B        00780          DEC     DE              ;-1
7F68 013F00    00790          LD      BC,63           ;63 MORE PLACES TO ZERO
7F6B EDB8      00800          LDDR                    ;DO IT GOING BACKWARDS
7F6D 22A47F    00810          LD      (BUFIN),HL      ;SAVE POINTERS
7F70 22A67F    00820          LD      (BUFOUT),HL     ;       .
7F73 C9        00830          RET                     ;AND GET OUT
7F74 13        00840 INCDE    INC     DE              ;DE=DE+1
7F75 7B        00850          LD      A,E             ;CHECK E FOR OVERFLOW
7F76 FEE8      00860          CP      BUFFER+64(8(-8  ;PAST END OF BUFFER?
7F78 C0        00870          RET     NZ              ;  NO: ALL DONE
7F79 D640      00880          SUB     64              ;  YES: POINT TO FRONT END
7F7B 5F        00890          LD      E,A             ;       AND RESTORE
7F7C D0        00900          RET     NC              ;IF NO BORROW THEN DONE
7F7D 15        00910          DEC     D               ;BORROW FROM D
7F7E C9        00920          RET                     ;ALL DONE FOR SURE
```

```
7F7F ED5BA47F 00930 BDSPLY  LD      DE,(BUFIN)      ;ONE PAST LAST CHAR
7F83 21403C   00940         LD      HL,3C40H        ;SCREEN'S TOP RH CORNER +1
7F86 0608     00950         LD      B,8             ;DISPLAY 8 CHARACTERS
7F88 2B       00960 BLOOP   DEC     HL              ;BACKWARDS
7F89 1B       00970         DEC     DE              ;       .
7F8A 7B       00980         LD      A,E             ;PAST FRONT OF BUFFER?
7F8B FEA7     00990         CP      BUFFER-1(8(-8   ;       .
7F8D 2006     01000         JR      NZ,DEOK         ;  NO: OKAY
7F8F C640     01010         ADD     A,64            ;  YES: POINT TO END
7F91 5F       01020         LD      E,A             ;       AND RESTORE
7F92 3001     01030         JR      NC,DEOK         ;IF NO CARRY THEN OKAY
7F94 14       01040         INC     D               ;CARRY TO D
7F95 1A       01050 DEOK    LD      A,(DE)          ;GET CHAR FROM BUFFER
7F96 FE20     01060         CP      20H             ;CONTROL CHARACTER?
7F98 3002     01070         JR      NC,REGCH        ;  NO: DISPLAY AS IS
7F9A F680     01080         OR      80H             ;  YES: MAKE A GRAPHIC
7F9C 77       01090 REGCH   LD      (HL),A          ;PUT CHAR ON SCREEN
7F9D 10E9     01100         DJNZ    BLOOP           ;BACK FOR ANOTHER ONE
7F9F C9       01110         RET                     ;OVER AND OUT
0002          01120 TEMP    DEFS    2               ;FOR STACK POINTER
7FA2 00       01130 IBUF    DEFB    0               ;FOR IMMEDIATE CHAR
7FA3 00       01140 TOGGLE  DEFB    0               ;FOR HALT INDICATOR
7FA4 A87F     01150 BUFIN   DEFW    BUFFER          ;LAST CHAR POSITION+1
7FA6 A87F     01160 BUFOUT  DEFW    BUFFER          ;FIRST CHARACTER POSITION
0040          01170 BUFFER  DEFS    64              ;*** THE QUEUE! ***
7FE8 2A1640   01180 START   LD      HL,(4016H)      ;PUT GETBUF IN KBD CHAIN
7FEB 22E77E   01190         LD      (PUTBUF+1),HL   ;       .
7FEE 210B7F   01200         LD      HL,GETBUF       ;       .
7FF1 221640   01210         LD      (4016H),HL      ;       .
7FF4 21E47E   01220         LD      HL,PUTPTR       ;LINK PUTBUF W/25MS CLOCK
7FF7 221045   01230         LD      (4510H),HL      ;       .
7FFA CD607F   01240         CALL    BUFCLR          ;ZERO BUFFER
7FFD C3CC06   01250         JP      06CCH           ;RETURN TO BASIC READY
7FE8          01260         END     START           ;AUTOSTARTS @ START
00000 TOTAL ERRORS
BDSPLY 7F7F 00930   00330 00700
BLOOP  7F88 00960   01100
BUFCLR 7F60 00740   00430 01240
BUFFER 7FA8 01170   00740 00860 00990 01150 01160
BUFIN  7FA4 01150   00240 00320 00810 00930
BUFOUT 7FA6 01160   00370 00690 00820
DEOK   7F95 01050   01000 01030
GETBUF 7F0B 00350   01200
GETCH  7F50 00650
GETOK  7F32 00520   00400
GETOUT 7F5E 00720   00510 00540 00610 00640
GNOBRK 7F23 00450   00420
IBUF   7FA2 01130   00220 00380 00500
IMM    7EF3 00220   00190
INCDE  7F74 00840   00310 00680
MEM16K 7EE4 00100   00130
MEM32K BEE4 00110
MEM48K FEE4 00120
PUTBUF 7EE6 00150   00140 00360 01190
PUTCH  7EF7 00240   00210
PUTPTR 7EE4 00140   01220
REGCH  7F9C 01090   01070
START  7FE8 01180   01260
TEMP   7FA0 01120   00550 00570
TOGGLE 7FA3 01140   00440 00460 00480 00520
```

by Wildon Terrell
Woodford, VA

A Simple Letter Writer

Here is a short program which you may find to be very handy for writing letters or memos for which you want more than one copy printed by the line printer. It is designed to take up to forty strings, but this can be changed by changing the "FOR X=" statement on Lines 18 and 100.

Each string will hold almost three lines on the screen, but you must press the DOWN ARROW to give a line feed to the printer at the right time. A line feed is also given to the printer after each ENTER. With this program "QUOTE MARKS" can also be printed.

After you have written all of the material that you want, just press ENTER without any spaces having been typed and the program will proceed to make your copies. You are also given the opportunity to make more copies if you want. Once you have answered the question on Line 150 with a "N" you cannot make any more copies since all of the data is cleared and you are brought back to the point of writing more data. ●

This Program Will Only Work with DISK BASIC

```
1 REM * * * LETTER/PRT
2 REM * * * BY WILDON TERRELL
10 CLS
15 CLEAR5000
16 DIMA$(40)
18 FORX=1 TO 40
20 A$=" ":LINEINPUTA$(X)
25 IFA$(X)=" "THEN90
30 NEXT
90 INPUT "HOW MANY COPIES DO
   YOU WANT";Y
97 INPUT "PRESS ENTER WHEN
   PAPER IS READY";W
100 FORX=1 TO 40:
110 PRINTA$(X)
120 IFA$(X)=" "GOTO 140
125 LPRINTA$(X)
130 NEXT
140 W=W+1:IFW<Y GOTO 97
145 B$="N"
150 INPUT "DO YOU WANT ANY
   MORE COPIES (Y/N)";B$
155 W=0
160 IFB$="Y"THEN90ELSE15
999 END
```

Fun & Games Department

by David Bohlke
Coggon, IA

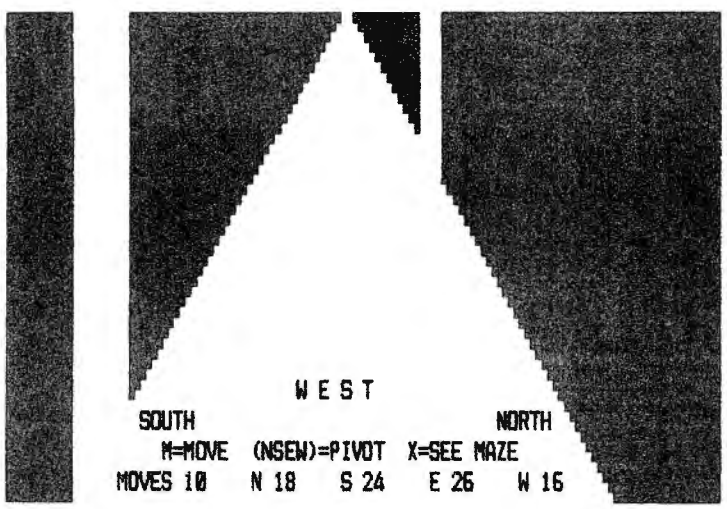
MAZE

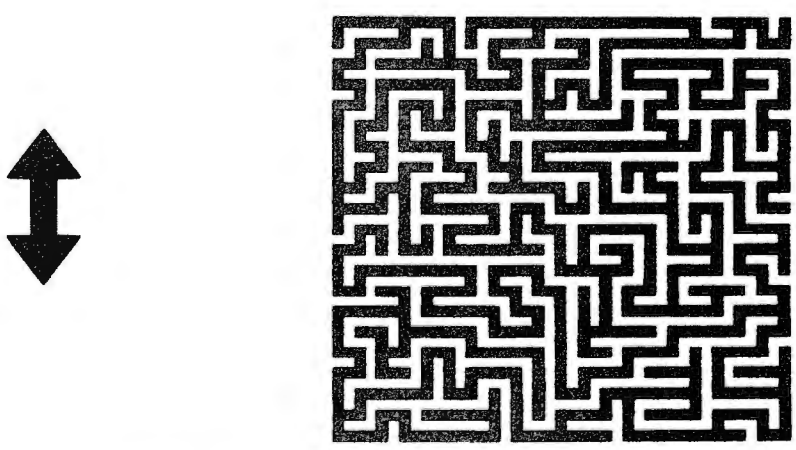
With *SOUND!*

Imagine that peace broke out one day. Also imagine that the Pentagon has long been deserted and empty. Now think of what would happen if someone led you into the center, blindfolded, and let you find your way out.

The first photo below is a three dimentional view of what you may encounter. Unlike the Pentagon, this Maze is a square, and you have the choice of it being 11 X 11 units through 45 X 45 units. You are told the straight-line distance to any wall, but you never know where you are to start or which way the corridors will take you. Each time the game is played, a new maze is constructed, an interesting thing to watch in itself.

If you really feel lost, you may use the "X" option to display the maze on the video and see where you are. If you do that though, you lose three times the maze dimension in points!





This game is for Model I, 16K Level II and up. It uses string packing and fast graphics. Suggestion: Don't start playing if you do not have several hours to finish. It is one of the few programs that can keep the kids occupied all afternoon and into the night!

```
10 GOSUB1510
20 CLEAR1000:DEFINT A,M,N,K,S:IFPEEK(16396)=201 THEN S2=16526
      ELSES2=23316
30 DIMA(45,45),M(400),N(400)
40 CLS:PRINTCHR$(23):PRINT"M A Z E    E N C O U N T E R":PRINT
50 PRINT"ENTER DIMENSION OF MAZE (11-45) ?";
60 A$=INKEY$:IFA$=""THEN60ELSEPRINTA$;:N=ASC(A$)-48
70 A$=INKEY$:IFA$=""THEN70ELSEPRINTA$;:Z=ASC(A$)-48
80 NZ=N*10+Z
90 IFNZ(11 OR NZ)45,40
100 IFNZ/2=INT(NZ/2),NZ=NZ+1
110 MZ=NZ
120 CLS
130 FORK=1TO27
140 READX:M$=M$+CHR$(X):NEXT
150 DATA
      205,127,10,77,68,62,1,105,
      211,255,45,32,253,60,105,
      211,255,45,32,253,13,16,
      238,175,211,255,201
160 S1=VARPTR(M$):POKES2,PEEK(S1+1):POKES2+1,PEEK(S1+2)
170 PRINT@50,"CURRENT":PRINT@114,"RESET":PRINT@178,"HIGH"
180 PRINT@960,"M A Z E    C O N S T R U C T I O N";:K=0
190 FORM=0TOMZ-1:A(M,0)=1:A(M,NZ-1)=1:NEXT
200 FORM=0TONZ-1:A(0,M)=1:A(MZ-1,M)=1:NEXT
210 M=RND(INT(MZ/4)*2)*2:N=RND(INT(NZ/4)*2)*2
220 IFM(4 OR M)MZ-4 OR N(4 OR N)NZ-4THEN210
230 SET(N*2,M):SET(N*2+1,M):A(M,N)=1
240 IFA(M+2,N)=1 ANDA(M-2,N)=1 ANDA(M,N+2)=1 ANDA(M,N-2)=1THEN3
    70
```

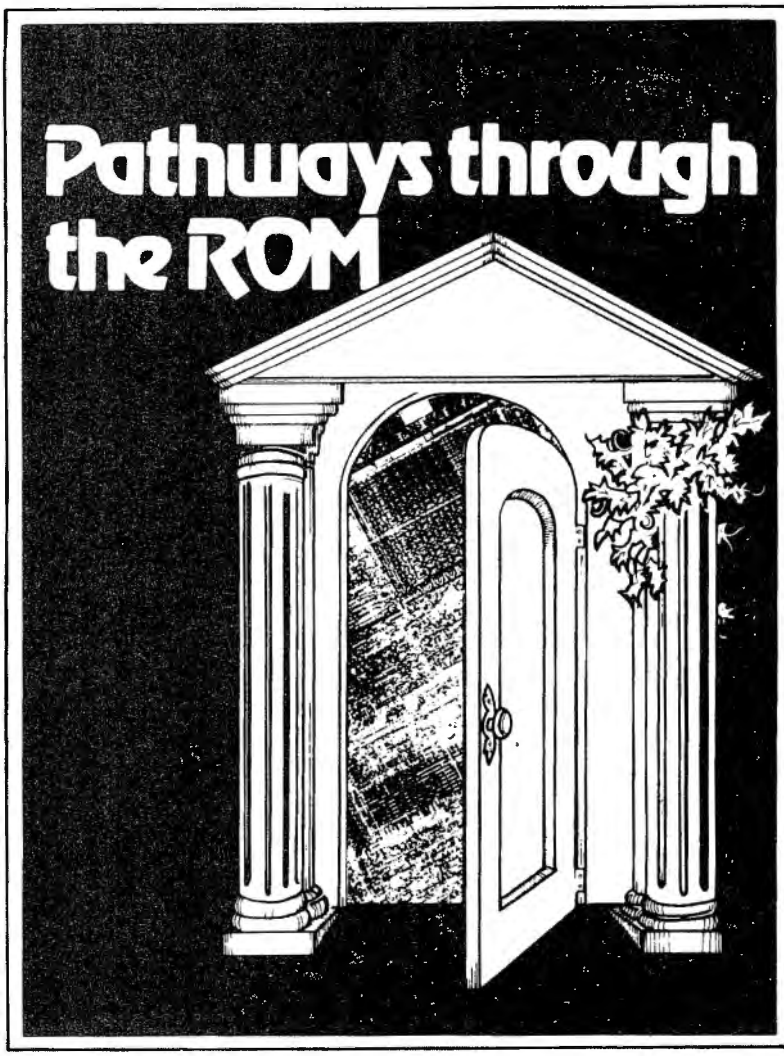
```
250 D=RND(4):ON D GOTO260 ,270,280,290
260 M1=-1:N1=0:GOTO300
270 M1=0:N1=1:GOTO300
280 M1=1:N1=0:GOTO300
290 M1=0:N1=-1
300 IFA(M+M1*2,N+N1*2)=1THEN250
310 SET((N+N1)*2,M+M1):SET((N+N1)*2+1,M+M1)
320 SET((N+N1*2)*2,M+M1*2):SET((N+N1*2)*2+1,M+M1*2)
330 A(M+M1,N+N1)=1:M=M+M1*2:N=N+N1*2:A(M,N)=1
340 K=K+1:M(K)=M:N(K)=N:PRINT@58,K;:L=USR(8E3+K)
350 IFAM(KTHENAM=K:PRINT@186,AM;:L=USR(5E3-K)
360 GOTO240
370 M=M(K):N=N(K):K=K-1
380 PRINT@122,K;:L=USR(9E3+K)
390 IFK=0THEN410
400 GOTO240
410 DATA 1,16,63,33,0,60,62,191,119,35,16,252,6,64,13,32,247,20
    1
420 PRINT@690,"PACKING";:PRINT@754,"STRINGS ---";
430 B$=" ":D$=CHR$(26)+CHR$(24):E$=CHR$(135):F$=CHR$(129)
440 G$=CHR$(159):H$=CHR$(139):I$=CHR$(130):J$=CHR$(175)
450 DR$=B$+D$+B$+D$+B$+D$+B$+D$+B$+D$+B$+D$+B$+D$+B$
460 DR$=DR$+D$+DR$
470 D$=CHR$(26):L$=CHR$(29)
480 W$=E$+STRING$(25,128)+H$+D$+L$+STRING$(15,191)+G$+F$
490 W$=W$+STRING$(27,128)+I$+J$+D$+L$+STRING$(14,191)+E$
500 W$=W$+STRING$(31,128)+H$+D$+L$+STRING$(12,191)+G$+F$
510 W$=W$+STRING$(33,128)+I$+J$+D$+L$+STRING$(11,191)+E$
520 W$(1)=W$+STRING$(37,128)+H$
530 W$=E$+STRING$(13,128)+H$+D$+STRING$(17,24)
540 W$=W$+G$+F$+STRING$(15,128)+I$+J$+D$+STRING$(20,24)
550 W$=W$+E$+STRING$(19,128)+H$+D$+STRING$(23,24)
560 W$(2)=W$+G$+F$+STRING$(21,128)+I$+J$
570 W$=E$+STRING$(7,128)+H$+D$+STRING$(11,24)
580 W$(3)=W$+G$+F$+STRING$(9,128)+I$+J$
590 W$(4)=E$+" "+H$+D$+STRING$(5,24)+G$+F$+"    "+I$+J$
600 W$=G$+F$+STRING$(39,128)+I$+J$+D$+L$+STRING$(8,191)
610 W$=W$+E$+STRING$(43,128)+H$+D$+L$+STRING$(6,191)
620 W$=W$+G$+F$+STRING$(45,128)+I$+J$
630 FORI=1TO18:READX:Z$=Z$+CHR$(X):NEXT
640 X1=PEEK(VARPTR(Z$)+1):X2=PEEK(VARPTR(Z$)+2)
650 R=INT(MZ/2)+1:C=R:RR=R:CC=C:MV=0:EG=0
660 IFA(R,C)=0THEN680
670 R=RR:C=CC:R=R+RND(3)-2:C=C+RND(3)-2:GOTO660
680 D=RND(4)
690 CLS:POKES2,X1:POKES2+1,X2:X=USR(0):POKE16383,191
700 PRINT@841,W$;
710 POKES2,PEEK(S1+1):POKES2+1,PEEK(S1+2):L=USR(4E3)
720 ON D GOTO730 ,740,750,760
730 D$=" N O R T H ":L$=" WEST ":R$=" EAST ":GOTO770
740 D$=" E A S T ":L$=" NORTH ":R$=" SOUTH ":GOTO770
750 D$=" S O U T H ":L$=" EAST ":R$=" WEST ":GOTO770
760 D$=" W E S T ":L$=" SOUTH ":R$=" NORTH "
770 R1=R:C1=C:WL=1
780 ON D GOTO790 ,800,810,820
```

```
790 R1=R1-1:GOTO830
800 C1=C1+1:GOTO830
810 R1=R1+1:GOTO830
820 C1=C1-1
830 IFA(R1,C1)=1 OR WL=5,960
  0 IFWL=1THENP=529 ELSE IF WL=2,P=279 ELSE IF WL=3,P=154 ELSEP
    =29
850 PRINT@P,W$(WL);:L=USR((WL+4)*1000)
860 J=0:K=0:GOSUB1190:ON WL GOTO870 ,900,930,950
870 IFJ=1PRINT@15,DR$;:PRINT@16,DR$;:PRINT@17,DR$;
880 IFK=1PRINT@43,DR$;:PRINT@44,DR$;:PRINT@45,DR$;
890 GOTO950
900 IFJ=1PRINT@22,DR$;:PRINT@23,DR$;
910 IFK=1PRINT@37,DR$;:PRINT@38,DR$;
920 GOTO950
930 IFJ=1PRINT@26,DR$;
940 IFK=1PRINT@34,DR$;
950 WL=WL+1:GOTO780
960 IFR(2ORC(2ORR)=MZ-2ORC)=MZ-2,EG=1:GOTO1320
970 K=0:J=0:R1=R:C1=C:GOSUB1190
980 IFJ=1THENFORI=6TO10:PRINT@I,DR$;:NEXT
990 IFK=1THENFORI=50TO54:PRINT@I,DR$;:NEXT
1000 PRINT@793,D$;:PRINT@843,L$;:PRINT@875,R$;
1010 CT=0
1020 PRINT@970,"MOVES";MV;"   N";R-1;"   S";MZ-R-2;"   E";MZ-C-
     2;"   W";C-1;
1030 PRINT@910,"M=MOVE  (NSEW)=PIVOT  X=SEE MAZE";
1040 A$=INKEY$:IFA$=""THEN L=USR(8E3+CT):CT=CT+1:GOTO1040
1050 IFA$="N"THEND=1:GOTO690
1060 IFA$="E"THEND=2:GOTO690
1070 IFA$="S"D=3:GOTO690
1080 IFA$="W"D=4:GOTO690
1090 IFA$="M"GOTO1120
1100 IFA$="X"MV=MV+3*MZ:GOTO1400
1110 GOTO1040
1120 IFD=1ANDA(R-1,C)=0R=R-1:GOTO1170
1130 IFD=2ANDA(R,C+1)=0C=C+1:GOTO1170
1140 IFD=3ANDA(R+1,C)=0R=R+1:GOTO1170
1150 IFD=4ANDA(R,C-1)=0C=C-1:GOTO1170
1160 GOTO960
1170 MV=MV+1
1180 GOTO690
1190 ON D GOTO1200,1230,1260,1290
1200 IFA(R1,C1-1)=0J=1
1210 IFA(R1,C1+1)=0K=1
1220 RETURN
1230 IFA(R1-1,C1)=0J=1
1240 IFA(R1+1,C1)=0K=1
1250 RETURN
1260 IFA(R1,C1+1)=0J=1
1270 IFA(R1,C1-1)=0K=1
1280 RETURN
1290 IFA(R1+1,C1)=0J=1
1300 IFA(R1-1,C1)=0K=1
1310 RETURN
1320 CLS:PRINT@832,"MOVES";MV
1330 IFEG=1PRINT@768,"YOU HAVE ESCAPED !!!";:FORI=-100TO75:L=US
     R(11111-I):NEXT
1340 PRINT@0,"YOUR OPTIONS . . .":PRINT:CT=0
1350 PRINT"1 - NEW GAME, NEW MAZE
2 - NEW GAME, SAME MAZE
3 - CONTIUE CURRENT GAME";
1360 A$=INKEY$:IFA$=""L=USR(9E3-CT):CT=CT+1:GOTO1360
1370 X=ASC(A$)-48:IFX=1RUN20ELSEIFX=2CLS:GOTO650
1380 IFX()31320
1390 GOTO690
1400 CLS
1410 FORX=2TOMZ-2:L=USR(7E3+X):FORY=2TOMZ-2
1420 IFA(X,Y)=1SET(Y*2,X):SET(Y*2+1,X)
1430 NEXT:NEXT
1440 PRINT@965,CHR$(93);" CURRENT POSITION   ***   ";
1450 PRINT"(ENTER) TO CONTINUE . . .";
1460 SET(C*2,R):SET(C*2+1,R):PRINT@960,CHR$(140);
1470 A$=INKEY$:IFA$()""1320
1480 RESET(C*2,R):RESET(C*2+1,R)
1490 L=USR(RND(1000))
1500 PRINT@960," ";:GOTO1460
1510 CLS:PRINTTAB(5),"M A Z E    E N C O U N T E R":PRINT
1520 PRINT"     YOU WILL SOON BE PLACED IN THE CENTER OF A MOUS
     E MAZE;"
1530 PRINT"BUT UNLIKE THE MOUSE, YOU WILL INITIALLY SEE A PERSP
     ECTIVE"
1540 PRINT"OF THE MAZE AS IT IS CONSTRUCTED.  THERE IS NO CHEES
     E IF YOU"
1550 PRINT"SHOULD ESCAPE, BUT PERHAPS A LITTLE PERSONAL SATISFA
     CTION.":PRINT
1560 PRINT"     TO MOVE THROUGH THE MAZE, JUST PRESS THE  -M-
     KEY."
1570 PRINT"YOU WILL THEN MOVE ONE LOCATION IN THE DIRECTION YOU
      ARE"
1580 PRINT"FACING.  TO CHANGE YOUR DIRECTION, PRESS  N,S,E, OR
     W.  EACH"
1590 PRINT"MOVE YOU MAKE WILL BE TABULATED; SO YOU CAN JUDGE YO
     UR"
1600 PRINT"MOUSE IQ.  IF YOU GET CONFUSED, PRESS THE  -X-  KEY
     AND YOUR"
1610 PRINT"POSITION IN THE MAZE WILL BE DISPLAYED.  SEEING THE
     MAZE WILL"
1620 PRINT"COST YOU  3  MOVES TIMES THE DIMENSION OF THE MAZE."
1630 PRINT:INPUT"(ENTER) TO CONTINUE . . .";A$
1640 CLS:PRINTTAB(10)"M A Z E    E N C O U N T E R":PRINT
1650 PRINT"     TO BEGIN, YOU MUST SELECT THE DIMENSION OF THE
     MAZE."
1660 PRINT"THIS SIZE MAY RANGE FROM 11 TO 45 SQUARE.  A SIZE OF
      11 IS"
1670 PRINT"EASY TO ESCAPE FROM, BUT A SIZE OF AROUND 25 GETS DI
     FFICULT."
1936 PRINT"THE LARGER SIZES ARE PURE FRUSTRATION.":PRINT
1690 PRINT"     SINCE YOU CAN'T SMELL THE CHEESE TO AID IN YOUR
      QUEST,"
1700 PRINT"THE STRAIGHT LINE DISTANCE FROM YOUR LOCATION TO THE
      MAZE EDGE"
1710 PRINT"WILL BE DISPLAYED (N,S,E,W).  ALSO, HOOK THE CASSETT
     E AUX"
1720 PRINT"PLUG TO A SMALL AMPLIFIER FOR THE SOUND EFFECTS."
1730 PRINT:PRINT"DAVID BOHLKE     COGGON, IA     52218":PRINT
1740 INPUT"(ENTER) TO BEGIN . . .";A$:RETURN
```

● Unlock the Hidden Power Of Your Computer!

● Program fast & easy!

● Use ROM routines in your BASIC and Assembly Language programs! All you need to know is in . . .



INCLUDES:

SUPERMAP
from Fuller Software
previously sold at $18.95

TRS-80 Disassembled Handbook
by Robert Richardson
previously sold at $10.00

Hex Mem
by John Phillipp
monitor written in BASIC.

Z-80 Disassembler
by George Blank

TO ORDER CALL TOLL-FREE
1-800-258-1790
In NH call 673-5144

● Guide to Level II BASIC and DOS Source Code $19.95 plus $1 shipping

Description of the contents of the Level II BASIC ROM by memory location, by function, and in lesson format. Includes several BASIC and Assembly Language programs in listing format to examine and use ROM routines.

A SoftSide Publication

The Software Exchange Library

TO ORDER TOLL-FREE

1-800-258-1790

(In NH call 673-5144)

MasterCard VISA

UNDERSTANDING MICROCOMPUTERS Scelbi

Here are the answers to hundreds of questions about microcomputers. Written in simple English. You get the fundamental concepts behind the operation of virtually all microcomputers. Convenient glossary covers all the key words.

Introduces BASIC language programming. Tells how to select a small computer system. Basic instructions for almost every class of microprocessor are illustrated along with details on how a CPU is organized and how it follows directives and solves problems. Explains flow charts, program worksheets and memory maps. $9.95 plus $1

INTRODUCTION TO LOW RESOLUTION GRAPHICS

Now you can produce amazing computer graphics - even if you can't draw a straight line. Literally! Learn how to draw lines and shapes, make graphs, draw pictures and even do animation. The simple secrets of how to do all this are contained in SCELBI's new book "Introduction to Low Resolution Graphics."

Today's exciting personal and small business computing machines are generally provided with at least some kind of "low resolution" graphics capability. What is low resolution graphics? It is graphics presented on a point-by-point basis where the number of points is limited to about 8000 or less. The APPLE II by APPLE computers, Inc., The Radio Shack TRS-80 and the Commodore PET all have low resolution graphics capability. So do many other kinds of microcomputers.

What can you do with low resolution graphics? All kinds of things . . . If you know how! You can plot plain and simple or fancy and complex graphs to consolidate data for business or pleasure purposes. But you can use the capabilities to improve the presentation and impact of almost anything you want your computer to tell people. It can be used to animate games or data, clarify and amplify educational materials, or just plain entertain people. $11.95 plus $1.

VINYL BINDERS

Quality vinyl binders with 12 metal rods, perfect for storing your back issues of **SoftSide** and upcoming catalogues. $4.95 plus $1

STIMULATING SIMULATIONS, 2nd Edition

by C. W. Engel

An exciting handbook containing twelve "simulation programs," which are actually game programs. They include: Art Auction, Monster Chase, Lost Treasure, Gone Fishing, Space Flight, Starship Alpha, Forest Fire, Nautical Navigation, Business Management, Rare Birds, Diamond Thief, and The Devil's Dungeon. Clearly written in a BASIC format that is compatible with all microcomputers, they are all original and well-documented. Each program is presented with a listing, sample run, instruction, and program documentation, including flow chart and ideas for variations.

$4.95 (plus $1.00 shipping charge)

A CONSUMER'S GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS by Stephen Freiberger and Paul Chew

Both an introduction to the principals of microcomputers that assumes no previous knowledge on the reader's part, and a review of 64 microcomputer products from over 50 manufacturers. Other features include: extensive illustrations to reinforce the discussions, a selection and sources section to assist in reviewing, selecting, and purchasing microcomputer products; summary charts of major microcomputer products offering a quick summary of specifications for a given product, and comment sections covering the advantages, disadvantages, and best-buy tips for each microcomputer product.

$8.60 (plus $1.00 shipping charge)

CP/M SOFTWARE SUMMARY GUIDE

Rainbow Associates announces the CP/M Software Summary Guide -- a concise, handy summary of the major software used on most CP/M systems. Included are summaries of the CP/M operation system, Microsoft BASIC, CBASIC™, and the CP/M utilities DESPOOL™, MAC™, and TEX.

The CP/M summary covers the commands (DIR, ERA, REN, SAVE, TYPE, USER) and utilities (ASM, DDT, DUMP, ED, LOAD, MOVCPM, PIP, STAT, SUBMIT, SYSGEN, XSUB). Each one is explained briefly and clearly with examples. The booklet also sumarizes all features of Microsoft's BASIC-80 (including the compiler) and Compiler System's CBASIC. Error codes for CBASIC-1, CBASIC-2, and BASIC-E are also summarized in a single alphabetical list. Examples and definitions clearly explain the workings of DESPOOL, MAC, and TEX utilities offered by Digital Research.

The CP/M Software Summary Guide, 60 pages long, is designed to be especially easy to use. Features are organized alphabetically, so the reader can find an explanation quickly rather than having to page through various function sections. This solves the single most common complaint programmers have about manuals and summary guides.

About the size of a paperback book, the Guide is handy to hold. Indenting and capitalization are used to make the information "jump out" at the reader. "The idea for this booklet is attributed to need; the format is attributed to common sense," says Bruce A. Brigham, Rainbow's Technical Consultant and author of the Guide. $3.75 plus $1

THE BASIC HANDBOOK

by Dr. David Lien

This book is unique. It is not a textbook. It's far more than a dictionary. It is a virtual ENCYCLOPEDIA of the BASIC language.

While not favoring one computer over another, it explains over 250 BASIC words, how to use them, and alternate strategies. Since over 50 computers are represented, yours is probably in here too. $14.95

THE MIND APPLIANCE: HOME COMPUTER APPLICATIONS

by T.G. Lewis

Chock full of unique and challenging ideas for applying your computer to home use. Make your computer write poetry, balance a checkbook, dial a telephone automatically, handle household budgets, menu planning, shopping lists, and income tax calculations. Dozens of BASIC language programs, including: Buying for the Home; The Garage (Fundamentals of BASIC); The Living Room (Simple Retrieval); The Kitchen (Advanced Retrieval Systems); The Bedroom (Word/Text Processing); The Den (Business Processing); The Bathroom (Graphics); The Split Level Home (ideas); and a Glossary.

$8.60 (plus $1.00 shipping charge)

TEN EASY PIECES: CREATIVE PROGRAMMING FOR FUN AND PROFIT

by Hans Sagan and Carl Meyer, Jr

An introduction to the BASIC language through computer games. Written in an informal style, it stimulates interest in creative programming of games of chance and of skill. Teaching by example, these games illustrate the various programming techniques at stages of difficulty which are suitable to almost unlimited modification, simplification, or amplification. Emphasis is on prompting creativity on the part of the reader. The text requires little knowledge of elementary mathematics.
$7.95 (plus $1.00 shipping charge)

CP/M SOFTWARE SUMMARY GUIDE

A concise, handy summary of the major software used on most CP/M systems. Included are summaries of the CP/M operating system, Microsoft BASIC, CBASIC$, and the CP/M utilties DESPOOL$, MAC$, and TEX.

The CP/M summary covers the commands (DIR, ERA, REN, SAVE, TYPE, USER) and utilities (ASM, DDT, DUMP, ED, LOAD, MOVCPM, PIP, STAT, SUBMIT, SYSGEN, XSUB). Each one is explained briefly and clearly with examples. The booklet also summarizes all features of Microsoft's BASIC-80 (including the compiler) and Compiler System's CBASIC. Error codes for CBASIC-1, CBASIC-2, and BASIC-E are also summarized in a single alphabetical list. Examples and definitions clearly explain the workings of DESPOOL, MAC, and TEX utilties offered by Digital Research.

The CP/M Software Summary Guide is designed to be especially easy to use. Features are organized alphabetically, so the reader can find an explanation quickly rather than having to page through various function sections.

$3.75 (plus $1.00 shipping charge)

Z-80 AND 8080 ASSEMBLY LANGUAGE PROGRAMMING

The best introduction to assembly language we sell. You should have experience in BASIC. By Kathe Spracklen $7.95 plus $1

INTRODUCTION TO LOW RESOLUTION GRAPHICS

Covers PET, TRS-80, Apple II. How to draw lines, create shapes, animated figures, charts. $11.95 plus $1

THE BASIC HANDBOOK

A definite reference/idea book. Explains over 50 favorite versions of the BASIC language in detail as used in micros, minis, and mainframes. Everything you need to know about the most important statements, functions, operators, and commands. From the same author as the TRS-80 Users/Learners Manual. $14.95 plus $1

Z80 INSTRUCTION HANDBOOK

Convenient pocket-size manual describes Z80 capabilities in easy-to-understand terms. Designed as a practical reference to mnemonics, machine codings, usage. For programmers of every level - beginner to professional - anyone working in Z80 machine or assembler language. Appendixed. $5.95 plus $1.

SARGON HANDBOOK

Complete documentation covering all algorithms in Sargon can be found in this guide book. Contains complete table of contents, block diagram of the program, four part introduction, Z-80 listing, index to subroutines. Fully annotated. $15.95 plus $1

TRS-80 ASSEMBLY LANGUAGE PROGRAMMING

Covers Z-80, and introduction to Assembly Language, the Radio Shack Editor/Assembler and T-Bug, and debugging methods; also explains how to move data, the use of arithmetic, compare, logic, and bit operators, shifts, strings, tables, input and output, and 12 commonly used subroutines. Well indexed and illustrated. $3.95 plus $1

THE SECRET GUIDE TO COMPUTERS

By Russ Walters A quickie course on computers. $5.95 plus $1

Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK

Over 100 usable subroutines, plus how to use them. $15.95 plus $1

TRS-80 DISK AND OTHER MYSTERIES

The serious programmers' guide to Disk BASIC and the wonders of NEWDOS +. $22.95 plus $1

6502 SOFTWARE GOURMET GUIDE AND COOKBOOK

Includes instruction set, floating point and decimal arithmetic, search and sort routines, and more. $12.95 plus $1

TRS-80 INTERFACING

What you need to know to connect your TRS-80 to the world. Assumes knowledge of some machine language programming. $8.95 plus $1.

PATHWAYS THROUGH THE ROM

The definitive guide to Level II BASIC. Includes Super Map by Fuller Software The TRS-80 Disassembled Handbook by Robert Richardson, HexMem by J. Philip, Z-80 Disassembler by George Blank and DOS Map by John Hartford. $19.95 plus $1

LEARNING LEVEL II

The User's Guide to Radio Shack Level II BASIC. $15.95 plus $1

LEARN MICRO-COMPUTERS

A new multimedia information package. Includes text ("Understanding Microcomputers") plus high-quality cassette. For the beginner just starting in microcomputers. Covers all the basics quickly, easily and enjoyably. Companion tape includes chapter-by-chapter synopsis of the book. Great new idea for self-study. $14.95 plus $1

TAKE MY COMPUTER . . . PLEASE!

An uproariously funny book about the true-life misadventures of author Steve Ciarcia and his computer's inability to cooperate. Hardcover. $5.95 plus $1

MICROCOMPUTER POTPOURRI

A pocket-sized reference for the beginner. Has a really great glossary covering all the jargon. Full digest on understanding microcomputers. $3.95 plus $1

CALCULATING WITH BASIC

Use your computer to calculate home mortgage payments, interest rates, payback periods and more. Complete routines already worked out for problem solving using BASIC language. Also includes mathematics, finance and statistics, mechanical engineering and electronics. $8.95 plus $1

PERSONAL INFORMATION MANAGEMENT SYSTEM

When you're in business you've got a personal stake in how information is managed. That's because information is your key to success. PIMS will allow you to unleash the power of a microcomputer - and you don't have to become a programmer first. Use a computer for accounts receivable . . . accounts payable . . maintain inventory records . . .run a mailing list . . .keep track of credit charges. These are just a few of the many things you can do with PIMS. It is a ready-to-use data base management program for computers like the Commodore PET and Radio Shack TRS-80. You can define and construct your own data bases. Each record can contain up to ten fields You can search, list and sort. There's even a command that let's you sum columns of numbers PIMS provides a valuable introduction to data base management. Complete source listing. $11.95 plus $1

APL - AN INTERACTIVE APPROACH

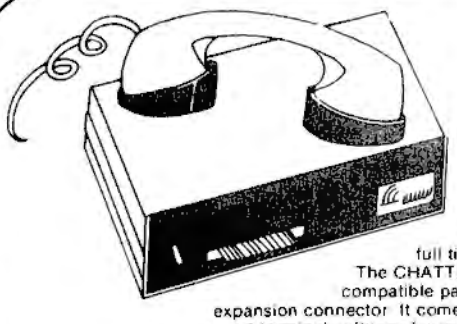
This revised second edition of APL - An Interactive Approach has been renamed to reflect the fact that several versions of APL are currently being offered. In recognition of APL's growing use in business applications, more examples have been included, and the body of the text itself has undergone a modest shift in orientation toward commercial uses of APL.

Additional functions and features now available in both the IBM and Scientific Time Sharing implementations have been included in this edition, and the chapters on workspace management and function definition have been substantially rewritten providing additional graphic aids to the student. Where appropriate, sections have been included on distinctive features of the IBM 5100 Computer.

For this edition, nearly all the example functions in the text have been placed in a workspace named 1 CLASS. If your APL system lacks this workspace, it may be obtained from Scientific Time Sharing Corporation. $16.95 plus $3.

LITTLE BOOK OF BASIC STYLE

Ideal reference for BASIC programmer, junior high to research scientist. Indexed, illustrated, 151 pages. By John M. Nevison $5.95 plus $1



The Chatterbox

A TRS-80 Interfacing Alternative

The CHATTERBOX 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 barebones 4K TRS-80* into a full time-sharing terminal.

The CHATTERBOX includes built-in programmable 50-19200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic originate modem, and a spare TRS-BUS expansion connector. It comes complete with power supply, ribbon cable and connector, user's manual, and smart terminal software for immediate operation. When the modem is in use, the complete data conversion is automatically routed to the serial output port where it can be logged on a printer.

The CHATTERBOX is the only peripheral needed to allow a TRS-80* to communicate with time-sharing systems such as MICRONET and the SOURCE.

It is completely hardware and software compatible with existing TRS-80* products and connects either to the keyboard connector or screen printer port on the RS Expansion Interface. Features: Full 8-bit parallel port, RS-232-C serial port (up to 19,200 baud), Acoustic modem, TRS-BUS connector for future expansion. Connects to Keyboard or E.I. Includes terminal software, Users manual, Power supply. $259.95

PRINTERS

	LIST PRICE	OUR PRICE
Centronics 730	$795.00	$749.00
Centronics 730-3	$895.00	795.00
Centronics 737	$995.00	$869.00
Centronics 779	$1395.00	$1095.00
Centronics 779 w/lower case	$1595.00	$1195.00
NEC 5510 SpinWriter	$3195.00	$2595.00
NEC 5520 SpinWriter	$3395.00	$2895.00
NEC 5530 SpinWriter	$3195.00	$2495.00
NEC Tractor-Feed Option	$249.00	$225.00
LRC 7000[8] (64-col.)	$405.00	$299.00
LRC 7000[8] (40-col.)	$389.00	$289.00
Okidata Microline-80	$800.00	$699.00
Tractor-Feed Option	$140.00	$129.00
RS-232-C Option	$299.00	$279.00

LRC to TRS-80	$20.00
LRC to PET, IEEE	$59.00
LRC to RS232C, male or female	$65.00
730 or 737 to TRS-80	$29.00
NEC or 779 to TRS-80	$35.00
RS-232-C to RS-232-C, male to male	$24.95

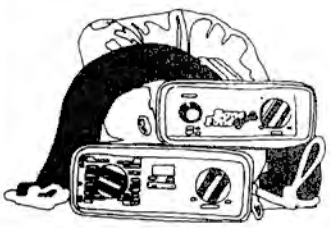
GENERAL INTEREST

	LIST PRICE	OUR PRICE
BSR System X-10 Home Controller	$124.95	$109.95
BUSY BOX, TRS-80	$114.95	$99.95
BUSY BOX, S-100	$119.95	$114.95
AC-SFK-31 Line Filter	$24.95	$19.95
ISO-2 Line Filter & Isolator	$56.95	$49.95
ISO-2/CBS Line Filter-Isolator	$70.95	$59.95
ISO-7/CB Super Filter-Isolator	$146.95	$99.95
Mini-Flex Diskette File	$24.95	$19.95
CASIO C-80 Calculator Watch	$49.95	$44.95
BONE FONE	$69.95	$56.95
LOGOS-9 Printing Calculator	$99.95	$79.95

SPECIAL

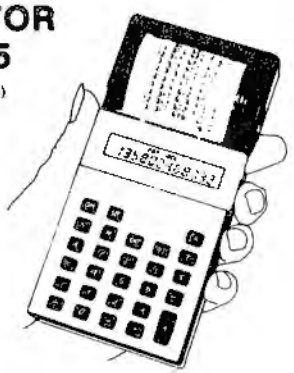
CASIO C-80 CALCULATOR WATCH $44.95

(plus $2.50 postage and handling)



BONE FONE $56.95

(plus $2.50 postage and handling)



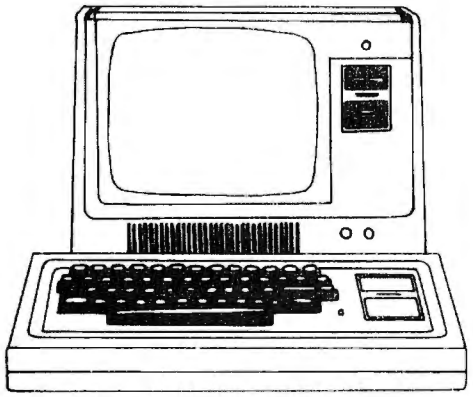
OLIVETTI LOGOS-9 PRINTING CALCULATOR $79.95

(plus $2.50 postage and handling)

TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

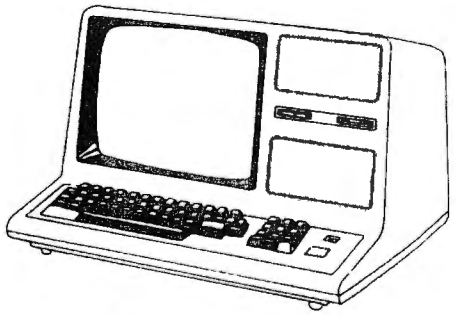
MasterCard VISA

HARDSIDE™

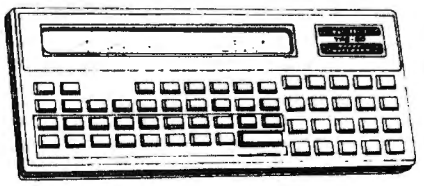


MODEL I $699

	LIST PRICE	OUR PRICE
Model-I, Level-II, 4K	$649.00	$619.00
Model-I, 16K no Keypad	$768.00	$669.00
Model-I, 16K w/Keypad	$849.00	$729.00
Model-II, 64K RAM	$3899.00	$3799.00
Model-III, 16K RAM	$999.00	$929.00
Model-III, 32K Dual Disk	$2495.00	$2299.00
Pocket Computer w/Interface	$298.95	$269.00
TRS-80 Color Computer	$399.00	$359.00
TRS-80 Color Computer Expanded	$599.00	$519.00
COMM-80 Interface	$179.95	$159.95
CHATTER BOX Interface		$259.95
DISK-80 Interface	$349.95	$329.95
Expansion Interface, no RAM	$299.00	$279.00
Expansion Interface, 16K RAM	$418.00	$339.00
Expansion Interface, 32K RAM	$537.00	$389.00
RS-232-C Board	$99.00	$89.00

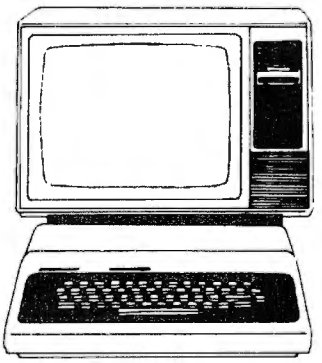


MODEL III $929



POCKET COMPUTER $269

	LIST PRICE	OUR PRICE
TRS-232 Printer Interface		$59.95
16K Memory Kit, TRS-Keypad	$119.00	$59.00
16K Memory Kit, TRS-Exp. Int.	$119.00	$59.00
Upper/lower Mod Kit	$59.00	$24.95
Video Reverse Kit		$23.95
CPU Speed-up Kit		$24.95
Data Dubber		$49.95
Percom Electric Crayon, w/cable		$279.95
TRS-80 Dust Cover (3pc set)	$9.95	$7.95
TRS-80 Computer Case	$109.00	$99.95
TRS-80 Monitor Case	$84.00	$84.00



COLOR COMPUTER $359

	LIST PRICE	OUR PRICE
Percom, TFD-100, 40-track	$429.95	$399.00
Percom, Dual TFD-100 Drives	$849.00	$799.00
Percom, TFD-40, 40-track	$399.95	$379.00
Percom, TFD-200, 77-track	$675.00	$629.00
Hardside, 40-track Disk Drive	$399.00	$359.00
Percom Data Separator		$29.95
Percom Extender Card	$15.95	$15.00
2-Drive Cable	$29.95	$29.00
4-Drive Cable	$39.95	$39.00

TERMS:

TERMS: Prices and specifications are subject to change. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders; Personal checks accepted (takes 3 weeks to clear). HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders OVER $100.00. On all orders over $100 a $2.50 handling charge must be added. COD orders accepted (orders over $250 require 25% deposit) there is a $5.00 handling charge. UPS, Blue Label, and Air Freight available at extra cost.

HARDSIDE™

TO ORDER TOLL-FREE:

1-800-258-1790

(In NH call 673-5144)

MasterCard VISA

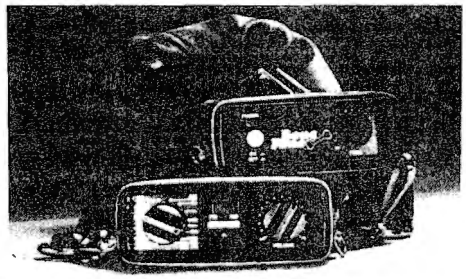
Bone Fone

The BONE FONE is a miniaturized AM/FM stereo system that drapes over your shoulders like a scarf. The BONE FONE gives new meaning to the word portable. You don't carry a BONE FONE around with you. You wear it. Ski with it, cycle, jog, or just take a leisurely stroll through the woods with it. It's the hands-free stereo that brings beautiful music to every indoor or outdoor activity without disturbing those around you and without covering your ears.

The BONE FONE stereo is covered with a sleeve made of Lycra Spandex - the same material used to make expensive swim suits, so it's easily washable. You simply remove the sleeve, dip it in soapy water, rinse and let dry. The entire system is also protected against damage from moisture and sweat making it ideal for jogging or bicycling.

The BONE FONE has been selling nation-wide for $70. HARDSIDE's special price is ONLY $56.95 plus $2.50 handling (we pay the UPS charges).

9902 ..... $56.95



TO ORDER
TOLL FREE
1-800-258-1790
(In NH call 673-5144)

THE WATCH THAT REPLACES EVERYTHING.

In fact it even replaces the age-old myth that a timepiece loaded with talent has to be expensive.

The Casio C-80 has a list price of $50. It has been advertised nationally for $75. HARDSIDE's special price is only $44.95 plus $2.50 handling (we pay the UPS charges).

A CALCULATOR MADE FOR REAL FINGERS

Here's the first calculator chronograph that can be operated easily with your fingertips. No need to use a stylus or pen; the keys are far enough apart that even the broadest fingers can work them.

Add, subtract, multiply, divide - even perform chain calculations up to eight places, plus decimal. Numbers appear in crisp, liquid, crystal digits.

STOPWATCH, 2 TIME ZONES - AND MORE

You get 1/100 of a second accuracy, lap times, and up to 24 hours of track-style timekeeping.

You also get two time zones; the second zone is set for 24-hour time. Hours, minutes and seconds are displayed with reliable 15-second-a-month quartz precision.

Press another button and the month and date appear. And press another for built-in night light.

FEATHER-LIGHT, YET RUGGED

The 1-3/4 oz. case and band are constructed of high-impact black plastic. And so lightweight and comfortable you may not realize that you are wearing it. This Casio chronograph is rugged, resists scratching, and is water-resistant.

99-01 ..... $44.95

TOLL FREE ORDERS:

1-800-258-1790

(in NH call 673-5144)

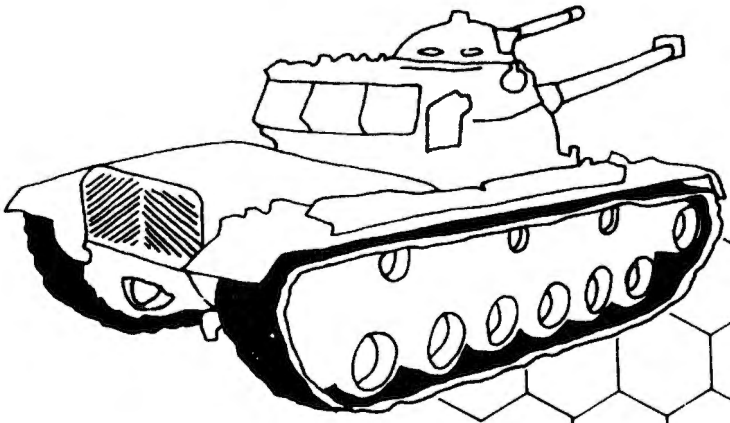
HARDSIDE™

6 South Street, Milford, NH 03055

Three from Potkin.

Wargamers delight!

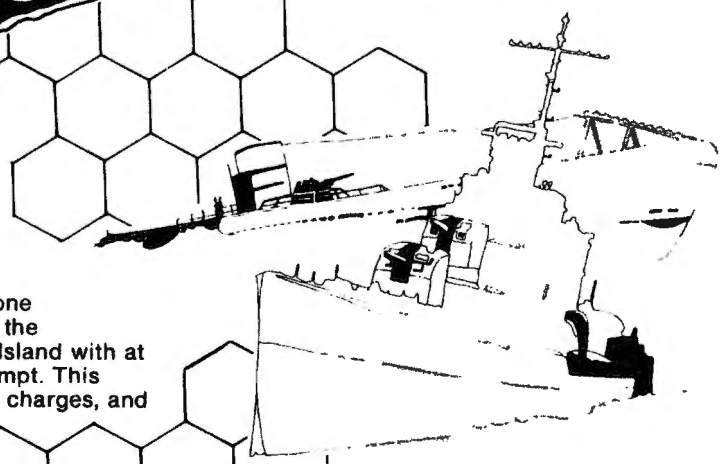
1). Kriegspiel II.



A much improved two-player version of the original. Kriegspiel II is a war-gamer's delight. Choose the number of mountains (up to 200) and pick a scenario from the 9,999 possible, and then watch the computer set up the pieces, towns, mountains and a river. To win, you must enter the capital city of your opponent or reduce his fighting strength to below half of your own

S-80 Level II, 16K cassette $14.95

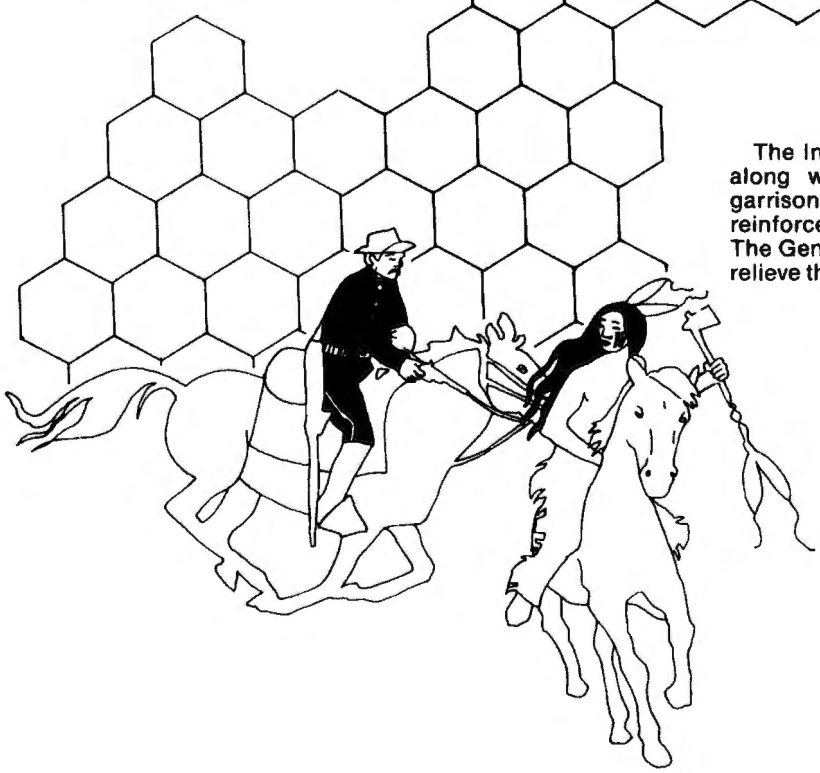
2). Up Periscope



The author of the popular Kriegspiel II has done it again. This time the action takes place at sea with one player controlling the submarines while the other attempts to sail around RADSHA Island with at least three of his fleet surviving the attempt. This realistic wargame includes sonar, depth charges, and torpedoes.

S-80 Level II, 16K cassette $14.95

3). Warpath.



The Indians are on the warpath! The Chief, along with 24 braves, is out to take the garrison at the fort, or at least to stop reinforcements from entering the stockade. The General, with his 14 troopers, is trying to relieve the garrison before the flag is captured. The player determines the scenario through placement of boulders that provide both shelter and obstacles. Favorite scenarios may be replayed.

S-80 Level II, 16K cassette $14.95

RAMWARE™

CALL TOLL-FREE
1-800-258-1790
In NH call 673-5144

software you can rely on.

PERSONAL/BUSINESS

SMALL BUSINESS BOOKKEEPING (S-80)
Roger Robitaille Sr
16K Level II $24 95
Extended version 32K disk
Without Journal $29 95

PERSONAL FINANCE
Lance Micklus
Level II, 16K $9 95

GAMES

X-WING II (S-80)
Chris Freund
Level II, 16K $9 95

INVASION (S-80)
Chris Freund
Level II, 16K $9 95
32K Disk $14 95

SPACE BATTLES (S-80)
Level II, 16K tape $14 95
Level II, 32K disk $19 95

SNAKE EGGS (S-80)
Level II, 16K $14 95

LIFE TWO (S-80)
Level II, 16K $14 95

BEE WARY (S-80)
Level II 16K $14 95

ANDROID NIM (S-80)
Level II, 16K $14 95

AMAZIN' MAZES (S-80)
Robert Wallace
Level II, 16K $9 95

STAR TREK III.5| (S-80)
Lance Micklus
Level II 16K $14 95

UP PERISCOPE (S-80)
Ron Potkin
Level II 16K $14 95

WARPATH (S-80)
Ron Potkin
Level II 16K $14 95

MEAN CHECKERS MACHINE (S-80)
Lance Micklus
Level II, 16K ..... $ 9 95
Disk $14 95

KRIEGSPIEL 2 (S-80)
Ron Potkin
Level II, 16K (cassette only) $14 95

EDUCATIONAL GAMES

9 GAMES FOR PRE-SCHOOL CHILDREN
George Blank
Level II, 16K $9 95

UTILITIES

UTILITY (S-80)
Nepenthe Publications
32K and 48K disk systems $29 95

SUPER SIMON (S-80)
George Blank
Level II, 16K (disk or cass) $9 95

STAD (Symbolic Trace and Debug) (S-80)
Paul van der Eijk
16K, 32K and 48K versions
on one cassette $24 95

DYNAMIC DATA BASE (S-80)
Ken Knecht
Disk 32K $39 95

MICRO TEXT EDITOR (S-80)
Don Coons
Level II 4K $9 95

FLOPPY DISK DIAGNOSTIC (S-80)
Dave Stambaugh
16K to 48K, 1 to 4 disk drives $24 95

Z80 ZAP (S-80)
ORG-TEX Industries
16K Disk $29 95

TINY COMP (S-80)
Dave Bohike
Tape version $19 95
Disk version $24 95

RPN CALCULATOR (S-80)
Russell Starkey
Level II 16K $9 95

KEYBOARD 80 (S-80)
John Adamson
Level II, 16K $9 95

SPOOLER (S-80)
Miles Smither
Disk $24 95

KVP (S-80)
Lance Micklus
16K Level II Tape $14 95

INVENTORY 'S' (S-80)
Roger Robitaille
Level II 16K $24 95
32K Disk $59 95

ST80D (S-80)
Lance Micklus
32K Disk systems $79 95

ST-80 UC (S-80)
4K Level II Cassette $24 95

BASIC CROSS REFERENCE
RX-X-REF-RENUMX
Lance Micklus
RX (Disk, 32K) 1 disk system $24 95
XREF (Level II version
16K on Cassette $19 95
RENUMX (XREF w/renumber and rescue)
Level II on cassette $24 95
RENUMBER (separately) $7 95

LANGUAGES

APL-80 VERSION 3 0 (S-80)
Phelps Gates
32K single disk $39 95
16K Level II cassette $14 95

ADVENTURES

ADVENTURES ON TAPE
Scot Adams
Count, Pirates Cove, Fun House, Pyramid of Doom, Strange Odyssey, Ghost Town, Mission Impossible, Land, VooDoo $14 95

PATHWAYS THRU THE ROM
Blank, Fuller, Hartford,
Phillipp and Richardson ..... $19 95 + $1

3D (Apple 2)
Mark Pelczarski
A Rom Disk 48K $29 95

MAGIC PAINTBRUSH
Mark Pelczarski
Disk 32K $29 95

ANDROID NIM
Disk/24K $17 95

Screen Protect For The Mod II

James W Crocker
Technical Editor

Many of you have probably noticed that certain programs on the Model II seem to "protect" part of the screen from scrolling. Up till now, you've probably assumed that this was some sort of devious magic that sprang full grown from the head of Zeus. Here is a plain and simple way to use this built-in feature of the Mod II, along with an introduction to the wonders of the Supervisory Call (SVC).

On most large computers, the operating system contains all the I/O drivers. The individual programmers are neither expected or allowed to perform their own I/O. Any time a program desires to output to the disk drives or printer or magnetic tape units, a Supervisory Call is performed. This allows the system to keep track of what's on each disk, tape, and printer. When the Mod II was designed, similar capabilities were included in the DOS.

There are a total of 256 possible SVC's. The ones already programmed for you include screen, disk, printer, and RS232 I/O, as well as keyboard input. A SVC call requires that the number of the call be loaded in the A register, and parameters be passed in the other registers. Some SVC's require no parameters, such as SVC 36, JP2DOS. A SVC is accessed via the RST 8 instruction. Therefore, to return control to DOS, we would load the A register with 36 (decimal), and perform a RST 8.

One of the pre-programmed SVC functions is the scroll protect, SVC 27. When a RST 8 instruction is performed with 27 (1BH) in the A, then the number of lines specified in the B register (in the range 0-22) will be protected from scrolling until either another SVC 27 is encountered, or a CLS command is executed, either from DOS READY, BASIC, or by printing a character 1BH via the SVC 8 (which will print one character, the ASCII of which is in the B).

For our purposes, we wish to be able to access the SVC 27 from BASIC. We decided that the USR function would be the best way to do so. Therefore, the first thing we did was to create a machine language program high in memory that would accept a parameter from BASIC, and then perform the RST 8. To do this, enter DEBUG. From TRSDOS READY, type "DEBUG {ON}". The computer should respond with something to the effect of "DEBUG IS NOW ON". Now type "DEBUG". This should result in the debug program being loaded and executed.

The first thing we want to do is display the addresses that we want to change. To do this, type M (no [*ENTER*] needed) this will get the prompt "A=....". Debug is looking for an address. Type F2C0 (again, no enter required) and the program should display the addresses and data starting F2C0. You should have zeroes from F2C0 to F2FF. The data from F300 on is part of the DEBUG program. If you don't have all zeroes, go back to DOS (hit the [*ESC*] key, then type S) and execute the CLEAR statement and start all over.

Now that we are looking at the correct area of memory, we are going to write a short machine language program. To modify memory, we first must be in the M mode (the "A=..." prompt should be on the screen). Hit the F1 key. This will put the cursor at the first byte of the displayed area. We are now in the modify memory mode. Type the following:

FE 02 C0 46 3E 1B CF C9

and hit the F2 key. This makes the changes permanent.

Now that our program is in memory, we want to get it onto the disk. Get out of the memory examine mode by hitting the [*ESC*] key. Then hitting S will get us back to TRSDOS READY. Now type:
DUMP SCRNPROT/USR START=F2C0, END=F2C7, TRA=F2C7, RORT=R
and [*ENTER*]. This will dump our program out to the disk. The program we just wrote looks like this:

```
CP    2       ;THE A REG CONTAINS THE
                VARIABLE TYPE
RET   NZ      ;RETURN IF NOT INTEGER
LD    B,(HL)  ;GET NUMBER OF LINES TO PROTECT
LD    A,27    ;FOR SVC 27
RST   8       ;PERFORM THE CALL
RET           ;RETURN TO BASIC
```

For a little insight on the whys and wherefores, take a look at the Mod II's BASIC manual, pages 3/144 and 3/145. When a USR call is made, the A register contains the variable type used for the argument. Because of the way numbers are stored in Microsoft BASIC, we want to make sure we are getting an integer for the argument. This is the reason for the CP 2 and RET NZ.

The USR also loads the HL with the address of an Argument Storage Area. If the argument is an integer, the value of (HL) should be the least significant byte of the argument. Since SVC 27 wants this number in the B register, we LD B,(HL). Finally, a LD A,27 tells TRSDOS what SVC we want done, the RST 8 actually performs the function, and RET takes us back to BASIC.

The BASIC program to access this function couldn't be much simpler. All you need is three lines:

```
10 CLEAR1000,62151
20 DEFINTA-Z:SYSTEM"LOAD SCRNPROT/USR"
30 DEFUSR9=&HF2C0:A=USR9(10)
40 END
```

Line 10 CLEARs 1000 bytes of string space, and sets memory size at 62151 (you may want to change this if you have other routines that require memory space to be set aside.) Line 20 assures that all variables will be integers, and passes the command to DOS to load the program we just created. Line 30 DEFines our USeR routine address to be F2C0 hex, and executes the USR function, telling the USR routine that we want to set aside 10 lines.

That is all there is to it! You now may LIST your program, or execute several PRINT statements, and the top 10 lines of the screen will not even budge. The 12 lines below it will scroll normally, except that it will act like the 11th line is the top of the screen.

You should note that we haven't put much error trapping here. If you give a value greater than 22 or less than zero to the USR routine, it will merrily pass that value along to the SVC routine (which will summarily ignore it). If you try to pass a non-integer argument to the USR, it will simply return without doing anything. The goal of this article was not to develop the perfect program, but to give you an introduction to the SVC routines and their uses, and to give you a useful, workable example. The descriptions of the SVC routines and what they do begins in the TRSDOS section of the Mod II manual on page 4/9. With a little care, you can do all sorts of wonderous things using these routines. ●

BUSINESS COMPUTING

Check Write

A personal finance system

For Model I, Level II 16K and up.

R.A. Shmina

14030 Arnold
Detroit, MI 48239

Part 2 of a 2 Part Series

Before we examine SUMMARY/ONE and /TWO, let's review the operation of the system to the point of where we would desire to run "SUMMARY" and see what form our data is in at that point.

Let us load and run "CKWRTDAT/ONE". *(See installment #1, 80-U.S. Journal Sept/Oct 80)* Turn on your printer, and let's go.

First, we are asked to enter the date. Do this in the form of MMDDYY where MM is the two digit representation of the month, DD is the two digit representation of the date, and YY is the two digit representation of the year (example: February 1, 1980, would be 020180). Next we are asked to enter the complete filespec name. I use "NEWCHKS/AYY" and progress alphabetically. This gives a possible 26 separate identifications which is more than enough for average use. Of course, you can use any filespec you desire, if you don't like this one. Now we are requested to enter the actual data using the format Date, Check No., Amount, Payee, and Budget Code. Enter the data and separate each item by a comma and press ENTER after you have entered the five items requested. I use four digits for the date (MMDD). You can use six digits (MMDDYY) if you desire. The check number is self-explanatory. The amount is limited to six digits ($9,999.99) by the portion of this program which prints the checks. This can be changed, of course, but for the average user..... The budget code can be any alphanumeric designation you create. The program limits the length to four characters. For some examples, I use 'MILK' for Bordens, 'FOOD' for A & P, 'EXEC' for Execucharge , 'WAT' for water bill, 'GAS' for the gas bill, 'CAR' for any car expenses, 'ELEC' for the electric bill, etc. I also have a small cottage up north and use 'UN' for any of its expenses.

This gives you an idea of what you can do. Type carefully, but if you make an error, you will have a chance to correct it later. You should enter all checks manually written from the time of the last session, as well as all the checks you desire to write at this session. After this data is entered, you type 0,0,0,0,0 to exit this mode. Now your printer will print out a list of the checks you have just entered. Upon completion of the printing, the video will request any corrections to be made. If you have detected an error, answer yes. Let's assume we have errored in the amount of the third entry. When we are requested to enter I,J, & T$, we type 3,2, and the correct amount and press "ENTER". I is the actual row number of the entry (3 in this case). J is the column number where 0=Date, 1=Ck. No., 2=Amount, 3=Payee, and 4=Code. T$ is the actual correct data we desire. Also type 0,0,0 and ENTER to exit this mode. Now the corrected list will be printed. If it is okay this time, respond 'no' to "Any Corrections Necessary?". The next requested entries merely update the balance for your immediate knowledge rather than waiting until you run CKWRTDAT/TWO.

The video now requests if you desire another copy of this list for your records. Respond accordingly and press ENTER. Don't forget to advance the paper, if you do ask for another copy. You are now asked if you are on tape or disk storage. Enter accordingly. The program will now enter the data you just saved and you can see on the screen if it's okay. If it's not, press BREAK then type "GOTO480" and ENTER and try again. You would not be the first person who forgot to advance the tape beyond the leader. Assuming a good load, we are now prompted to enter the blank checks in the printer. We also enter the current check number and hold on. All our checks from the current number on will be printed. Naturally the format of printing (Lines 1010 - 1310) must be changed to accommodate your own check. The program run is now complete and we have a "NEWCHKS/A" file on tape or disk.

At any given point in your own system, you will have some outstanding checks (not yet cancelled by the Bank). We can use CKWRTDAT/ONE and enter these outstanding checks and create our file of "OUTSTNDG/CXX", which we need for running CKWRTDAT/TWO. All we have to do is enter the proper filespec (In this example, let us use "OUTSTNDG/C00"). If you're on a tape system, use a separate tape. When CKWRTDAT/ONE asks if we are on a tape or disk system, we must press BREAK and type "SUM=X" where X equals our current check book balance (assuming, of course, we have reconciled the account with the Bank). After typing "SUM=X", press ENTER, then type "CONT" and press ENTER again. The program will resume and save our file on disk or tape.

We now have an "OUTSTNDG/C00" file and a "NEWCHKS/A" file; and are now ready to run CKWRTDAT/TWO. Turn on your printer and let's go.

Upon running, we are immediately requested to update the filespecs. If you are on tape, you can ignore this part. All you have to do is modify the program as previously described in the line by line analysis and insert the proper cassette at each point. The program already has a built-in stop to allow you to do this. But now, back to the disk users. You must update by editing each filespec. In our example, we would edit Line 300 to "OUTSTNDG/C00", Line 470 to "NEWCHKS/A", Line 1120 to "OUTSTNDG/C01" and Line 1030 "CANCECKS/FEB" (assume this is a February program run). Now rerun the program and skip the file update request. Next we are requested to enter the date. You can use any format you desire here. I prefer the written-out method, as it is easier to read on the printout.

We are now prompted to enter the proper cassette or disk. Press ENTER and "OUTSTNDG/C00" data is input to program. Hardcopy "audit trail" output is generated. Now change cassettes or disk again for the input of "NEWCHKS/A" data. Actually, you lucky people on disk don't have to change disks because you can get an awful lot of files on one disk -- I'm just saying this so the guys on cassette systems don't feel so bad. Again, hardcopy output is generated. We are now requested to cancel the same checks that have cleared the Bank. You merely enter the check number

of the cancelled check and press ENTER. Enter a zero (0) to escape this mode. We now get a listing of the checks we have just cancelled. Now we are requested to enter any deposits made since the last session. Follow the indicated format. A list of deposits is then generated. The current balance is calculated and printed. Then the list of outstanding checks is printed. We are now prompted to save the cancelled checks. If you are on a cassette system, you must load the proper cassette -- if you are on a disk system, you're all set with the disk that's already being used -- assuming there is enough space left. Either system, you've always got to be on guard to insure you have enough media for the data. You only have to run short once, then you'll remember forever. After this, the program saves the cancelled checks data.

On a cassette system, the best way to do this is to get as many files as you can on one tape. Keep a record of the Index settings and always advance to the end of the data already on tape, then leave a little gap (about five units) and start again. The final programs "SUMMARY/ONE and /TWO" use this data (DATAFILE file). For cassette users, one long tape with many files that will fit or many tapes with one file each (your choice) can be used. For disk users, each "CANCECKS/XXX" is "Appended" to the previous file. The master file is then renamed to "DATAFILE/CYY" where YY is a two digit representation of the year. A new master file is made for each year, obviously.

Meanwhile, back to the finish of the run of "CKWRTDAT/TWO". We are prompted to enter the Bank balance and any deposits not credited yet by the Bank. We obtain the Bank balance from the monthly Bank Statement. Any deposits *not credited by the Bank* are entered *even if you have entered them earlier* in the program! You then obtain a listing of the calculations done to reconcile the system. The item "Correction to Program Balance" is the invisible cushion I keep in my system. In other words, the checkbook balance is always greater than indicated by this amount. Hopefully, it keeps us from spending when the balance gets low -- but on the other hand, I always know that it's there..... The program run is now complete.

Now it sometimes happens we find we have an error or we desire to change the budget code assignment of an item on an existing file. A program titled "CORRECT/ION" is provided for this purpose. It will read in any file, allow you to make corrections, and resave the file. Of course, you have to adapt, by editing, certain parts of the program to accommodate the format of the file with which you are working.

After a number of sessions when we have accumulated a master file of cancelled checks, we are ready to run the "SUMMARY" programs. "SUMMARY/ONE" gives a video and printed output which shows the dollar total for all the budget codes, while "SUMMARY/TWO" will give a complete listing (both video and printer) for any code of payee selected. These programs are set up for disk files that are too large for the available memory. The line by line analysis shows what revisions are necessary for cassette usage. Either of these programs use the master file of all "DATAFILE" files from each program which have been "appended" together. For disk systems, it can be used as is and the necessary modifications for cassette use are contained in the Remark statements. You will really appreciate this system of programs at tax time when you make a yearly summary.

Okay now, if you have read this far you are still interested, so sit down and type the programs in your computer

If any of you have any problems or questions regarding these programs, drop me a line, and I'll try to help.

SUMMARY/ONE ANALYSIS

This program will summarize a DATAFILE of any size. It does not depend upon the amount of memory in your system.

Line 100-1030 Since this program is written for disk operation these lines contain the revisions which must be made for cassette operation.

Line 1050 Description of program operation.

Line 5990 Clears string space; defines A and D as strings; dimensions array A to six elements (0 - 5); dimensions S & D$ arrays to 31 elements. If you elect to add more budget codes, this value must be changed.

Line 6000-6008 Fills array D$(N) with assigned budget codes from the data statement in Line 8000. Change Line 8000 to your own budget codes.

Line 6025 Reassigns N=1. I like to use N as a counter and its previous value was left @ 31. This reinitializes N=1. We also OPEN DATAFILE. Remember this is all our separate CANCECKS files appended together.

Line 6030-6400 One large loop to examine one item of data input and assign its value to the proper array element. All of these lines can be changed to your own assigned budget codes. These codes must agree with the data statement in L 8000.

Line 6500 This is the key line, if you are on a cassette system. When the "GOTO6030" is deleted, this line stops the program and allows you to read other cassettes or other files, if you have more than one file on a cassette.

Line 9010-9140 Routine for video and line printer output.

SUMMARY/TWO ANALYSIS

This program will give a complete list of all items in DATAFILE according to the payee or code selected. Size of DATAFILE is not dependent upon memory size.

Line 520-530 Contains the necessary revisions for cassette operation.

Line 900 Clears string space; defines A as a string; initializes W=0,X=0,Q=0.

Line 1000-1900 Offers option to list items from DATAFILE by payee or budget code. If you select payee, enter enough letters to identify payee. You will obtain a listing of every item in DATAFILE whose first letters equal what your response was. If you elect budget code, you will get a listing of every item in DATAFILE whose code matches the code you entered

Line 2000-2010 Opens file and inputs first item which is a counter.

Line 2020-2080 Loops through a file looking for a match to selected code or payee. Notice use of flag Q to control which comparison is made, depending upon selection made (code or payee). If a match is found, the line is printed and the amount is added to the running total.

Line 2100-2130 Again for cassette operation the line allows any number of separate files to be read whether or not they are on one cassette or many cassettes.

Line 5000-6000 Video and line printer output routine. I let the listing erase the screen. It gives you something to watch while the program is running. It also shows you how to list a given number of items on the screen and stop the program until you have examined them and pressed ENTER. Notice that Line 2100 uses PRINT@ to display message and Line 2010 erases the message. I like this, hope you do, too.

CORRECT/ION ANALYSIS

This is a program to read in any data file created by this system and will allow you to make any needed corrections. Instructions for its use are contained in the program. As written, it will accommodate tape or disk systems. ●

```
5 REM ******** S U M M A R Y / O N E **************
7 REM ********       VERSION 2.3       **************
1000 REM INSTRUCTIONS FOR CASSETTE OPERATION:
1010 REM 1.  REPLACE L#6025 W/L#6025 N=1:INPUT"PRESS ENTER WHEN
          CASSETTE IS READY (PLAY)";Z$
1020 REM 2.  EDIT L#6030 TO: INPUT#-1,I:PRINT I
1030 REM 3.  DELETE 'GOTO 6030' FROM L#6500
         4.  DELETE L#6330
         5.  EDIT L#9000 TO READ: GOTO 9010
         6.  EDIT L#6040 FROM INPUT#1,....TO INPUT#-1,.....
1040 :
1050 REM THIS PROGRAM WORKS FROM DISK OR TAPE AND DOES NOT USE
       MEMORY TO STORE THE FILE.  THE OUTPUT IS A LIST SHOWING
       BUDGET CODE AND ITS TOTAL SUM.
1060 CLS
2000 PRINT"        ***************************************************
    *****"
2010 PRINT"        ***
      ***"
2020 PRINT"        ***          S U M M A R Y / O N E
      ***"
2030 PRINT"        ***                  R.A.SHMINA
      ***"
2040 PRINT"        ***
      ***"
2050 PRINT"        ***************************************************
    *****"
5990 CLEAR 2000:DEFSTR A,D:DIM A(5),S(31),D$(31)
6000 N=1
6005 READ D$(N):PRINT D$(N);" ";:IF D$(N)="-1" THEN 6010
6008 N=N+1:GOTO 6005
6010 PRINT:PRINT"NOW READING THE FILES...."
6025 N=1:OPEN"I",1,"DATAFILE"
6030 INPUT#1,I:PRINT I;" ";
6035 FOR N=1 TO I
6040 INPUT#1,A(0),A(1),A(2),A(3),A(4),A(5)
6045 IF A(4)="DEP" THEN S(1)=S(1)+VAL(A(2))
6050 IF A(4)="TAX" THEN S(2)=S(2)+VAL(A(2))
6060 IF A(4)="DON" THEN S(3)=S(3)+VAL(A(2))
6070 IF A(4)="MED" THEN S(4)=S(4)+VAL(A(2))
6080 IF A(4)="DENT" THEN S(5)=S(5)+VAL(A(2))
6090 IF A(4)="INS" THEN S(6)=S(6)+VAL(A(2))
6100 IF A(4)="MBT" THEN S(7)=S(7)+VAL(A(2))
6110 IF A(4)="ELEC" THEN S(8)=S(8)+VAL(A(2))
6120 IF A(4)="GAS" THEN S(9)=S(9)+VAL(A(2))
6130 IF A(4)="WAT" THEN S(10)=S(10)+VAL(A(2))
6140 IF A(4)="TUIT" THEN S(11)=S(11)+VAL(A(2))
6150 IF A(4)="D&S" THEN S(12)=S(12)+VAL(A(2))
6160 IF A(4)="GIFT" THEN S(13)=S(13)+VAL(A(2))
6170 IF A(4)="PALQ" THEN S(14)=S(14)+VAL(A(2))
6180 IF A(4)="HUD" THEN S(15)=S(15)+VAL(A(2))
6190 IF A(4)="CAR" THEN S(16)=S(16)+VAL(A(2))
6200 IF A(4)="JAC" THEN S(17)=S(17)+VAL(A(2))
6210 IF A(4)="SEAR" THEN S(18)=S(18)+VAL(A(2))
6220 IF A(4)="MC" THEN S(19)=S(19)+VAL(A(2))
6230 IF A(4)="VISA" THEN S(20)=S(20)+VAL(A(2))
6240 IF A(4)="AMEX" THEN S(21)=S(21)+VAL(A(2))
6250 IF A(4)="EXUC" THEN S(22)=S(22)+VAL(A(2))
6260 IF A(4)="CITI" THEN S(23)=S(23)+VAL(A(2))
6270 IF A(4)="MISC" THEN S(24)=S(24)+VAL(A(2))
6280 IF A(4)="FOOD" THEN S(25)=S(25)+VAL(A(2))
6290 IF A(4)="MILK" THEN S(26)=S(26)+VAL(A(2))
6300 IF A(4)="LANE" THEN S(27)=S(27)+VAL(A(2))
6310 IF A(4)="RAS" THEN S(28)=S(28)+VAL(A(2))
6320 IF A(4)="UN" THEN S(29)=S(29)+VAL(A(2))
6330 IF EOF(1) THEN 9000
6400 NEXT N
6500 GOTO 6030:PRINT"ANY MORE FILES TO READ"
6510 Z$=INKEY$:IF Z$="" THEN 6510
6515 IF LEFT$(Z$,1)="N" THEN 9000
6520 IF Z$="Y" THEN 6030
6530 IF Z$()"Y" OR Z$()"N" THEN 6510
8000 DATA DEP,TAX,DON,MED,DENT,INS,MBT,ELEC,GAS,WAT,TUIT,
     D&S,GIFT,PALQ,HUD,CAR,JAC,SEAR,MC,VISA,AMEX,EXUC,CITI,
     MISC,FOOD,MILK,LANE,RAS,UN,-1
9000 CLOSE
9010 CLS:PRINT"CODE    TOTAL            CODE    TOTAL            CODE
       TOTAL"
9020 X=64:FOR N=1 TO 10:PRINT@X,D$(N);:PRINT@(X+7),USING"$###,##
     #.##";S(N);:X=X+64:NEXT
9025 X=86:FOR N=11 TO 20:PRINT@X,D$(N);:PRINT@(X+7),USING"$###,#
     ##.##";S(N);:X=X+64:NEXT
9030 X=108:FOR N=21 TO 31:PRINT@X,D$(N);:PRINT@(X+7),USING"$###,
     ###.##";S(N);:X=X+64:NEXT
9100 FOR N=1 TO 29
9110 IF D$(N)="DEP" THEN LPRINTTAB(35)"DEPOSITS";TAB(45)USING"$
     ##,###.##";S(N):GOTO 9130
9120 LPRINTTAB(10)D$(N);TAB(20)USING"$###,###.##";S(N):ST=ST+S(N
     )
9130 NEXT N
9135 LPRINTSTRING$(60,"-")
9140 LPRINTTAB(10)"TOTAL";TAB(19)USING"$####,###.##";ST
```

```
500 REM ********** S U M M A R Y / T W 0 **************
510 REM **********       VERSION 2.1       **************
520 REM INSTRUCTIONS FOR CASSETTE OPERATION:
        1. EDIT L#2010 TO READ: .....:INPUT#-1,I:PRINT I;" ";
        2. DELETE L#2070
        3. DELETE 'GOTO 2010' FROM L# 2100
530 REM  4. DELETE 'CLOSE' FROM L#6000
        5. EDIT L#2030 FROM INPUT#1, TO INPUT#-1,
        6. NEW L#2000: PRESS ENTER WHEN CASSETTE IS READY
                       (PLAY);Z$
620 PRINT"        ***         S U M M A R Y / T W O
    ***"
630 PRINT"        ***                R.A.SHMINA
    ***"
900 CLEAR 2000:DEFSTR A:W=0:X=0:Q=0
1000 PRINT"DO YOU DESIRE A LIST BY BUDGET CODE OR BY PAYEE ?"
1010 PRINTTAB(20)"ENTER  'C'  OR  'P'"
1020 Z$=INKEY$:IF Z$="" THEN 1020
1030 IF Z$="P" THEN 1200
1035 IF Z$="C" THEN 1100
1040 IF Z$<>"C" OR Z$<>"P" THEN 1020
1100 INPUT"ENTER DESIRED CODE";D$
1110 IF D$="" THEN 1100
1120 GOTO 1900
1200 INPUT"ENTER ENOUGH CHARACTERS TO IDENTIFY PAYEE";D$:Q=1
1900 PRINT"NOW READING THE FILES....."
2000 OPEN"I",1,"DATAFILE"
2010 PRINT@936,"                ":INPUT#1,I:PRINT I;" ";
2020 FOR N=1 TO I
2030 INPUT#1,A(0),A(1),A(2),A(3),A(4),A(5)
2035 IF Q=1 THEN 2060
2040 IF A(4)=D$ THEN GOSUB 5000
2050 IF Q=0 THEN 2070
2060 IF LEFT$(A(3),LEN(D$))=D$ THEN GOSUB 5000
2070 IF EOF(1) THEN 5500
2080 NEXT N
2100 GOTO 2010:PRINT@936,"ANY MORE FILES ?";
2110 Z$=INKEY$:IF Z$="" THEN 2110
2120 IF Z$="Y" THEN 2010
2130 GOTO 6000
5000 REM
5010 PRINT@W,A(0);TAB(8)A(1);TAB(13)USING"$######.##";VAL(A(2));
    :PRINTTAB(24)A(3);TAB(56)A(4);TAB(62)A(5)
5015 LPRINTTAB(0)A(0);TAB(8)A(1);TAB(13)USING"$###,###.##";VAL(A
    (2));
    :LPRINTTAB(24)A(3);TAB(56)A(4);TAB(62)A(5)
5020 S2=S2+VAL(A(2))
5030 W=W+64:X=X+1:IF X=12 THEN 5050
5040 RETURN
5050 W=0:X=0:PRINT:INPUT"PRESS 'ENTER' TO CONTINUE";Z$:CLS:RETU
    RN
5500 PRINT"TOTAL =";TAB(13)USING"$######.##";S2
     :LPRINTTAB(6)"TOTAL";TAB(13)USING"$###,###.##";S2:GOTO 6000
6000 CLOSE:PRINT"END OF PROGRAM":END
```

```
1 REM****    *****  M E R G S O R T / A U T  *****    ****
2 REM**** USED TO LOAD 'CANCECKS' AND SORT FOR ANY ITEM ****
10 CLEAR20000:DEFSTR A,D:DIM A(550,5)
12 INPUT"DISK READY";Z$
15 PRINT"READING DISK ----"
20 I=0:T=0:K=0:S1=0
30 OPEN"I",1,"CANCECKS/C79"
40 INPUT#1,I:PRINT"I=";I
50 GOSUB 1000
100 T=I
125 PRINT"READING DISK AGAIN ----"
130 INPUT#1,K:PRINT "I+K=";I;" + ";K;" = ";:I=I+K:PRINT I
135 GOSUB1000
140 GOTO 100
999 STOP:'****DON'T RUN INTO SUBROUTINE****
1000 FOR N=T+1 TO I
1010 INPUT#1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5):PRINTN;"
     ";
1020 IF EOF(1) THEN 5000
1030 NEXT N
1040 PRINT:RETURN
5000 CLOSE 1:PRINT:PRINT"DO YOU WANT A SUMMARY BY AUTOMATIC COD
    E OR BY PAYEE ?"
5010 PRINT"ENTER 'C' OR 'P'"
5020 Z$=INKEY$:IF Z$="" GOTO 5020
5030 IF Z$="C" GOTO 6000
5035 IF Z$<>"C" AND  Z$<>"P" THEN GOTO 5020
5040 PRINT"ENTER ENOUGH CHARACTERS TO IDENTIFY PAYEE":INPUTD$
5050 S1=0:N=0
5060 FOR N=1 TO I
5070 IF LEFT$(A(N,3),LEN(D$))=D$ THEN GOSUB 7000 ELSE 5080
5080 NEXT N
5090 GOTO 5000:LPRINTSTRING$(3,138)
6000 S1=0:N=1:READ D$
6020 PRINTTIME$:FOR N=1 TO I
6022 IF D$="-1" GOTO 9500
6025 IFA(N,4)=D$ THEN GOSUB 7000 ELSE 6040
6040 NEXT N
6042 IF D$="DEP" GOTO 9000
6045 LPRINTTAB(10)D$;TAB(20)USING"$##,###.##";S1
6050 S2=S2+S1:GOTO 6000
7000 S1=S1+VAL(A(N,2))
7020 RETURN
8000 DATA DEP,TAX,DON,MED,DENT,INS,MBT,ELEC,GAS,WAT,TUIT,
    D&S,GIFT,PALQ,HUD,CAR,JAC,SEAR,MC,VISA,AMEX,EXUC,CITI,
    MISC,FOOD,MILK,LANE,RAS,UN,-1
9000 LPRINTTAB(30)D$;TAB(40)USING"$##,###.##";S1:GOTO 6000
9500 LPRINTSTRING$(80,45):LPRINTTAB(20)USING"$###,###.##";S2:PR
    INT"END";TIME$
```

```
1000 REM ***** ***** C O R R E C T / I O N ***** *****
1010 REM *****  *****    VERSION 3.1     *****  *****
1020 CLS:PRINT"THIS PROGRAM CAN BE USED TO LOAD ANY DATA FILE.
     MAKE
        NECESSARY CORRECTIONS, AND RESAVE THE FILE."
1030 PRINT"FOR DISK USE YOU MUST EDIT L#'S 2000 AND ENTER
        THE CORRECT FILESPEC. YOU MUST ALSO EDIT L#'S 2010 &
        2080 TO ACCOMODATE THE FORMAT OF THE FILE BEING USED.
        (I.E. (DATE,NO.OF ITEMS,SUM) VS (NO.OF ITEMS))"
1040 CLEAR 3000:DEFSTRA,D:DIMA(100,5)
1050 PRINT"ARE YOU USING TAPE OR DISK STORAGE (T/D) ?"
1060 Z$=INKEY$:IF Z$="" THEN 1060
1070 IF Z$="T" THEN 3000
1080 IF Z$="D" THEN 2000
1090 IF Z$()"T" AND Z$()"D" THEN 1060
2000 OPEN "I",1,"FILESPEC"
2010 INPUT#1,D,I,SUM:PRINTD,I,SUM
2020 FOR N=1 TO I
2030 INPUT#1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5):S2=S2+VA
     L(A(N,2))
2040 PRINT A(N,0)";",":A(N,1)";",":A(N,2)";",":A(N,3)";",":A(N,4)";"
     ,":A(N,5)";" "!N
2050 NEXT:CLOSE:PRINT"S2=$"!S2
2060 GOSUB 4000
2070 OPEN "O",1,"XXXXXXXX"
2080 PRINT#1,D!",",!I!",",!S2
2090 FOR N = 1 TO I
2100 PRINT#1,A(N,0)";",":A(N,1)";",":A(N,2)";",":A(N,3)";",":A(N,4)
     ";",":A(N,5)
2110 NEXT N:CLOSE:GOTO 4500
3000 INPUT#-1,D,I,SUM:PRINTD,I,SUM
3010 FOR N=1 TO I
3020 INPUT#-1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N,5):S2=S2+V
     AL(A(N,2))
3030 PRINT A(N,0)";",":A(N,1)";",":A(N,2)";",":A(N,3)";",":A(N,4)";"
     ,":A(N,5)";" "!N
3040 NEXT:PRINTS2:PRINT"S2=$"!S2
3050 GOSUB 4000
3060 PRINT#-1,D,I,S2
3080 FOR N=1TO10:PRINT#1,A(N,0),A(N,1),A(N,2),A(N,3),A(N,4),A(N
     ,5):NEXT N
4000 INPUT"ANY CORRECTIONS NECESSARY? TYPE (YES) OR (NO)";Z$
4010 IFZ$="YES"THEN 4030 ELSE IFZ$="NO"THEN RETURN
4020 GOTO 4000
4030 PRINT"ENTER '0,0,0' TO EXIT THIS MODE"
     :INPUT"ENTER I,J,&T$ (T$=DESIRED CORRECTION)";F,G,T$
4040 A(F,G)=T$
4050 IF F=0 AND G=0 AND T$="0" THEN RETURN
4060 GOTO 4030
4500 CLS:PRINT"NOW VERIFY THAT CORRECTIONS ARE OKAY BY EDITING
L#2000 TO FILESPEC' XXXXXXXX' AND READING THE FILE. WHEN ASKED
IF ANY CORRECTIONS NEEDED, PREES 'BREAK' IF ALRIGHT ELSE DO
COORECTIONS AGAIN.
4510 PRINT"IF CORRECTIONS ARE ALRIGHT, THEN WE MUST:
                1.KILL OLD FILESPEC
                2.RENAME 'XXXXXXXX' TO OLD FILESPEC
                3.KILL 'XXXXXXXX'
```

For Model I & II

This program works just fine on the Model I or Model II. Model II users simply delete lines 940 and 950, the "Printer Off Trap".

Disk and Printer Required.

A Simple Payroll Program

Mike Schmidt
Editor

"Payroll" is a rather simple payroll program. About the only frill it has is the fact that vacation hours can be accounted for and quarterly totals may be kept and cleared. These quarterly totals, of course, are necessary when filling out the various and sundry tax reports at the end of each quarter.

The item labeled "L & I" is all the same as State Welfare in most states. In our case it stands for "Department of Labor and Industry" - call it what you like, it is still a deduction. Only one class of occupation is accounted for in this program.

The program is designed for very small businesses, with just a few employees, although you could have as many as there are sectors on a disk. The employee file name on disk is the individual first name or nickname, but the paystub is printed out showing full name, social security number, etc.

Operation is quite simple: You just follow the prompts to create a new record (being careful not to use the same file name twice!). No provision has been made to edit a pay record, you must copy the old information by hand and create a new record (using the same name this time) for that employee. This would occur when there is a pay raise, or a change of tax status,.for example.

The program as written is designed to work only with a printer, but changing the LPRINTS to PRINTS will give the same information on the screen.

Running the payroll is easy. Simply enter the date, and call each employee up by their file name, answer the question about hours worked and vacation, and it will print out their paystub and update the disk file automatically. Any or all employees may be done in one session, and you use the sentinel ".END" to end the session.

Quarterly totals work almost the same way. You call each employee by file name and the printer will print their full name, etc., and their totals for the current quarter. This can be a real aid in tax preparation, where hours worked, deductions for FICA and Federal Withholding Tax (FEDTAX) are required. After the quarterly totals are printed for each employee, you get a total of totals, after which you may clear the current quarter to start a new one. Clearing clears only those items which do not accumulate for the year, that is, it does not clear year to date totals for any employee. W-2 forms may be created from the information on the last paystub of the year.

It should be noted that quarterly totals are called "quarterly" because that is the requirement of most state and federal taxes. You may call it "monthly" if that is what you need, and clear them at the end of the month. The program is not rigid.

Because of the flexibility of the program, it will be necessary to re-create all of your employee pay records at the end of the year.

Why not put all the frills in? Because we wanted to give you a simple, but useable, payroll program which is easy to understand. Once understood, you can surely put any type of bells and whistles you like on it.

Some of the features you may want to add are: Year end clear, an edit feature, more deductions (for bonds, etc), and maybe even a tax look-up table. But that is all your ball of wax, here is the bare-bones program, take it and run.... ●

```
10 REM * * * * * * * * * * * * * * * * * * * * * * * * * * * *
20 REM *          (C)1980 80-NW PUBLISHING CO              *
30 REM * * * * * * * * * * * * * * * * * * * * * * * * * * * *
40 CLEAR1000
50 CLS:PRINTTAB(5)," P A Y R O L L - MASTER INDEX"
60 PRINT:PRINT"ENTER 1 - TO CREATE A NEW PAY RECORD"
70 PRINT"ENTER 2 - TO AUDIT AN EXISTING PAY RECORD"
80 PRINT"ENTER 3 - TO DO THE PAYROLL"
90 PRINT"ENTER 4 - FOR ACCUMULATED TOTALS"
100 PRINT"ENTER 5 - TO END SESSION"
110 PRINT:INPUT"MAKE YOUR CHOICE AND ENTER";GG
120 ONGGGOTO140,400,740,1120,1570
130 REM * * * * * CREATE PAY RECORD * * * * * * * * * *
140 CLS:PRINT"    P A Y  D A T A  R E C O R D  C R E A T I O N"
150 PRINT:PRINT"FOLLOW THE PROMPTS TO CREATE A NEW FILE FOR AN
    EMPLOYEE. ENTER ZERO AMOUNTS IF APPLICABLE"
160 LINEINPUT"EMPLOYEE NAME";A$
170 LINEINPUT"SOCIAL SECURITY NUMBER";B$
180 INPUT"HOURLY RATE OF PAY";R
190 INPUT"FED TAX % (IE, .12)";H2
200 INPUT"MED INSURANCE PER PERIOD";H3
210 INPUT"OLD FICA";O1
220 INPUT"OLD FEDTAX";O2
230 INPUT"OLD L & I";O3
240 INPUT"OLD MEDICAL";O4
250 INPUT"OLD VACATION HOURS";O6
260 Q1=0:Q2=0:Q3=0:Q4=0
270 INPUT"HOURS THIS QTR";Q1
280 INPUT"TOTAL NET THIS QTR";Q2
290 INPUT"TOTAL FICA THIS QTR";Q3
300 INPUT"TOTAL FEDTAX THIS QTR";Q4
310 INPUT"YEAR TO DATE GROSS PAY";YG
320 INPUT"ENTER EMPLOYEE'S FILE NAME";AB$
330 IFAB$=""THEN320
340 INPUT"PRESS ENTER TO POST TO DISK FILE";
350 GOSUB610
360 PRINTAB$"'S  DATA POSTED TO DISK FILE"
370 INPUT"ENTER 1 TO CREATE ANOTHER, 2 FOR INDEX";GG
380 ONGGGOTO140,50
390 REM * * * * * *  AUDIT A PAY RECORD * * * * * * * * * *
400 CLS:PRINT"  A U D I T  A  P A Y  R E C O R D"
410 INPUT"ENTER EMPLOYEE FILE NAME";AB$
420 GOSUB670
430 PRINT"EMPLOYEE NAME ------------------ ";A$
440 PRINT"SOCIAL SECURITY NUMBER -------- ";B$
450 PRINT"HOURLY RATE OF PAY ------------- ";R
460 PRINT"FEDERAL TAX PERCENTAGE -------- ";H2
470 PRINT"MEDICAL DEDUCTION -------------- ";H3
480 PRINT"YEAR TO DATE FICA -------------- ";O1
490 PRINT"YEAR TO DATE FED TAX ----------- ";O2
500 PRINT"YEAR TO DATE L & I ------------- ";O3
510 PRINT"YEAR TO DATE MEDICAL ----------- ";O4
520 PRINT"TOTAL UNUSED VACATION HOURS --- ";O6
530 PRINT"TOTAL HOURS (THIS QTR) --------- ";Q1
540 PRINT"TOTAL NET PAY (THIS QTR) ------- ";Q2
550 PRINT"TOTAL FICA (THIS QTR) ---------- ";Q3
560 PRINT"TOTAL FED TAX (THIS QTR) ------- ";Q4
570 PRINT"YEAR TO DATE GROSS ------------- ";YG
580 INPUT"FOR ANOTHER ENTER 1 ELSE FOR INDEX ENTER 2";GG
590 ONGGGOTO390,50
600 END
610 REM * * * * * * OPEN FILE AND WRITE SUBROUTINE * * * * * *
620 OPEN"O",1,AB$
630 PRINT#1,CHR$(34);A$;CHR$(34);CHR$(34);B$;CHR$(34);",";
     R;H2;H3;O1;O2;O3;O4;O6;Q1;Q2;Q3;Q4;YG
640 CLOSE1
650 RETURN
660 END
670 REM * * * * *  OPEN FILE AND READ SUBROUTINE * * * * * * *
680 OPEN"I",1,AB$
690 INPUT#1,A$,B$,R,H2,H3,O1,O2,O3,O4,O6,Q1,Q2,Q3,Q4,YG
700 IFEOF(1)THEN 710
710 CLOSE1
720 RETURN
730 END
740 REM * * * * * POST PAYROLL AND PRINT STUB * * * * * * * * *
750 CLS:PRINTTAB(5),"P A Y R O L L"
760 PRINT:PRINT"TURN ON THE PRINTER AND ALIGN THE PAPER"
770 PRINT"USE '.END' FOR EMPLOYEE NAME TO END SESSION"
780 INPUT"ENTER DATE OF PAYROLL (10 AUG 80)";D$
790 PRINT
800 INPUT"EMPLOYEE FILE NAME";AB$
810 IFAB$=""THEN800
820 IFAB$=".END"THEN50
830 GOSUB670
840 PRINT"HOURS ";AB$;" WORKED THIS PERIOD":INPUTH0
850 PRINT"WAS VACATION TAKEN THIS PERIOD? (Y/N)"
860 V$=INKEY$:IFV$=""THEN860
870 IFV$="Y"THEN880ELSE900
880 INPUT"HOW MANY HOURS WERE TAKEN?";HT
890 O6=O6-HT
900 O6=O6+1.5385:GP=R*H0:CF=GP*.0613:CT=GP*H2:CM=H3:CL=H0*.0126
    :    NP=GP-(CF+CT+CM+CL)
910 O1=O1+CF:O2=O2+CT:O3=O3+CL:O4=O4+CM:Q1=Q1+H0:Q2=Q2+GP
    :Q3=Q3+CF:Q4=Q4+CT:YG=YG+GP
920 CF=INT(CF*100+.5)/100:CL=INT(CL*100+.5)/100:CT=INT(CT*100+.
    5)/100
930 NP=INT(NP*100+.5)/100:O1=INT(O1*100+.5)/100:O3=INT(O3*100+.
    5)/100
940 REM * PRINTER OFF TRAP *
950 IFPEEK(14312)127THEN50
960 LPRINTTAB(2);A$;TAB(25);B$;TAB(38);"FOR WEEK ENDING ";D$
970 LPRINT" ":LPRINT"HOURS";TAB(8);"VAC HRS AVAIL";TAB(26);"RAT
    E";TAB(36);"GROSS PAY";TAB(48);"NET PAY"
980 O7=INT(O6*10+.5)/10
990 LPRINTTAB(1);H0;TAB(10);O7;TAB(25);R;TAB(38);GP;TAB(48);NP
1000 LPRINT" "
1010 LPRINT"                              ----- DEDUCTIONS -----"
1020 LPRINTTAB(21);"FICA";TAB(36);"FEDTAX";TAB(48);"MED";TAB(57
    );"L&I"
1030 LPRINT"CURRENT-----------";TAB(20);CF;TAB(36);CT;TAB(46);C
    M;TAB(55);CL
1040 LPRINT"YEAR TO DATE------";TAB(20);O1;TAB(36);O2;TAB(46);O
    4;TAB(55);O3
1050 LPRINT"YEAR TO DATE GROSS = $ ";YG
```

```
1060 LPRINTSTRING$(64,45)
1070 FORI=1TO2:LPRINT" ":NEXT
1080 GOSUB610
1090 AB$=""
1100 GOTO790
1110 END
1120 REM * * * * * PRINT QUARTERLY TOTALS * * * * * * * * *
1130 CLS:PRINT"Q U A R T E R L Y  T O T A L S  &  C L E A R"
1140 PRINT
1150 PRINT"THIS ROUTINE WILL PRINT AN ACCUMULATED SUMMARY OF HO
     URS"
1160 PRINT"WORKED, GROSS PAY, FICA AND FEDTAX FOR THE CURRENT"
1170 PRINT"QUARTER. TURN ON YOUR LINE PRINTER...."
1180 PRINT
1190 PRINT"ENTER 1 - FOR SUMMARY"
1200 PRINT"ENTER 2 - TO CLEAR ACCUMULATED TOTALS FOR QUARTER"
1210 PRINT"ENTER 3 - TO RETURN TO MASTER INDEX"
1220 PRINT:INPUT"MAKE YOUR CHOICE AND ENTER";GG
1230 ONGGGOTO 1240,1440,50
1240 CLS:PRINT"DON'T FORGET EMPLOYEES WHO MAY HAVE QUIT DURING"
1250 PRINT"THE QUARTER - YOU HAVE TO COUNT THEM TOO!"
1260 PRINT:PRINT"FOLLOW THE PROMPTS TO ENTER EACH EMPLOYEE'S"
1270 PRINT"TOTALS"
1280 LPRINT"TOTAL HOURS","TOTAL GROSS","TOTAL FICA","TOTAL FEDT
     AX"
1290 PRINT"ENTER 'END' FOR EMPLOYEE NAME TO END INPUT AND"
1300 PRINT"PRINT SUMMARY"
1310 INPUT"ENTER EMPLOYEE FILE NAME";AB$
1320 IFAB$=""THEN 1310
1330 IFAB$=".END"THEN 1400
1340 GOSUB670
1350 LPRINTA$;"  ";B$
1360 LPRINTG1,G2,G3,G4
1370 T1=T1+G1:T2=T2+G2:T3=T3+G3:T4=T4+G4
1380 AB$=""
1390 GOTO1310
1400 LPRINT"TOTAL HOURS","TOTAL GROSS","TOTAL FICA","TOTAL FEDT
     AX"
1410 LPRINTT1,T2,T3,T4
1420 GOTO1120
1430 END
1440 REM * * CLEAR ACCUMULATED QUARTERLY TOTALS * * * *
1450 CLS:PRINT"MAKE SURE YOU PRINT A SUMMARY BEFORE YOU DO THIS
     '":PRINT
1460 PRINT"ENTER 'END' FOR EMPLOYEE NAME TO RETURN TO MASTER I
     NDEX"
1470 INPUT"CLEAR TOTALS FOR EMPLOYEE FILE NAME";AB$
1480 IFAB$=""THEN1470
1490 IFAB$=".END"THEN 1120
1500 GOSUB670
1510 G1=0:G2=0:G3=0:G4=0
1520 GOSUB610
1530 PRINTAB$"'S QUARTERLY TOTALS ARE CLEARED"
1540 AB$=""
1550 GOTO1470
1560 REM * * * * * EXIT ROUTINE * * * * * * * * * * * * * *
1570 PRINT"DON'T FORGET TO BACKUP YOUR DATA!":CLOSE:END
```

Here are three samples of the simple payroll program. The first two are samples of the paystub, while the third is an example of the printout of the quarterly totals.

```
  JOHN Q PUBLIC              123-34-3456  FOR WEEK ENDING 7 NOV 80

HOURS    VAC HRS AVAIL      RATE        GROSS PAY    NET PAY
  80        3.1              5             400        316.13

                          ----- DEDUCTIONS -----
                      FICA            FEDTAX        MED        L&I
CURRENT-----------    24.52             56         2.34       1.01
YEAR TO DATE------    30.65             70         4.68       1.26
YEAR TO DATE GROSS = $  500
```

```
  MARY DOE                   123-80-8080  FOR WEEK ENDING 7 NOV 80

HOURS    VAC HRS AVAIL      RATE        GROSS PAY    NET PAY
  80        3.1              6             480        381.14

                          ----- DEDUCTIONS -----
                      FICA            FEDTAX        MED        L&I
CURRENT-----------    29.42             67.2       1.23       1.01
YEAR TO DATE------    44.13            100.8       2.46       1.51
YEAR TO DATE GROSS = $  720
```

```
TOTAL HOURS      TOTAL GROSS      TOTAL FICA       TOTAL FEDTAX
JOHN Q PUBLIC  123-34-3456
 100               500              30.65             70
MARY DOE  123-80-8080
 120               720              44.136            100.8
TOTAL HOURS      TOTAL GROSS      TOTAL FICA       TOTAL FEDTAX
 220               1220             74.786            170.8
```

ANATOMY of the Payroll Program

R C Bahn

I. INTRODUCTION

The specific capabilities of this program are well described and illustrated in the previous text. In general, this Basic program demonstrates a modular self-documenting programming technique, use of Basic subroutines, acquisition of data by interactive keyboard procedures, writing to and reading from serial (sequential) disk data files, a procedure for review of files on the video screen and practical use of printer output.

The statements and functions of the Basic Language have been designed to result in listings of programs which can be read by a human being with relative ease. These attributes of the language can be obscured or illuminated by the programmer. Increased clarity of the listing can always be produced by using one line for each statement and adding at strategic locations precise and even graphic comments.

The organization of program flow should be initially worked out on paper. The overall objectives of the program, can be broken down into small modules. Within these, subtasks which are repeated in the program can be identified. These subtasks can be eventually written as subroutines. Flow between modules should be arranged in a manner which minimizes the "jumping around". Ideally, each module should have a single entrance and a single exit. Such an organization reduces the necessity for large numbers of GOTO statements.

The Payroll program contains two subroutines: One for writing serial disk files (line 610) and one for reading a serial disk file (line 670). The remainder of the program is organized into eight modules beginning with lines 10, 130, 390, 740, 940, 1120, 1430 and 1560 respectively. All modules are clearly identified in the program listing. The execution of Payroll begins with a display of options for the selection of the desired module. Except for branching to subroutines, the execution within each module is essentially linear.

Finally, clarity of the program can be increased by thoughtful selection of the names of variables. The selected symbols should first give the reader a hint of the function of the variable. In addition, the scheme for naming variables may be systemized into "series" in which similar names or similar sequences perform somewhat different but similar funtions. Ultimately numeric portions of variable names can appear as single or multiple subscripts. Section II, Variables, illustrates all of these principles except subscripting.

In summary, clear, self-documenting Basic programs are usually characterized by (1) the use of one line for each Basic statement, (2) strategically placed often graphic comments, (3) modular programming, (4) linear program flow with few GOTO statements, (5) use of subroutines, and (6) judicious choice of variable names.

II VARIABLES

A. String Variables

1. A$, employee's name, first used in line 160.
2. AB$, file name, first used in line 320.
3. B$, social security number, first used in line 170.
4. D$, date of payroll, first used in line 780.
5. V$, keyboard input variable, used in lines 860, 870.

B. Single precision variables

1.C series: Current values, first used in line 900.
   a. CF, current FICA deduction.
   b. CL, current L&I deduction.
   c. CM, current medical deduction.
   d. CT, current federal tax deduction.
2. H Series: Current input variables.
   a. H0, hours worked this period (line 840).
   b. H2, federal tax rate (line 190).
   c. H3, current medical deduction (line 200).
   d. HT, hours vacation time this period (line 880).
3. O(n) series: initialized as old values and modified later, first used in lines 210 - 250.
   a. O1, FICA (line 210).
   b. O2, FEDTAX (line 220).
   c. O3, L&I (line 230).
   d. O4, medical (line 240).
   e. O6, vacation hours (line 250).
   f. O7, rounded vacation hours (line 980).
4. Q series: Quarterly values first used in lines 260-300.
   a. Q1, quarterly hours.
   b. Q2, net for quarter.
   c. Q3, quarterly FICA.
   d. Q4, quarterly federal tax.
5. T series: Total (year to date) values used in lines 1370 and 1400. T1, T2, T3, T4 are analogous to Q1 through Q4.
6. Other variables.
   a. GG, Keyboard input variable first used in line 110.
   b. GP, Gross pay used in lines 900 and 910.
   c. I, Loop index used in line 1070.
   d. NP, Net pay used in lines 900, 930, 990.
   e. R, Hourly rate of pay first used in line 180.
   f. YG, Year to date gross pay first used in line 310.

III. DESCRIPTION OF MODULES

10-120	Initialization module with video screen-keyboard interactive routines.
10-30	Program title.
40	Clear variable space and reserve space for string variables.
60-110	Print master index on video screen.
110	Prompt user to make choice.
120	Branch to appropriate line depending on user's choice.
130-380	Module for creation of pay record through video screen-keyboard interactive routines.
260	Set Q series of variables equal to zero.
330	Return to line 320 if ENTER was pressed before file name was typed.
350	Branch to write a disk file and return.
380	Exit option to create a new record or return to the master index.
390-600	Module for auditing a pay record.
420	Branch to read a disk file and return.
590	Exit option to review another file or return to the master index.
610-660	Subroutine to open or write a serial data file.
620	File AB$ is opened in buffer #1.
630	Note, this is one of the most crucial lines in the entire program. It is the only record of the format of the disk file. Compare line 630 with line 690. The sequence of variable names in these statements must correspond exactly. The three added ASCII characters in line 630 are delimiters to allow DOS to properly interpret the leading string variables A$ and B$.
640	Close buffer #1. In general, and to avoid disasters, all buffer files should be closed as soon as possible.
670-730	Subroutine to open file and read subroutines.
690	Compare this line again with line 630 and review associated comments.
740-1110	Module to post payroll and print stub.
750-880	Interactive data input.
810	Don't forget file name.
820	Note leading period in ".END", this is the exit.
890-930	Major computations of deductions and net pay.
900	Constants are vacation constant, FICA rate and L & I rate respectively.
910	Modify 0(h) and Q series of variables, update YG.
920-930	Round all appropriate variables to ne irest penny.
940, 950	If printer is not turned on return to master index, this avoids a system "hang-up".
960-1060	Printer output routine.
1070	Skips two lines on printer.
1080	Write new file.
1090	Set up line 800 for end routine.
1100	Go to end routine exiting at line 820.
1110	Precautionary ending.
1120-1430	Module to print quarterly totals.
1130-1270	Video screen printing routines.
1190-1220	Display menu of options.
1230	Choose option, option 3 is the exit
1240-1410	Interactive keyboard-printer routines.
1420	Return to module menu of options.
1440-1550	Module to clear accumulated quarterly totals.
1490	".END" denotes exit.
1560-1570	Module to exit from program.
1570:1	DON'T FORGET TO BACKUP YOUR DATA!
1570:2	Precautionary closure of buffers.

U.S. POSTAL SERVICE

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

(Required by 39 U.S.C. 3685)

1. TITLE OF PUBLICATION: THE 80-U.S. JOURNAL	A. PUBLICATION NO.: 0 1 9 1 0 3 5	2. DATE OF FILING: 1oct 80
3. FREQUENCY OF ISSUE: Bi-Monthly	A. NO. OF ISSUES PUBLISHED ANNUALLY: Six (6)	B. ANNUAL SUBSCRIPTION PRICE: $16.00

4. LOCATION OF KNOWN OFFICE OF PUBLICATION (Street, City, County, State and ZIP Code) (Not printers): 3838 South Warner St., Tacoma, Washington 98409

5. LOCATION OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS (Not printers): 3838 South Warner St., Tacoma, Washington 98409

6. NAMES AND COMPLETE ADDRESSES OF PUBLISHER, EDITOR, AND MANAGING EDITOR

PUBLISHER (Name and Address): Irvin Mike Schmidt 3836 S Warner St., Tacoma, WA 98409

EDITOR (Name and Address): Same

MANAGING EDITOR (Name and Address): Same

7. OWNER (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.)

NAME	ADDRESS
Same as Item 6	

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES (If there are none, so state)

NAME	ADDRESS
None	

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 132.122, PSM)

The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)

☐ HAVE NOT CHANGED DURING PRECEDING 12 MONTHS ☐ HAVE CHANGED DURING PRECEDING 12 MONTHS (If changed, publisher must submit explanation of change with this statement.)

10. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES PRINTED (Net Press Run)	9000	12007
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS AND COUNTER SALES	1843	3574
2. MAIL SUBSCRIPTIONS	3776	4504
C. TOTAL PAID CIRCULATION (Sum of 10B1 and 10B2)	5619	8078
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	657	2507
E. TOTAL DISTRIBUTION (Sum of C and D)	6276	10585
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED AFTER PRINTING	2712	1422
2. RETURNS FROM NEWS AGENTS	12	0
G. TOTAL (Sum of E, F1 and 2—should equal net press run shown in A)	9000	12007

11. I certify that the statements made by me above are correct and complete. SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER: Irvin Mike Schmidt

12. FOR COMPLETION BY PUBLISHERS MAILING AT THE REGULAR RATES (Section 132.121, Postal Service Manual)

39 U.S.C. 3626 provides in pertinent part: No person who would have been entitled to mail matter under former section 4359 of this title shall mail such matter at the rates provided under this subsection unless he files annually with the Postal Service a written request for permission to mail matter at such rates.

In accordance with the provisions of this statute, I hereby request permission to mail the publication named in Item 1 at the phased postage rates presently authorized by 39 U.S.C. 3626.

SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER: Irvin Mike Schmidt

PS Form 3526, Aug 1978 (Page 1) (See instructions on reverse)

A Plotter Interface

Anthony J Wilson
Columbia, MO

An economical Plotter for your Model I TRS-80

I have always been impressed by the elaborate graphic designs that can be produced by computers equipped with plotting devices. Quite some time ago, I decided that I would like to have this capability on my TRS-80, but was soon discouraged by the high prices of the various different plotters on the market. I found it hard to justify spending $1000 or more for another peripheral device - which is considerably more than I spent on my printer.

I had almost given up the idea of buying a plotter until one day, while browsing through the Heathkit catalog, I discovered that they made an X-Y plotter kit for less than $500, a price a little easier to live with. The big catch was that the Heath plotter is an analog device controlled by voltage levels at the X and Y inputs, and therefore not capable of being driven directly by a computer.

After discussing the problem with a few of my electronic expert friends, I decided that a digital to analog converter to interface the TRS-80 to the Heath plotter would not be too difficult to design, and without further ado, went ahead and ordered the plotter kit.

As with most of the kits I have ordered from Heath, it was not very long before a large box arrived from Benton Harbor. Upon opening the box I found a typical Heath presentation with simple, detailed and easy to follow instructions and the tremendous attention to detail that always impresses me whenever I buy one of their products.

I sat down immediately and began to assemble the kit. Three nights later it was together - I don't remember how many hours I put in, but it didn't seem like very long. Upon completion and the initial testing I plugged it in and it worked perfectly. The next step was calibration, and this was simply a matter of following instructions again.

So I had my plotter, but without an interface to the computer it was little more than an expensive "Etch-A-Sketch". Now came the hard part; how to get the TRS-80 to talk to the plotter.

After some experimenting with the manual controls of the plotter I decided that I would need resolution of at least 1024 by 1024 (10 bits) to get adequately smooth curves. I had already decided to connect the plotter to the computer using parallel ports and digital to analog (D to A) integrated circuits, but if I needed 10 bit resolution from an 8 bit computer I would have to use two ports for each axis (X and Y), and somehow couple them together.

After studying the electrical characteristics of various Digital to Analog Integrated Circuits (D to A IC's), and their prices, I decided to use one DAC-02 Integrated Circuit for each axis and decode the ports with a 74LS154 (IC). The DAC-02 is a 10 bit D to A converter and the two I have produce very close to 0.1 volt per bit, which gives about 10.25 volts maximum. This is ideal for the plotter's 1 volt per inch range. I was very tempted to use 12 bit D to A converters, but the tremendous difference in price soon changed my mind. The DAC-02 is not particularly cheap either, at $22 per chip.

The circuit that I am now using is shown here and is the result of a fair amount of experimentation, but adheres to the original principle; 5 parallel ports are used and I have chosen 1 through 5 for convenience. Two ports are used for the X axis, two for the Y axis, and one to raise and lower the pen. The ports are all latched together so that the appropriate X-Y voltages are all sent to the plotter simultaneously. This is necessary to avoid "glitches" in the tracing which occur if the two components of any axis are sent separately. It is also set up so that an "IN" from port 5 raises the pen, while an "OUT" to the same port will lower the pen. I have arbitrarily assigned ports 1 and 2 to the X axis, and ports 3 and 4 to the Y axis. In each case the odd numbered port controls the lower 8 bits and the even numbered port the upper 2 bits with port 1 also having the function of sending the stored X and Y values to the plotter. Thus, the value for port 1 should always be the last of the four values sent to the interface.

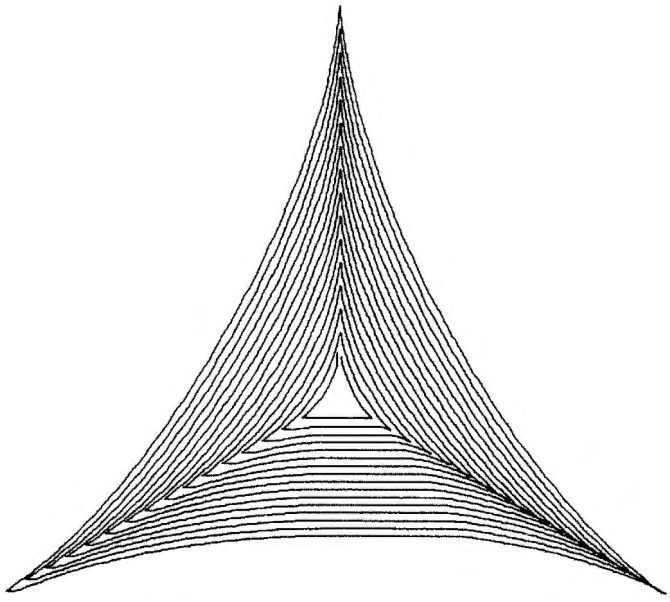


Figure 1

Sample output of the Heathkit X-Y Plotter under TRS-80 control.

This circuit is designed to put only one low power Schottky input on any one bus line and is therefore well within the manufacturer's specified limit of 1 standard TTL input per line. There are no outputs from the interface back to the computer.

How does this circuit work? Basically it is very simple. The port addresses are decoded by U1 (which is a 1 of 16 data distributor) in combination with two of the gates from U2. U2 decodes the top 4 bits of the port address and U1 the bottom 4 bits. Thus, whenever the correct address appears on the bus the appropriate output line of U1 goes low. If an OUT* signal is received at the same time, the two signals are ANDed by one of the gates of U5 and send a high to the clock of the appropriate latch (U6, U7 or U8) for that port.

U8 is a 4 bit latch and handles the 2 bits for both port 2 and port 4, U6 and U7 are both 8 bit latches and handle the 8 bits for ports 1 and 3 respectively. U12 and U13 are the D to A chips. The lower 8 bits of U12 are directly connected to U6, while the upper bits of U12 and all 10 bits of U13 are connected to the outputs of U10 and U9. U10 and U9 store the 10 bits for the Y axis and the upper 2 bits for the X axis until an OUT is received on port 1. The clocks of U6, U10 and U9 are all connected together, resulting in simultaneous transfer of data from their input lines to the input lines of the two D to A converters.

U3 and U4 are simple non-inverting buffers whose sole function is to insure that only one low power Schottky input is put on any of the bus lines. U11 is a flip-flop which lifts the pen in one state and lowers it in the other. The flip-flop is controlled by the port 5 output being coupled by the remaining two gates of U2 to both the IN* and OUT* signals. Thus, an IN from port 5 raises the pen, and an OUT to port 5 lowers the pen.

The circuit diagram does not include the capacitors and power supply connections for any of the "LS" series Integrated Circuits. All of the LS IC's should have a 0.1 uF capacitor connected between the V+ and ground pins. Each of these IC's must be connected to the power supply as follows:

U1 pin 24 is +5 volts, pin 12 is ground
U6, U7 & U9, pin 20 is +5 volts, pin 10 is ground
U3, U4, U8 & U10, pin 16 is +5 volts, pin 8 is ground
U2, U5 and U11, pin 14 is +5 volts, pin 7 is ground

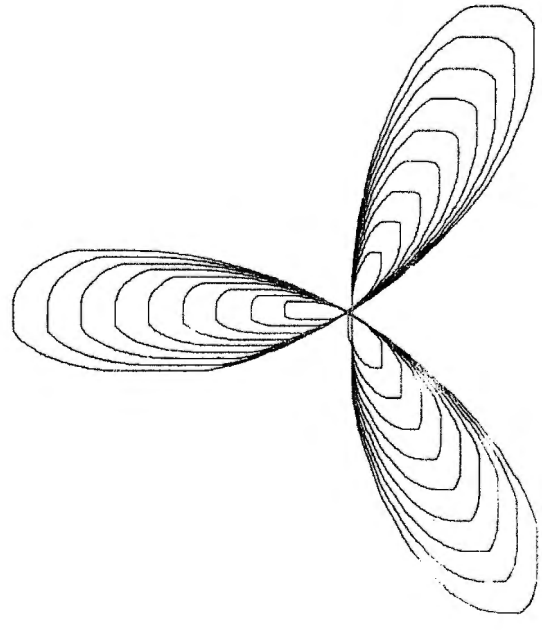
A 40 pin conductor ribbon cable with an edge card connector is used to physically connect the interface to the TRS-80. Only 19 of the bus lines are used, and these are outlined in Table 1, with figure 5 showing the numbering of the edge card contacts at the rear of the keyboard. Non-expansion interface machines can be connected to the plotter interface by plugging the cable assembly on the edge connector at the rear of the keyboard. For expansion interface users, the keyboard edge connector is already occupied. If the TRS-80 has an expansion interface of the older type (like mine), with a buffered cable, it will be necessary to obtain a special buffered cable with an auxillary edge card from your neighborhood Radio Shack dealer. Newer expansion interfaces, I have been told, have an operative bus extension port.

Once the interface is connected to the bus and its power is turned on the pen will be kept in the same position until an OUT command is sent to port 1. The initial position of the pen is random and the state of the raise/lower flip-flop is also random. Therefore, it pays to use the manual pen raise on the plotter until everything is set up as desired by software.

* an asterisk is the symbol used here for an active low bus line.

Figure 2

Another sample drawing.



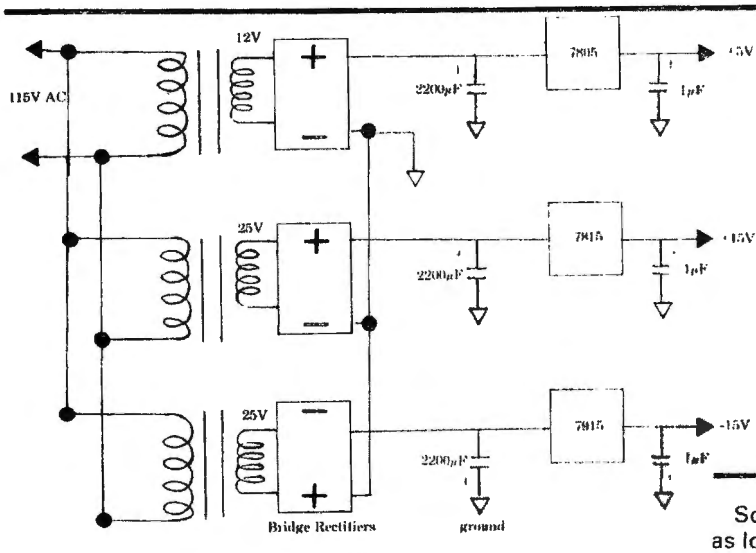


Figure 3
Circuit for power supply.

This is the power supply used by the author, but any regulated supply providing +5, +15 and -15 volts with a similar current rating will work as well.

Transformers

1 - 12 volt 300 ma (Radio Shack)
2 - 25 volt 300 ma (Radio Shack)

Integrated Circuits

1 - each, 7805, 7815, 7915

Capacitors

3 - 2200 uf electrolytic
3 - 1 uf tantalum

Bridge Rectifiers

3 - 50 volt, 500 ma
3 Heat sinks for voltage regulator IC's

Software to drive the plotter is relatively simple to write, as long as all plots are calculated using X and Y values. I use the following subroutine to decode simple X and Y values into the correct values for the four X-Y ports of the plotter:

```
10000 IF X<0 THEN X=0 ELSE IF X>1024 THEN X=1024
10010 IF Y<0 THEN Y=0 ELSE IF Y>1024 THEN Y=1024
10020 OT(2)=INT(X/256): OT(1)=(X/256—OT(2) )*256
10030 OT(4)=INT(Y/256): OT(3)=(Y/256—OT(4) )*256
10040 FOR I=4 TO 1 STEP —1: OUT I, OT(I): NEXT I
10050 RETURN
```

To raise the pen the following line is used:

```
10100 DUMMY = INP(5): RETURN
```

To lower the pen I use the following:

```
10200 OUT 5,0: RETURN
```

Schematic for 10-bit D to A converter to drive Heathkit X-Y Plotter.

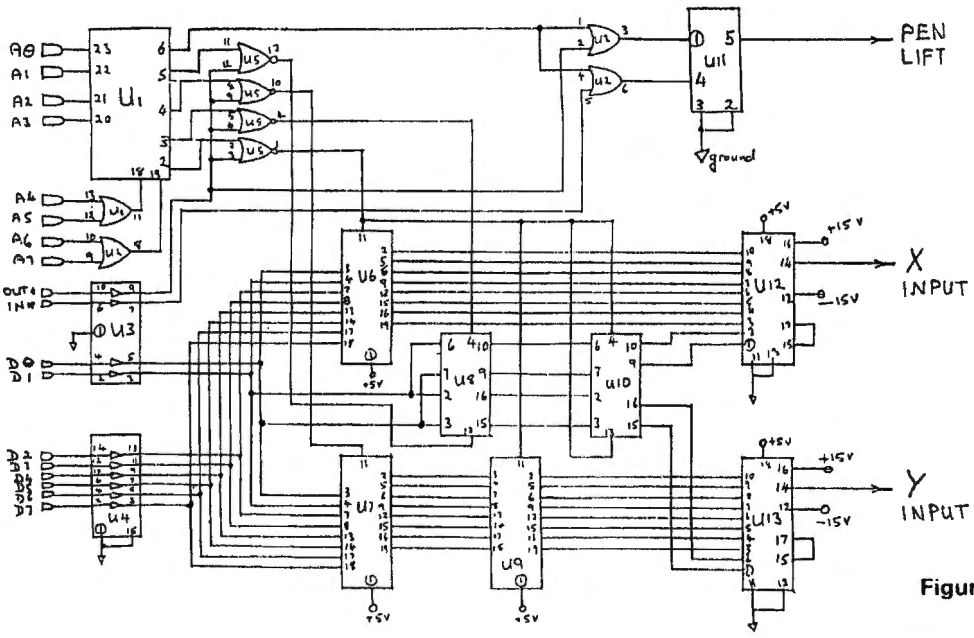


Figure 4

TABLE 1
Bus lines used by the interface

12 = OUT*	25 = A0	31 = A4
18 = D4	26 = D3	32 = D2
19 = IN*	27 = A1	34 = A3
20 = D7	28 = D5	35 = A5
22 = D1	29 = GROUND	36 = A7
24 = D6	30 = D0	38 = A6
		40 = A2

*Asterisk is the symbol for an active low

TABLE 2
Components required for this interface are as follows

1 - 74LS5154 I.C.
1 - 74L532 I.C.
2 - 74L5367 I.C.'s
1 - 74L502 I.C.
3 - 74L5273 I.C.'s
2 - 74L575 I.C.'s
1 - 74L574 I.C.
2 - DAC02 I.C.'s
11 - .1 μF ceramic capacitors.
1 - 40 conductor ribbon cable with edge card connector.
1 - Perf board or P.C. board for mounting components.

(I used the former with wire wrap sockets for all I.C.'s and wire wrap posts to connect to ribbon cable)

1 - Box to protect interface from children, pets, and other hazards. (I used a Radio-Shack no.270-261).
1 - Power supply with +5, +15, and —15 volts (Fig 3)

By using subroutines for these functions they can be used many times during a program without having to rewrite them each time. For example the following line raises the pen and zeroes its position in preparation for a new plot:

```
100 GOSUB 10100: X=0: Y=0: GOSUB 10000
```

A simple program to draw sine curves:

```
100 FOR J=0 TO 628
110 X=J/100*K
120 Y=SIN(J/100)*K
130 GOSUB10000
140 NEXT J
```

A value must be given to "K" before running this sequence and the size of the sine wave will be a function of that value.

I have had a great deal of fun experimenting with this interface and the plotter and hopefully some of the more adventurous readers will too. ●

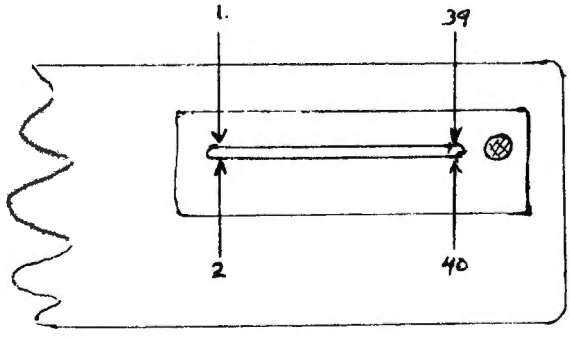


Figure 5

Edge Connector viewed from rear of keyboard.

NOTES

In case you hadn't noticed, there is a small problem with Basic's handling of "raise to the power" function. For example, raising 2 to a power can be handy for converting decimal to binary. Unfortunately, here is what you get out of both Model I and II Basic for the powers of two:

2^1=2
2^2=4
2^3=8
2^4=16
2^5=32
2^6=64
2^7=128
2^8=256
2^9=512
2^{10}=1024
2^{11}=2048
2^{12}=4096
2^{13}=8192.01
2^{14}=16384
2^{15}=32768

This really isn't much of a problem, as the only one that is off is 2^{13}. Unfortunately, when this is converted to binary for ANDing later, the results can get somewhat strange. No fix is offered, this is just something to watch.

Are you tired of typing a menu selection and having it give you a bad response because you used the wrong number? Then try this:

```
200 REM * * * * * * START OF MENU * * * * * *
        lines 210 - 240 are menu selections 1 to 4
250 PRINT@896,"SELECTION: ";
260 C$=INKEY$:IFC$=""THEN260ELSEIFVAL(C$)<1
    ORVAL(C$)>4THEN260
270 PRINT C$;:C=VAL(C$)
280 ON C GOSUB 1000,2000,3000,4000
290 GOTO 200
```

This looks much neater on the screen than using an INPUT statement since there is no response at all to the wrong key.

Try this for an interesting looking error trap that will catch an operator's attention if something goes wrong. Early in the program, put in the statement:

ON ERROR GOTO 20000

This will send your program to an error trap at line 20000 if you have a problem. Then put this in at line 20000:

```
20000 PRINT@960,TAB(60);
20010 FORZZ=1 TO 30
20020 C$=INKEY$:IFC$<>""THEN IF ASC(C$)=
    13 THEN RESUME 200
20030 NEXT ZZ
20040 PRINT@960,"ERROR ";ERR/2+1;" IN LINE ";
    ERL;"PRESS ENTER FOR MENU";
20050 FORZZ=1 TO 30
20060 C$=INKEY$:IFC$<>""THEN IF ASC(C$)=
    13 THEN RESUME 200
20070 NEXT ZZ
20080 GOTO 20000
```

Line 200 is the program menu or other start point. Let's see what this does. First, the last line on the screen is cleared by line 20000. Then a short loop leaves it clear and checks to see if the ENTER key has been pressed (ASCII code 13). If it has, then the program resumes normally, otherwise it keeps looking.

Then, the program prints the error number, line of occurence, and a message to press enter on the last line and goes back to check the ENTER key again. If not pressed, the cycle starts over with blanking the line again. Now we have a flashing error message!

Note that the repetition rate of the flashing is controlled by the length of the FOR..NEXT loop. The value of 30 seems to give a pleasant display. Also, Model II users will want to change the PRINT @ location in lines 20000 and 20040 to somewhere around 1840, and the ERR/2+1 can be replaced with ERR.

NOTICE NOTICE NOTICE

The video controller of the Model II is mapped to I/O port 255(FFH). If an output is made to this port with a value of less than 25, there is a very good chance that YOUR VIDEO WILL BE DESTROYED! We don't know why this happens yet, but we do know that about a dozen of them across the country have been blown in this fashion, resulting in an expensive repair bill. If for any reason your screen should go blank unexpectedly, accompanied by a very high-pitched whining noise, you have less than 7 seconds to TURN THE COMPUTER OFF! Don't worry about having disks in the drives or anything. Even the data you lose is cheap compared to having to do without a computer while it is being repaired.

There is a problem in TRSDOS when using random files. It seems that once disk space has been allocated, the only way to de-allocate is to kill the file and reopen it with the appropriate new end-of-file. Of course, you are left up to the whims of fate as to whether or not your file will survive this intact. If you need to compress your file to make more space available on the disk there simply is no way to do it. However, some searching reveals some interesting facts. When you enter Basic and tell it how many files you want, memory space is set aside for the 256 byte disk record and for the Data Control Block (DCB, also known as the Device Control Block). The DCB is 32 bytes long, and contains the filespec before OPEN and (sometimes) after CLOSE. The 12th and 13th bytes of the DCB contain the number of the *last* record in the file after the file has been opened. Basic will be more than happy to increment this number for you, but has no provisions to decrement it. If just so happens that the DCB lives right above the buffer which Basic sets aside for disk I/O. To find the DCB, FIELD your buffer and find the VARPTR of the first item you have FIELDed. Then back up 32 bytes and there you are. To find the Ending Record Number (ERN), simply back

```
LIST
10 CLEAR1000
20 OPEN"R",1,"TESTDATA/DAT:3"
30 FIELD1,128AS A$
40 LSETA$="THIS IS A TEST TO SEE IF THE DCB IS WHERE I THINK IT
IS"
50 PUT1,1:PUT1,10
60 X=VARPTR(A$)
70 Y=PEEK(X+1)+(PEEK(X+2)*256)
80 Y=Y-19
90 PRINTPEEK(Y-1)+PEEK(Y)*256
READY
)RUN
 10
READY
```

Figure 1

```
DRIVE   3  LETTERS2  09/08/80   35 TRKS   55 FDES    28 GRANS

NFLPIX/W0              256=LRECL      46 RECS     10 GRANS
ZAKS/REV               256=LRECL      32 RECS      7 GRANS
TEXT2                  256=LRECL      57 RECS     12 GRANS
SCRNPROT/ART           256=LRECL       3 RECS      1 GRANS
SCRNPROT/USR           256=LRECL      21 RECS      5 GRANS
NOTES/NOV              256=LRECL       9 RECS      2 GRANS
TESTDATA/DAT           256=LRECL      (10) RECS    2 GRANS
READY
)_
```

Figure 2

```
)_
10 CLEAR1000
20 OPEN"R",1,"TESTDATA/DAT:3"
30 FIELD1,128AS A$
40 LSETA$="THIS IS A TEST TO SEE IF THE DCB IS WHERE I THINK IT IS"
50 PUT1,1:PUT1,10
60 X=VARPTR(A$)
70 Y=PEEK(X+1)+(PEEK(X+2)*256)
80 Y=Y-19
90 PRINTPEEK(Y-1)
100 POKEY-1,4
110 CLOSE
120 CMD"DIR :3(A)
```

Figure 3

```
DRIVE   3  LETTERS2  09/08/80   35 TRKS   55 FDES    29 GRANS

NFLPIX/W0              256=LRECL      46 RECS     10 GRANS
ZAKS/REV               256=LRECL      32 RECS      7 GRANS
TEXT2                  256=LRECL      57 RECS     12 GRANS
SCRNPROT/ART           256=LRECL       3 RECS      1 GRANS
SCRNPROT/USR           256=LRECL      21 RECS      5 GRANS
NOTES/NOV              256=LRECL       9 RECS      2 GRANS
TESTDATA/DAT           256=LRECL      (4) RECS     1 GRANS
READY
```

Figure 4

up 19 bytes instead of 32, and there you have the 16 bit value of the largest record allocated in the file. Just POKE whatever value you want into it, CLOSE the file and everything will be taken care of. Do be careful to make sure you have the right value here. Once the disk space has been released, it is fair game to be used by any other file.

Figures 1 through 4 are an example of this technique. Figure 1 is a program that creates a file called TESTDATA/DAT on drive 3. The PUT 1,1 get the data onto the disk, and the PUT 1,10 preallocates diskspace for 10 records. The rest of the program simply finds the Ending Record Number in the DCB and assures that it is indeed 10 as it should be. Figure 2 is a directory of the disk using the (A) option. As you can see, the ERN agrees with the NRECS (Number of Records) on the disk. Figure 3 is a slight modification of the program which POKEs a 4 into the least significant byte of the ERN in the DCB, then closes the file. Figure 4 is another directory using the (A) option, verifying that the new NRECS is 4.

TANDY Implements Tele-communication To Manage Parts Ordering System

by Dick Konop
Milwaukee, WI

Tandy Corporation has replaced a manual parts ordering system for their Radio Shack service centers with a computer based electronic system. The idea for the system was conceived in February 1980 by William Schroeder of Galactic Software and Ronnie Franklin of Radio Shack.

The system networks Tandy's National Parts warehouse in Fort Worth to over 100 service centers nationwide. Hardware used by the system is Tandy's TRS-80 Model II. Software which the system uses was written by Galactic Software.

The new system replaces a telephone ordering system which required the service centers to phone in orders daily to Fort Worth. These orders were manually recorded and recopied in sorted picking order onto a picking order form. From this form the orders could then be pulled, and finally shipped to the service centers. The old system made for expensive phone bills, was cumbersome (to say the least), and prone to transposition errors.

The new system is comprised of two operations; order entry, which is done at the service centers, and order receive, which is handled in Fort Worth. The system ties a TRS-80 Model II at each service center to one of several receiving Model II's in Fort Worth.

The order entry process allows for orders to be typed into a Model II at each service center. These orders are stored on disk, and may be edited and deleted prior to being placed. Before actual placement of the order, the order file on disk is manipulated by a pre-processing routine. This routine sorts the order file into picking order, assigns a unique number to each order, and stores this sorted order file on disk. The service center may choose to store a message directed to the parts warehouse along with this sorted order file.

After the order file has been processed, it may then be transmitted to Fort Worth in its sorted form. To send the order file down-line to Fort Worth, the service center brings it's Model II up in the system's terminal mode. Once in the terminal mode, the operator at the service center loads the order file into a RAM buffer of the Model II, and at a specified time places a call to Fort Worth.

"The system was put into full scale operation within 90 days of conception."

One of the Model II's, equipped with an auto answer modem, receives the call in Fort Worth. The service center is required to log onto one of these computers by giving its store number and a password. This protection hinders illegal access to the system.

The TRS-80 Model II's that receive the calls from the service centers are in a host mode, which allows data to be transferred to them from a remote device (at this point the Model II at the service center is the remote device).

Prior to accepting the order from the service center, the hosting Model II sends a bulletin board to the terminal at the service center. Up to 10 general messages from the warehouse may be displayed on this bulletin board. The system also allows for one specific message to be sent to each individual service center. After all this has been viewed by the service center, the RAM buffer is transmitted down-line from the service center to the hosting computer. After the file has been transmitted, the hosting computer drops the terminal at the service center, and stores the transmitted file permanently on disk.

Two separate files are stored at the receiving end. One file contains any messages sent from the service centers. The other file contains the order which was placed by the service center.

Both files are now available to be printed out at the receiving end (Fort Worth) in several different formats. One of these formats include printing the sorted order file on a picking form, after which the order can easily be pulled and shipped.

This system was implemented and put into full scale operation less than 90 days after conception. By June of 1980, just 3 months after the system's installation, Tandy reported the system to be a complete success, and had already realized savings which far surpassed expenditures for software development. ●

Victor Andrews
&
Soft Sector Marketing Inc
Introduce...

B.O.S.S. 2.1 by V Hester

The Basic operated single stepper program for programs written in Level II or Disk Basic. Runs in all known operating systems, except CP/M.

Cassette.....$29.95
Disk.....$34.95

Monsters Lair by A Meir

A D & D based game. Create your own character or let the machine select one for you. Search for gold and fight monsters in their lair. This machine language game requires 16K free memory.

Cassette.....$14.95
32K Disk...$19.95

Space Colony by K Watt

Written for the PROGRAMMA Hi-resolution graphic board, this has the best graphics of any "space" game on the market today for the TRS-80. (Requires the Programma board) **$9.95**

TRS Space Invaders by Larry Ashmun

The ***last*** and greatest writing of Larry Ashmun. This is an upgrade of the game called Invaders Plus. It has Stereo sound and requires 32K disk. Ten levels of play and BLITZ! Write or call for upgrade information from your Invaders Plus program ***(only available from SSM)***. Available on disk only ...**$24.95**

Matchbook Football with sound .. **Cassette $9.95**

Capture/Symon ..... **Cassette $12.95**

All Scott Adams Adventures .... **Cassette $14.95**

NUFORMAT **$9.95**

NUPURGE **$19.95**

Lowercase driver program for Radio Shack and others.....**$7.95**

NEWDOS 80 ..... **$139.00**

VTOS 4.0 ..... **$89.00**

Disk drives 40 track w/power supply and case ..... $350.00 **Special! $299.00***

PMC-80 w/16K Level II $645.00 **Special! $560.00***

LEDex monitors 12 Inch..... **$139.00**

Disk drive MPI 80 track w/power supply & case ..... $550.00 **Special! $450.00***

6250 Middle Belt
Garden City, MI 48135
Phone 1-313-425-4020
Visit our new store!
Open Tuesday thru Saturday

***These prices are higher in our store.**

Buy any three programs at one time and deduct 10%! VISA/MC accepted. Please add 2% of sale or $2.50, whichever is higher for shipping. Add $1.50 for COD (COD must be paid for by certified check or cash). NO COD over $150.00. All items in stock at time this ad was placed.

A Basic Memory Saver

by Peter A Lewis
Dayton, OH

You are a good programmer. When you write your Basic programs, you always include a generous number of remarks to explain what different parts of your program are doing. You also like your listings to be readable, so you leave spaces between words. To make the logic of the program clear, you indent FOR....NEXT loops.

Now that the program is finished, you admire the beautiful listing, note that there are still a few hundred bytes left, and confidently type RUN and press ENTER.

Oh ho! You forgot that you need some memory for variables and string space! So there you sit, with an "?OM" message staring you in the face. With tears in your eyes, you start to delete some remarks and maybe eliminate a lot of blanks. Will you be able to understand the program six months from now?

Even if the program does run, dosen't it bother you to wait for all those remarks and blanks to be read in from tape? Think how much faster it would load without them. Is there any way that you can have your remarks without eating them too?

There is now! This handy little program, called "SQUEEZE", will give you the best of both worlds. Write your program with all the remarks and spaces you like. Even add some unnecessary words like LET. Then save the pretty version on tape. Once it's saved, you execute SQUEEZE, and the following actions will be performed:

-All blanks will be removed (except within quotes).

-All remarks will be removed.

-If a statement consists of *only* a remark that starts

with REM (rather than '), the entire statement will be deleted. If you want to reference remark statements in a GOTO, GOSUB, or a RESUME, begin the remark with an ' (apostrophe) or precede the REM with a colon. In this case, the remark will be removed but the empty statement will remain.

- The command LET will be removed (as in LET A=B+C)
- The command GOTO will be removed if it follows THEN or ELSE.

Once this is done, the SQUEEZEd program can be saved on tape to be used for execution. The pretty version is available for listing or updating.

Included here is an assembly listing of the program. If you have the EDITOR/ASSEMBLER, enter the assembly language statements exactly as shown. The ORG statement on line 180 is for a 16K machine. Adjust accordingly for larger machines. (48984 for 32K, 65368 for 48K).

If you have T-BUG, you can enter the machine language program using the M-command, and save it on tape with the P command. The program is relocatable and should be placed at the top of memory. Before loading the program with the SYSTEM command, be sure to set MEMORY SIZE at one less than the load address (32599 for 16K, etc.). Once the program is loaded, press "BREAK" to return to BASIC. Now you can write your program and CSAVE it or CLOAD an existing one. When you are ready to SQUEEZE, enter the following:

```
SYSTEM
/32600 (or whatever the load address was)
CLEAR
```

Now you are free to LIST, RUN, or CSAVE the compressed program. Note that upon return from the SQUEEZE program, the first statement will cause an "OM ERROR". After this first command, everything will be all right.

At this point, I should mention a minor bug that I discovered in Level II while I was testing the program. Consider the following statement:

```
IF A$ = "A"GOTO 30 ELSE 50
```

If A$ is not "A", the program will fall through to the next statement (the ELSE clause will be ignored). A blank, comma, or THEN inserted before the GOTO will fix this. This bug occurs whenever an IF clause ends with a quote and a command other than THEN immediately follows the quote with no intervening blank. Please note that this is a bug in the Level II interpreter, not in the SQUEEZE program. Because of this, you should always include the word THEN in any similar IF statements.

PROGRAM DETAILS

For you masochists who like to work with assembly language (like me), here is a short description of the program. First, a list of the register usage:

HL - points to the next character to be examined in the original program.

DE - points to the next location to receive a character of the compressed program.

IX - points to the location where the current statement header has been moved.

IY - contains the number of bytes deleted so far.

B - switch:

BIT 0 on (1) means that blanks are not to be deleted (within quotes).

BIT 1 on means that all the remaining characters in this statement are to be deleted (remark).

BIT 2 on means that at least one character has been retained in the current statement (if the end of the statement is reached with this switch still off, the entire statement is deleted).

BIT 3 on means that the previous non-blank character was a colon (the colon is deleted if the next character is a remark).

BIT 4 on means that the previous non-blank character was THEN or ELSE (if the next character is a GOTO, it is deleted).

To get an idea of how the program works, please locate the *Sept/Oct 1979 issue of 80-U.S.* and refer to the figure on page 36. (This is the article that inspired me to write this program). I'll wait while you get it...*

OK, now that you have that diagram in front of you, let's get into some detail on how the program works. See **Figure 2** for an overview of how a statement is compressed. I'll list groups of statement numbers with a general explanation of their purpose.

**(See Figure 1, it is a reprint from that issue -- Ed.)*

```
                00100 ; SQUEEZE BASIC PROGRAM - 10/13/79 - PETER A. LEWIS
                00120 ;
                00140 ; INITIALIZATION FOR START OF PROGRAM
                00160 ;
7F58            00180          ORG     32600
7F58 2AA440     00200 INIT     LD      HL,(40A4H)      ;SET "FROM" ADDR
7F5B E5         00220          PUSH    HL              ;SET "TO" ADDRESS
7F5C D1         00240          POP     DE
7F5D FD210000   00260          LD      IY,0            ;ZERO DELETE COUNT
                00280 ;
                00300 ; PROCESS HEADER
                00320 ;
7F61 D5         00340 STMNT    PUSH    DE              ;NEW HEADER LOCN
7F62 DDE1       00360          POP     IX
7F64 010400     00380          LD      BC,4            ;MOVE HEADER
7F67 EDB0       00400          LDIR
                00420 ;
                00440 ; CHECK FOR END OF PROGRAM
                00460 ;
7F69 DD7E01     00480          LD      A,(IX+1)        ;PTR MSB TO A
```

```
7F6C B7       00500          OR       A                ;IS IT ZERO?
7F6D 2872     00520          JR       Z,ENDPGM         ;YES - END OF PGM
              00540 ;
              00560 ; INITIALIZATION FOR NEW STATEMENT
              00580 ;
7F6F 0600     00600          LD       B,0              ;CLEAR SWITCH
              00620 ;
              00640 ; MOVE STATEMENT TO NEW LOCATION
              00660 ;
7F71 7E       00680 MOVE     LD       A,(HL)           ;GET CURRENT CHAR
7F72 12       00700          LD       (DE),A           ;MOVE TO OUTPUT
7F73 23       00720          INC      HL               ;BUMP "FROM" ADDR
7F74 B7       00740          OR       A                ;END OF STMNT?
7F75 2845     00760          JR       Z,ENDSTM         ;YES
7F77 CB48     00780          BIT      1,B              ;SKIP EVERYTHING?
7F79 203D     00800          JR       NZ,DELETE        ;YES
7F7B FE22     00820          CP       '"'              ;QUOTE?
7F7D 2833     00840          JR       Z,QUOTE          ;YES
7F7F FE93     00860          CP       93H              ;REM?
7F81 286F     00880          JR       Z,REMARK         ;YES
7F83 FE8C     00900          CP       8CH              ;LET?
7F85 2831     00920          JR       Z,DELETE         ;YES - DELETE IT
7F87 FE8D     00940          CP       8DH              ;GOTO?
7F89 2821     00960          JR       Z,GOTO           ;YES - MAY BE DELETED
7F8B CB40     00980          BIT      0,B              ;DELETE BLANKS?
7F8D 2018     01000          JR       NZ,NOSKIP        ;NO
7F8F FE20     01020          CP       ' '              ;BLANK?
7F91 2825     01040          JR       Z,DELETE         ;YES
7F93 CBE0     01060          SET      4,B              ;SET THEN-ELSE SWITCH
7F95 FECA     01080          CP       0CAH             ;THEN?
7F97 280E     01100          JR       Z,KEEP           ;YES - LEAVE SW ON
7F99 FE95     01120          CP       95H              ;ELSE?
7F9B 280A     01140          JR       Z,KEEP           ;YES - LEAVE SW ON
7F9D CBA0     01160          RES      4,B              ;RESET SWITCH
7F9F CB98     01180          RES      3,B              ;RESET COLON SWITCH
7FA1 FE3A     01200          CP       ':'              ;COLON?
7FA3 2002     01220          JR       NZ,NOSKIP        ;NO - LEAVE SW OFF
7FA5 CBD8     01240          SET      3,B              ;SET COLON SWITCH
7FA7          01260 NOSKIP   EQU      $
              01280 ;
              01300 ; KEEP THIS CHARACTER
              01320 ;
7FA7 13       01340 KEEP     INC      DE               ;BUMP "TO" ADDR
7FA8 CBD0     01360          SET      2,B              ;AT LEAST 1 CHAR KEPT
7FAA 18C5     01380          JR       MOVE             ;GET NEXT CHAR
              01400 ;
              01420 ; ELIMINATE "GOTO" AFTER "THEN" OR "ELSE"
              01440 ;
7FAC CB60     01460 GOTO     BIT      4,B              ;THEN-ELSE SW ON?
7FAE 28F7     01480          JR       Z,KEEP           ;NO - KEEP GOTO
7FB0 1806     01500          JR       DELETE           ;YES - DELETE GOTO
              01520 ;
              01540 ; REVERSE "BLANK-DELETE" SWITCH
              01560 ;
7FB2 78       01580 QUOTE    LD       A,B              ;SWITCH TO A
7FB3 EE01     01600          XOR      01H              ;REVERSE IT
7FB5 47       01620          LD       B,A              ;BACK TO B
7FB6 18EF     01640          JR       KEEP             ;KEEP QUOTE
              01660 ;
              01680 ; ELIMINATE CHARACTER
              01700 ;
```

```
7FB8 FD23      01720 DELETE  INC     IY              ;BUMP DELETE COUNT
7FBA 18B5      01740         JR      MOVE
               01760 ;
               01780 ; END OF STATEMENT
               01800 ;
               01820 ; MODIFY NEXT STATEMENT POINTER
               01840 ;
7FBC CB50      01860 ENDSTM  BIT     2,B             ;ANY DATA KEPT?
7FBE 2817      01880         JR      Z,NODATA        ;NO - DELETE STMNT
7FC0 E5        01900         PUSH    HL              ;SAVE HL
7FC1 FDE5      01920         PUSH    IY              ;IY TO BC
7FC3 C1        01940         POP     BC
7FC4 DD6E00    01960         LD      L,(IX)          ;NEXT STMNT ADDR
7FC7 DD6601    01980         LD      H,(IX+1)
7FCA B7        02000         OR      A               ;CLEAR CARRY
7FCB ED42      02020         SBC     HL,BC           ;ADJUST
7FCD DD7500    02040         LD      (IX),L          ;STORE NEW VALUE
7FD0 DD7401    02060         LD      (IX+1),H
7FD3 E1        02080         POP     HL              ;RESTORE HL
7FD4 13        02100         INC     DE              ;KEEP END STMNT INDIC
7FD5 188A      02120         JR      STMNT           ;NEXT STATEMENT
               02140 ;
               02160 ; DELETE ENTIRE STATEMENT
               02180 ;
7FD7 DDE5      02200 NODATA  PUSH    IX              ;BACK UP "TO" PTR
7FD9 D1        02220         POP     DE
7FDA 010500    02240         LD      BC,5            ;DROPPING 5 CHARS
7FDD FD09      02260         ADD     IY,BC
7FDF 1880      02280         JR      STMNT           ;NEW STATEMENT
               02300 ;
               02320 ; END OF PROGRAM
               02340 ;
7FE1 1B        02360 ENDPGM  DEC     DE              ;SET NEW TAIL POINTER
7FE2 1B        02380         DEC     DE
7FE3 ED53F940  02400         LD      (40F9H),DE
7FE7 ED53FB40  02420         LD      (40FBH),DE
7FEB ED53FD40  02440         LD      (40FDH),DE
7FEF C3191A    02460         JP      1A19H           ;RETURN TO READY
               02480 ;
               02500 ; PROCESS REMARK STATEMENT
               02520 ;
7FF2 CBC8      02540 REMARK  SET     1,B             ;SET DELETE SWITCH
7FF4 CB58      02560         BIT     3,B             ;PREV CHAR A COLON?
7FF6 28C0      02580         JR      Z,DELETE        ;NO
7FF8 1B        02600         DEC     DE              ;DELETE COLON
7FF9 FD23      02620         INC     IY              ;BUMP DELETE COUNT
7FFB 18BB      02640         JR      DELETE          ;DELETE CHARACTER
7F58           02660         END     INIT
00000 TOTAL ERRORS
DELETE 7FB8 01720   00800 00920 01040 01500 02580 02640
ENDPGM 7FE1 02360   00520
ENDSTM 7FBC 01860   00760
GOTO   7FAC 01460   00960
INIT   7F58 00200   02660
KEEP   7FA7 01340   01100 01140 01480 01640
MOVE   7F71 00680   01380 01740
NODATA 7FD7 02200   01880
NOSKIP 7FA7 01260   01000 01220
QUOTE  7FB2 01580   00840
REMARK 7FF2 02540   00880
STMNT  7F61 00340   02120 02280
```

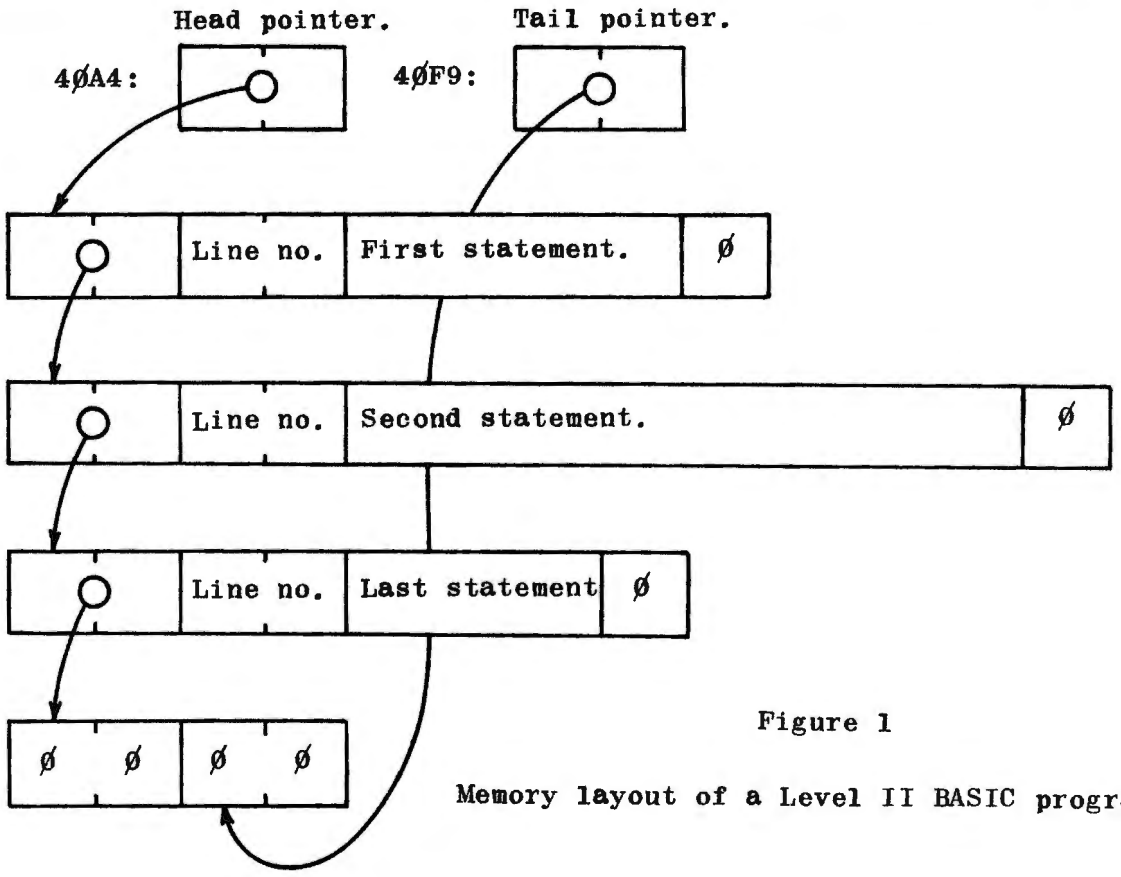


Figure 1

Memory layout of a Level II BASIC program.

BASIC MEMORY SAVER (Figure 2)

Here is the original statement:

```
10 LET A = B + 5 : REMARK
```

Here is what it looks like in memory:

```
          0 4 0 0 8       D       C           9         0
42E9      0 3 A 0 C A 5 B D 5 : 3 A R K       0
```

This is what it looks like after SQUEEZE:

```
          F 4 0 0   D   C   0
42E9      3 2 A 0 A 5 B D 5 0
```

And this is how it looks when you LIST it:

```
10 A=B+5
```

Statements	Purpose
200-240	Sets the address of the first header in registers HL and DE.
260	Initializes the count of characters deleted.
340-400	Executed for each new statement. IX is set to point to the new header after it is moved, then the 4-byte header is moved to the new location (always lower than or equal to the original location).

480-520	The MSB of the pointer to the next statement is checked for zero; if it is, the end of the program has been reached.
600	For each new statement, the switch (register B) is cleared.
680-760	Executed for each byte in the statement text. First, the next character is moved (if it turns out that this character is to be deleted, it will be overlayed by the next one). After incrementing the "FROM" register, the character just moved is checked for a zero. If it is, this is the end of this statement.
780-800	Checks to see if this character is within a remark. If it is, a jump is made to the routine that deletes it.
820-960	Checks for certain characters that require special treatment.
980-1000	If the switch is on indicating that the current character is between quotes, the rest of this section is bypassed.
1020-1040	If the current character is blank, it is deleted.
1060-1160	The result of these instructions is that if the current character is "THEN" or "ELSE", a switch is set which will cause the next non-blank character, if it is a "GOTO", to be deleted.
1180-1240	These instructions will cause a switch to be set if the most recent non-blank character, not between quotes, was a colon.
1340-1380	This routine causes the current character to be kept. This is done by incrementing the "TO" register so the next character will not replace this one. A switch is also set to indicate that at least one character was kept.
1460-1500	A "GOTO" character is processed here. If the previous non-blank character was a "THEN" or "ELSE", the "GOTO" is deleted.
1580-1640	When a quote is found, the setting of the switch is reversed. Since the switch is off at the beginning of the statement, an "ON" condition indicates that the current character is within quotes.
1720-1740	The current character is being deleted. This is done by incrementing the delete count and *not* incrementing the "TO" address. The next character will overlay this one.
1860-2120	This routine will adjust the header of the current statement so that it will point to the header of the next statement. This is done by subtracting the number of characters deleted so far from the header.
2200-2280	The statement is empty, so it is deleted. There are 5 characters in an empty statement: The "next statement" pointer (2 bytes), the line number (2 bytes), and the "end of statement" indicator (1 byte).
2360-2460	When the end of the program is reached, the tail pointers are adjusted to point to the new end of the program.
2540-2660	A remark character is processed by deleting it, the previous colon (if present), and setting a switch to cause the remaining characters in the statement to be deleted.

CONCLUSION

I hope you find this program as useful as I have. I feel very strongly that a program should have good documentation built in. The larger and more complicated a program is, the more important this becomes. This program will overcome the disadvantages of good documentation that exist with the use of BASIC. Happy programming! ●

by Bill Everett
Bellevue, WA

PURGE

PURGE is a Basic program written for the NEWDOS disk operating system. It allows the user to delete files or programs from a disk under program control. This program may be useful when deleting the many backup copies of a program that accumulate while developing software. Single disk drive owners will find it handy when creating backup disks with NEWDOS. Since all the utility programs are not needed (and leave little space for user programs), this program can delete all the unwanted files from a disk. You may use the auto command to run PURGE (AUTO BASIC RUN"PURGE").

```
100 REM PURGE
110 'BY  BILL EVERETT
         14645 NE 34TH #C-24
         BELLEVUE, WA  98007
         (206) 883 8475
120 'DATE LAST MODIFIED 02-16-80
130 CLEAR 500
140 C=15488:'SCREEN MEM LOC OF 1ST LETTER OF 1ST FILE
150 B$=STRING$(60," ")
160 C$="DO YOU WANT TO DELETE "
170 ON ERROR GOTO 620
180 CLS: PRINT@384,;
190 INPUT "WHICH DRIVE NUMBER DO YOU WISH TO USE (0 TO 3)";A
200 A=INT(A): IF A<0 OR A>3 THEN 180
210 G$=":"+RIGHT$(STR$(A),1)
220 F$="DIR "+G$
230 CMD F$
240 G=C
250 REM READ FILE NAME
260 A$="": E=0: F=G
270 D=PEEK(F)
280 IF D=32 THEN 320 : '32=SPACE
290 A$=A$+CHR$(D)
300 F=F+1
310 GOTO 270
320 D=PEEK (F+3)
330 IF D=80 THEN E=1: '80=P FOR PASSWORD
340 IF A$="" THEN GOSUB 590 ELSE 410
350 PRINT@896, "THATS THE LAST FILE";
360 FOR X=1 TO 1000: NEXT
370 CMD F$
380 PRINT: PRINT: CMD"FREE
400 END
410 REM KILL FILE
420 GOSUB 590
430 PRINT@832,C$;A$;
440 INPUT D$
450 IF LEFT$(D$,1)="Y" AND E=0 THEN 480
460 IF LEFT$(D$,1)="Y" AND E=1 THEN 500
470 GOTO 530
480 KILL A$+G$
490 GOTO 530
500 PRINT@896,"WHAT IS THE FILES PASSWORD";
510 INPUT E$
520 KILL A$+"."+E$+G$
530 D$=""
540 B=B+1
550 IF B=30 THEN B=0: C=15488: GOTO 230
560 G=G+20
570 IF G>C+40 THEN C=C+64: GOTO 240
580 GOTO 250
590 REM CLEAR LAST THREE LINES
600 PRINT@768,B$: PRINT@896,B$
610 RETURN
620 RUN
```

How it works

PURGE displays the disk directory on the screen and then uses PEEK to look at screen memory to construct the file names from the directory display. Each file name is displayed and you are asked if you want to delete it. An answer of "Y" will cause the file to be killed. Any other answer or just ENTER will cause the next file name to be displayed.

PROGRAM DESCRIPTION

130 Initializes string storage area

140 Stores 15488 in C which is the screen memory location for the first letter of the first file name in the directory.

150	Sets B$ to 60 blank spaces.
170	Reruns the program if an error occurs.
180-200	Asks what drive you wish to delete files from.
210-230	Displays the directory of the disk selected in 190.
240	Sets G to the value from line 140.
260	Initializes variables. A$=File name, E=Flag for file with password (0=No password and 1=password), and F=Current memory position on the screen that the program is looking at.
270	Get the ASCII equivalent of a character in the file name.
280	If the character is a space go to 320.
290	Add the character to the ones already read in the file name.
300	Add one to the address you are looking at to read the next character of the file name.
310	Loop back to read the next character.
320-330	Looks at the space three over from the last letter in the file name. If it is a "P" set the password flag (E = 1).
340	If A$ does not contain a file name go to the subroutine at 590 and return to line 350 otherwise go to line 410.
350-380	Close the program. Display the revised directory. Display the free space on the disk.
400	end program execution.
420	Clear three lines at the bottom of the screen.
430-440	Asks if you want to delete the current file.
450	If you answer 'Y' and there is no password go to 480
460	If you answer 'Y' and there is a password go to 530.
470	Go to 530.
480-490	Kill the file and go to 530.
500-520	Ask what the file's password is and kill the file if a correct answer is given.
530	Deletes the yes or no answer from D$.
540-550	If more than thirty file names have been looked at reinitialize the screen pointer and go to 230.
560	Move the pointer for the first letter of a file name over one column.
570-580	If the end of the line of file names has been reached move the pointer to the beginning of the next line and go to 250.
590-610	Clear to blank three lines at the bottom of the screen.
620	Rerun the program. This line is jumped to from 170.

Other Possibilities

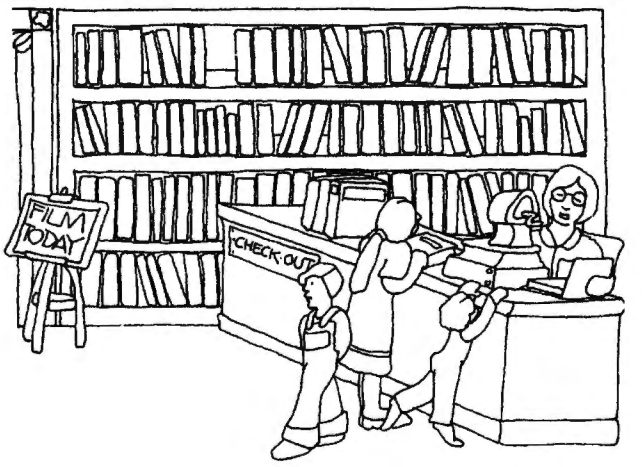
This technique could be used to write a program that would look at a number of different disk directories and make a master directory for all of them. It could also be used to search a number of directories looking for a single file or deleting backup copies of the same file. The possibilities are only limited by the information that is contained in the directory.

Limitations

This program will only work with the NEWDOS disk operating system. It will not work with TRSDOS 2.1, 2.2, or 2.3. I have not tried it with any other operating systems.

If there are more than thirty files on the disk directory the ENTER must be depressed during the directory display to allow the program to continue to run. If there are more than thirty files on the directory that you want to save then PURGE may not be able to delete all of the files you would wish to kill. The program won't crash under any set of circumstances. ●

Book Reviews



The BØØK

Volume I

Insiders Software Consultants
P.O. Box 2441
Springfield, VA 22152
$14.95 + $1.50 postage

Let's see. If I add the HL and DE, then the carry can go into shifting the B around to...no, I've got to shift the B into the D first, then use the carry to flag the inversion subroutine...except that I need to get the IX out of the stack, and it's about 3 levels down...wait a minute. Maybe if I try to decrement the HL enough times then I won't have to...

How many times have you been through this? It seems like every time you try to do any serious math in machine language, the whole earth turns against you in an attempt to keep you from doing it. It's especially frustrating when you realize that up there, unreachable, lies 12K of Microsoft Basic in ROM that would be more than happy to do all this math for you, if you only knew how to communicate with it.

Enter THE BOOK. 87 pages of just what you're looking for.THE BOOK explains how math is performed inside your S-80, and how to interface with the ROM.

Chapter one explains the terms used to describe the various types of variables used in BASIC, how they are stored, and how to find them. A very detailed discussion of how floating point numbers are generated is included. This includes procedures such as normalization, exponentation, and "excess 80 code". After the reader is thoroughly familiar and comfortable with these concepts (several examples and sample problems are given), the "memory accumulators" (which is where numbers are stored while they are being worked with) are described and their locations given, and the "EDCB" format for numeric storage is described. (If any of this doesn't make sense to you, then you need this book.)

Chapter two begins the discussion of data manipulation techniques to be used when interfacing with the ROM. The "TYPEFLG" byte is discussed, along with ways to test and manipulate it. Included in this chapter is an error recovery routine, so you can save it in case some kind of error should occur during ROM processing. Certain initialization routines that must take place are also discussed. These include routines for single precision division and the RND function. There are also listings for routines to convert from one data type to another (from single precision to double, double to integer, etc.), and routines for moving data around in memory. There are also descriptions of the ways of getting data back and forth to the user, including ASCII to binary, binary to ASCII, and print formatting (using the PRINT USING routines).

Finally, chapter 3 gets into the meat of the matter, assembly language interfacing with the ROM. This chapter tells how to get ROM to add, subtract, multiply and divide using all three variable types. There are several routines in the ROM to do this, since each variable type is handled differently, and THE BOOK describes how to use each of them in plain, easy to understand language. Each of the routines is accompanied by a source listing to make life just a little easier.

Also included in chapter three are those most elusive of dreams, the trig functions. Routines are given that will allow access to ABS, ATN, COS, EXP, FIX, INT, LOG, RND, SGN, SIN, SQR, TAN, and (up-arrow). Again, a source listing is provided for each case.

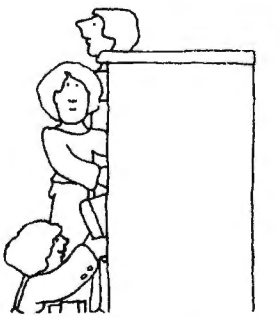
Chapter 4 is a series of disassembled listings of the ROM math routines. Due to copyright problems, a complete listing isn't possible, but instead, the instructions without operators are given, with comments added. Therefore, the instruction:

```
SYMBOL ADD HL,DE ;COMMENT
```

becomes:

```
SYMBOL ADD    ;COMMENT
```

This is really only a very minor inconvenience, compared to trying to figure the thing out yourself. If you want to get the entire thing, use any good disassembler to produce a listing, then

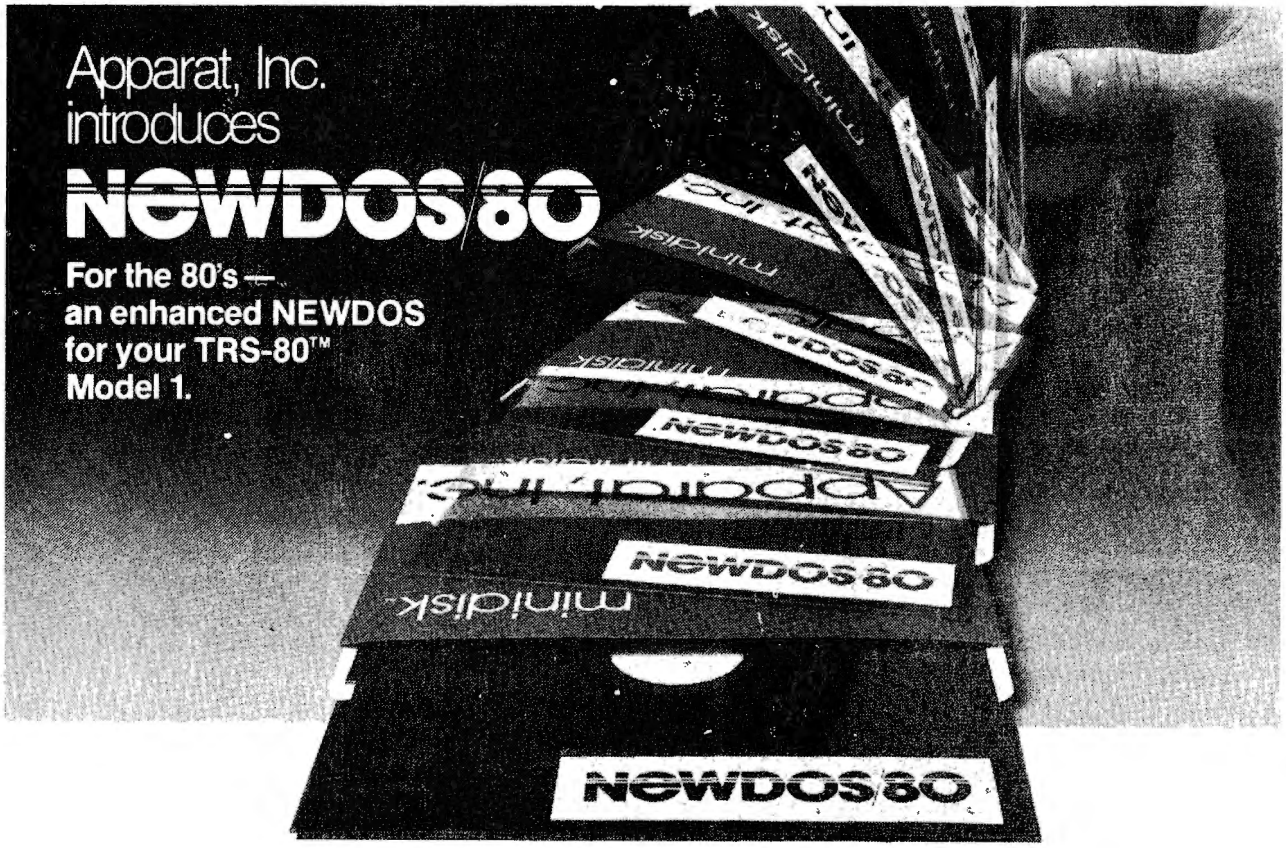


copy the comments over. You will probably want to do this anyway, since Volumes 2 and 3 of The Book promise to hold as much information as this one. Also, a listing of the ROM provides study material for budding programmers.

After this come 3 appendices; the "label table" (an incredibly complete map of the ROM and reserved areas of RAM, showing the addresses of various key routines), and a listing of a disassembler in BASIC.

THE BOOK is illustrated by Infinity Graphix, who's art is both funny and illustrative. Liberal sprinklings of humor throughout help make the rather dry world of high-powered machine language programming just a little bit easier to swallow. Overall, on a scale of 1-10, I'd give this one about a 9. The only reason I wouldn't give it a 10 is because it should have come out 3 years ago when I was first trying to figure the ROM out. When I think of the hours I spent with paper spread all over the living room floor... The only other problem I see with it, would be the new ROMs (the ones that come up MEM SIZE? instead of MEMORY SIZE?). Radio Shack keeps warning us that "undocumented" addresses in the ROM have been changed, but with the disassembled listings and disassembler in hand, you should have no problem finding what you need.

The introduction of The Book promises that it "will detail the operation of the Level II ROM". This is volume one of a proposed three volume set, and if volumes 2 and 3 are anywhere near as good as volume 1, they will have kept that promise quite well. The last page of volume 1 was a coupon for $3.00 off the price of volume 2. At least it was until I sent it in.. **J Crocker**



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

Apparat, Inc.

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

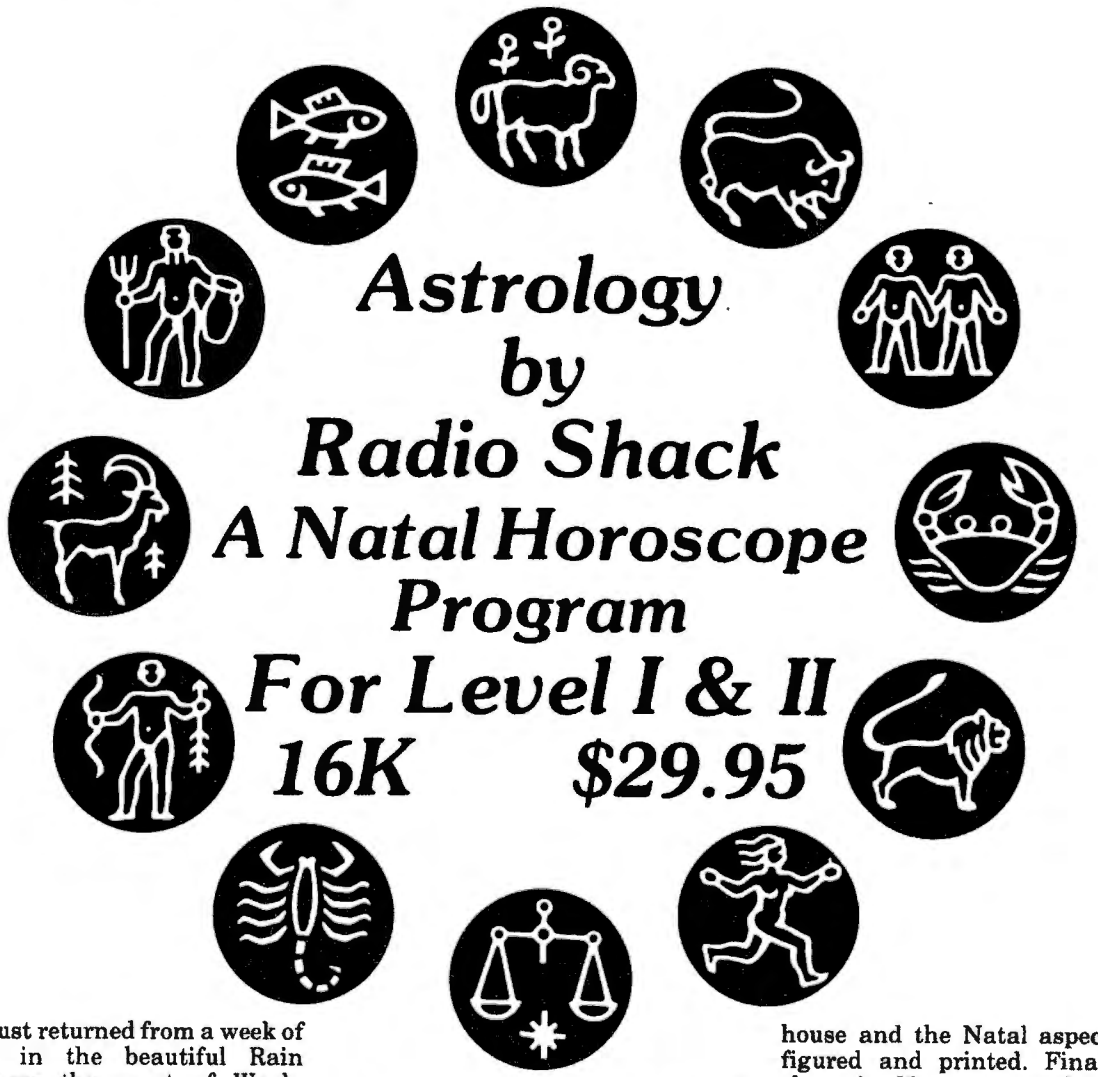
Name _____

Address _____

City _____ State _____ Zip _____

Phone _____

Software Review



Astrology by Radio Shack
A Natal Horoscope Program
For Level I & II
16K $29.95

I had just returned from a week of vacation in the beautiful Rain Forest near the coast of Washington, deep in the thick, green vegetation with no people, no electricity - just peace and quiet.

Peace and quiet were the last things on my mind though, as I headed for 80-U.S. I was excited because Mike Schmidt had left a message to come and see him. He had his hot hands on Radio Shack's Astrology program. I am an Astrologer, and this is what I had been waiting for ever since I purchased my TRS-80. I had been trying to come up with a good program for Astrology, but had gotten nowhere until now.

When I got there, Mike handed me a package, and like the nice, sensitive Pisces he is, didn't mention my aromatic, woodsy smell.

The package contained a cassette tape, a 33 page booklet of instructions, and a large, beautiful wall chart. The chart is very impressive, and contains a lot of Astrological information.

Shaking with excitment, (and dropping a few brown leaves from my clothes in the process) we loaded the tape. It took a few minutes to load, and then he entered a few bits of information, and presto! - his printer started printing.

What a printout! There was 32 inches of Astrological information, starting with a heading and the rising sign. It gave the retrograde information on each planet, Elements and Nodes (fire, water, fixed, etc.) and then the house placement for each planet.

There followed a section for the sign and degree for the cusp of each house and the Natal aspects were figured and printed. Finally, the chart itself was printed, about 9 inches high, with lots of space for adding more information like the transiting planets.

Next, we tried my birthdate, and in the three minutes it took to print, a few small chunks of dead moss fell from my hair. This was really exciting!

A review? But of course! I left the office of 80-U.S. with the package in hand, and smiled as I closed the door, wondering how long the aroma of Alder-wood campfire would linger.

A bath, change of clothes, and a quick shuffle of camping gear brought me to my TRS-80. I spent the rest of the day working with the program, testing, doing charts, comparing and was just amazed at what it did.

I found that if I had the correct

Programming the Z-80

By Rodnay Zaks
SYBEX Books
2344 Sixth Street
Berkeley, CA 94710
(In Europe: 18 Rue Planchat
75020 Paris, France)
Book C 280

Let's face it. Computer manuals are for people who don't need them. Most all of them start out assuming that you already know how to program, and you just need a reminder now and then.

Tutorials are even worse. They never seem to start out where you are, and you either feel left behind or like you have just wandered into a kindergarten class again. And, as if that weren't enough, if you do find one that actually helps, you go through it once and it ends up wasting space on your shelf because it isn't good for use as a reference manual.

Fortunately, there are exceptions to the rules, and *Programming the Z80* is one of the exceptions. Starting out with information that is designed for the beginner, yet not on a level that is insulting to the "old hand", this book will take you through the ins and outs of that wonderful chip of ours, the Z80.

With its 11 chapters, 7 appendices, and well over 100 drawings, charts, illustrations and program listings, this book ends up having 624 pages of information that is destined *not* to end up gathering dust on some shelf.

Chapter 1 begins with the rudimentary basics of programming, explaining such buzzwords as algorithm, coding, debugging, and other words you can use to convince folks that you really do know what you are doing (which, by the way, you will). Flowcharting is explained, with examples, along with the ways that numbers can be represented within the computer. Dr Zaks truthfully begins at the beginning, assuming no prior knowledge on the part of the reader. Don't let this put you off just because you have been doing this for a while. After all, who among us couldn't do with a little brush-up every now and then? Besides, no one said you *had* to read the first chapter. It's just strongly recommended. Every new idea that is brought up includes an example or two, along with a couple of test problems to work out for yourself. Every effort is made to make sure you understand and are comfortable with each concept before a new one is introduced.

Chapter 2 begins with the discussion of the hardware organization of the Z80 Central Processing Unit (CPU) and the basic computer system. The concepts of Arithmetic and Logic Unit (ALU), Registers, Shifting, addressing, stack and all the other strange things that go on inside that black box with 40 legs, are detailed and explained. Descriptions of how data gets from one place to another within the CPU help to make it seem a little less like magic, and a little more like the logical and easy to understand device that it really is.

Although this isn't a book on computer hardware, several difficult-to-grasp concepts are explained in such a way that you should have no trouble with them. These concepts include machine cycle, T states, fetch/execute, data bus, and more. Understanding how (and why) the CPU is structured can be a great help in implementing the most efficient code possible. Dr Zaks even goes so far as to include a chart explaining exactly what goes on inside the computer during each T state of every M cycle of most of the Z80's instructions. (The chart is actually an 8080 product, but the 8080's instructions are the same as the Z80's, only there are fewer.) This information comes under the heading of "nice to know", but I am of the belief that the more you know about your hardware, the easier the software becomes. Further examination of the hardware includes a pinout diagram of the actual chip with explanations of each pin (or group of pins, as in the 16 address lines) and its function.

Chapter 3 begins basic (*not* BASIC) programming techniques. This chapter begins the actual business of getting the computer to do something for us. The elementary skills learned in this chapter include 8-bit addition, 16-bit addition, 16-bit subtraction, binary coded decimal (BCD) arithmetic, multiplication and division (the Z80 has no built-in instructions for multiply or divide, and each must be programmed in a painstaking, step-by-step manner), logical operations (AND, OR, XOR, compare, etc.), and subroutines. Each skill includes well thought out examples, along with one step at a time explanations of what is taking place and why. Flow charts and memory diagrams make each new skill easier to grasp and hold onto. There is even a step-by-step tracing of a multiplication routine so you can follow each of the registers through the gyrations they must take.

Chapter 4 takes us into the instruction set of the Z80 (these are the instructions that the CPU is capable of executing). First, the five classes of instructions available to most computers are defined and briefly discussed, then we get into the actual set of Z80 instructions. These discussions are broken up into the same categories as the previous discussions. The ways that instructions are stored in memory are discussed, along with the capabilities and limitations of each class of instruction. The Z80's flag (F) register is also discussed, describing each of the individual flags and their uses. A quite handy chart of the flags and how they are affected by various instructions is included. Finally, there is a 247 page section with detailed descriptions of each of the Z80's instructions. These descriptions include the binary code of the instruction, a short explanations of how it works, an internal data flow diagram, a timing chart (i.e., how long does this instruction take?), a chart describing affects on the flags, and an example.

Chapter 5 discusses addressing techniques. The Z80 has several options as to how an address is accessed. The explanations are quite clear without being so wordy as to get confusing. As usual, several diagrams help to keep things in order.

Chapter 6 is Input/Output (I/O) techniques, and Chapter 7 talks about I/O devices. The Z80's three maskable (which means you can choose whether or not to accept them) interrupt modes and its non-maskable (that's the reset button to us Model I users) are discussed here. The I/O devices introduce a special set of integrated circuits invented by ZILOG especially for the Z80. These include the Programmable I/O (PIO) and the Serial I/O (SIO). These two chips can make hardware design much more simple.

Chapter 8 gets into some rather advanced programming, and by this time you should be ready for them. Directly useable programs and subroutines listed here include zeroing a block of memory, I/O device polling, character "bracket testing" (checking to see if a character fits within a specified range), ASCII to BCD conversion, hexadecimal to ASCII conversion, and a machine language bubble sort. These are programs that you will be able to use right away, or modify to fit your own special needs.

Chapter 9 is devoted to data structures. Chapter 10 is about program development and Chapter 11 is a bit of conjecture on the author's part about the future.

The appendices include two hexadecimal conversion tables (decimal to hex and hex to decimal), an ASCII character code table, relative branch tables (these get used a lot), decimal to BCD conversion table, a table of the hex codes for the Z80 instructions and conversions from 8080 to Z80 and Z80 to 8080.

Last but not least, the index of this book makes it very simple to find a particular subject that you want to look up.

In the final analysis though, if someone were to ask me for one reason to buy this book, I would have to recommend it because it was designed to be *read*. It is not a computer manual, nor is it funny. It is simply well thought-out, fascinating reading. If you are truly interested in learning to use your machine to its fullest capability, this is a book you should have.

J Crocker

birthdate information and entered it correctly, I could go through a whole chart in less than a minute. I do not have a printer yet, but the information can be viewed on the screen, very quickly and accurately.

You have a choice of having the information printed, or on the screen. In the screen-only mode, the actual chart is not shown due to screen size restrictions. All the necessary information is there though, and it is really nice to have a chart printed and ready to use. The printout takes about 3 minutes, while going by a chart on the screen takes under one minute. Can you imagine the early fathers of Astrology, thousands of years ago using this? Ptolemy or Hippocrates would have driven themselves crazy!

As you can probably see by now, this is a program for Astrologers or students of Astrology. The program gives only information necessary to erect a Natal chart using the Placidean system to determine the house cusps, and gives no delineations.

The instruction book which comes with the tape, besides the usual Radio Shack instructions on loading, gives only the basics of Astrology in their briefest form. You should be able to draw a chart and use an ephemeris easily before this program will mean something to you.

The program is loaded in machine language, and is recorded twice on each side, Level I 16K on one side and Level II on the other. It takes almost the whole 16K, and takes about four minutes and 15 seconds to load.

Ptolemy or Hippocrates would have driven themselves crazy!

Once the machine is loaded though, look out - because you have a tremendous tool at your disposal. It is accurate, very fast, and absolutely unbelievable.

Since my first encounter with it, I have put the program on my machine every morning and have used it extensively.

One day as a final test, I took 25 charts I had drawn up over the years, and fed them into the machine. The results were embarrassing! I failed the test! Out of 25 charts, the program showed me 3 errors of my own doing, and after checking the books, I even had to redraw one chart. The program had caught my error and had given me the correct information in less than a minute!

I am very impressed with the program, irritated by slow cassette loading and elated with the wall chart which comes with it. This is not a fun program, it is a serious workable tool for an Astrologer to use. The potential for research is unlimited. I have already had several ideas, various dates and times I want to run through, but what's that? The smell of an Alderwood campfire? Oh-oh, time for one last fling in the woods before winter comes. When I get back, I plan to use this program much more extensively. *Now then, the ax, the flashlight, oh yes, there're my boots..*

Richard D Frasier

HOMES for the TRS-80

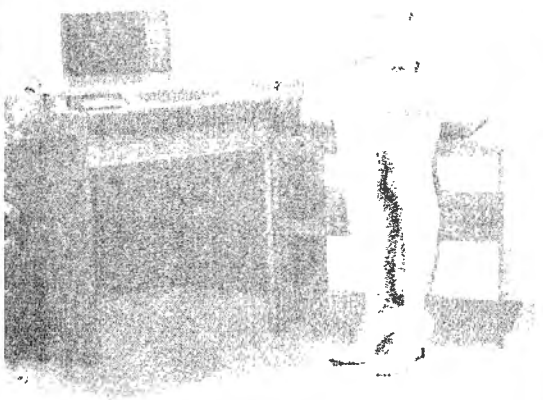
Featuring

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.

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

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.*

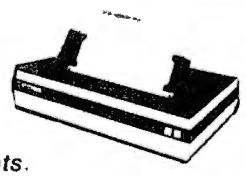
List $395.00

The EXATRON Stringy Floppy Mass Storage System

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

List $249.50

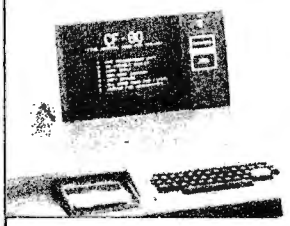
The BASE II Graphic Line Printer



- *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*

80-U.S. Journal

Index to Vol III

Jan 1980 thru Dec 1980

UNCLASSIFIED ADS

$2.50 per half inch and $2.50 for each additional half inch per insertion. Ads must be typewritten or printed clearly, and must be accompanied with payment in advance. No telephone orders or "bill me's" will be accepted. Unclassified Ads are always set in the type on this page, with the first line in bold type. Use the ads on this page as a guide to figure your size and payment. Send to: 80-U.S. Journal, 3838 S. Warner St., Tacoma, WA 98409

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)

PYRAMIND-CHALLENGING GAME OF logic and deductive reasoning. Can you solve the Pyramind Code? A different code with each game. With sound, graphics, scoring and more. L2 16K required. $11.95 on cassette. EDU-WARE, PO Box 336, Maynard, MA 01754 (110)

THE AMAZING BLACKJACK MACHINE. Starts where blackjack books end! For persons familiar with blackjack TABM is designed for the development, study and analysis of blackjack playing, counting strategies and betting systems. Requires 16K Level II. $25.00 on cassette only. CASINO GAMBLING WORKSHOP, Box 562, Toms River, NJ 08753 (201) 349-2799 (110)

ASTROLOGY..FOR FUN OR PROFIT..ASTRO LEARN is 2 16K Level II programs that teach you chart interpretation. Graphics, tests etc. For only $12.95! We also have amazingly accurate calculating programs. Natal charts, progressions, transits, composite, solunars, etc. ASTRO-LEARN PO Box 788 Lakewood, CA 90713 (110)

FOR TRS-80 RS-232 BOARD. FIX INTER-mittent 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)

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)

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)

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)

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)

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)

SCRIPSIT MODIFICATION: Converts Scripsit to work with non-Centronics printers. Now you can send extra linefeeds between text, and vertically center your text. Write for information or send a copy of Scripsit on disk (upper or lower case version) and a check for $10.00 to James L Moody, E. 101 Augusta, Spokane, WA 99207 (110)

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)

MATH-PAK 1 - MATH-PAK 2 The highly recommended math drill programs that let the user do the problems one step at a time. Both MATH-PAKS cover addition, multiplication, subtraction and division. Some of the MATH-PAKS features are: Immediate error check; user selected difficulty levels; auto digital positioning; 100 point scoring; carryovers, remainders (PAK 1); full four function calculator, simplification, reducing, and flashing numerator (PAK 2); games; documentation and more. MATH-PAK 1 for whole numbers, MATH-PAK 2 for fractions. $14.95 each (L2 16K required). EDU-WARE PO Box 336, Maynard, MA 01754 (110)

TRS-80 SUPER GAMES FOR 16K Level II. 10 exciting & challenging games on C90 cassette just $5.00, cash only. Castlestar Products, Dept CS-40, 8753 Windom Ave., St Louis, MO 63114 (011)

NIGHT FIGHTER: Exciting real-time simulation of a night interception mission in WWII. The ground controller directs you to the target, you acquire him on your cockpit radar, and the chase is on. Your aircraft can climb, dive, turn, speed up, slow down - but so may the German bomber. Program combines Basic, packed graphics, and machine language to speed action and enhance displays. Three levels of difficulty. On tape ($10.00) or disk ($15.00). Specify L2 or Disk, MEMSIZE. Rod Fitzgerald, Box 12709 Dallas, TX 75225 (011)

THE ERCC NEWSLETTER IS PUBLISHED monthly by Educational, Recreational Computer Club. Membership in ERCC costs $5.00 including a subscription to the newsletter. Club address is ERCC, Box 325 Owosso, MI 48867. The club meets the second Saturday of each month to discuss ideas, trade programs and to review new computer products. ERCC maintains a newsletter exchange with other clubs. To join the exchange, simply send us your newsletter. Non-profit computer clubs are free to reprint material from this newsletter, giving appropriate recognition to the source. The club meets at the Pines Country House Restaurant & Motel Banquet Room "A". Program time 5:00 PM, Dinner 6:00 PM. Address is 1730 E Main, Owosso, MI (517) 725-5164 OR Fly to the Owosso Airport, then taxi & park right at the Pines. (011)

FOURTEEN TRS-80 DISK BASIC
Programs for a buck apiece. Global Search & Replace utility, Travel Expense Report, Appointment Secretary, Construction Materials Cost Calculator, Programmable Audible Process Timer, Television Guide, Auto Cost Calculator, Magazine Subscription Records, Record Album Timing. Games: Donut Shop, Card Sharks, Phrase Challenge, Star Wars, Market. Supplied on disk: $13.94+ $1.00 postage. Busch, 515 E Highland Ave, Ravenna, OH 44266 (011)

TRS-80 TO H-14 PRINTER INTERFACE — NO software driver required! Print at 4800 baud with full handshaking & printer-status support. Interface connects to parallel printer card-edge of Radio Shack 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 Radio Shack 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, or send $89.95 plus $3.50 shipping & handling. (011)

Back Issue Availability

The following back issues of The 80-U.S. Journal are still available: July/Aug 1979, Nov/Dec 1979, Jan/Feb 1980, May/Jun 1980. Price of back issues is $3.00, postpaid. Please allow 3 to 4 weeks for delivery.

ADVERTISERS INDEX

TRS-80 Model I & II Software

(Exquisitely documented)

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 (West)
80-U.S. Software
Dept 80-F, 3838 South Warner Street
Tacoma, WA 98409 (206) 475-2219

Distributor (East)
Acorn Software Products Inc
Dept 80-F, 634 North Carolina Ave SE
Washington, DC 20003 (202) 544-4259

For further information, please contact the distributor nearest you.

★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

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)