

MICRO™

THE 6502/6809 JOURNAL



PET Feature

Growing Knowledge Trees

6502 and 6809 Memory Moves

Apple Printer Utilities



Turn your Apple into the world's most versatile personal computer.

The SoftCard™ Solution. SoftCard turns your Apple into two computers. A Z-80 and a 6502. By adding a Z-80 microprocessor and CP/M to your Apple, SoftCard turns your Apple into a CP/M based machine. That means you can access the single largest body of microcomputer software in existence. Two computers in one. And, the advantages of both.

Plug and go. The SoftCard system starts with a Z-80 based circuit card. Just plug it into any slot (except 0) of your Apple. No modifications required. SoftCard supports most of your Apple peripherals, and, in 6502-mode, your Apple is still your Apple.

CP/M for your Apple. You get CP/M on disk with the SoftCard package. It's a powerful and simple-to-use operating system. It supports more software than any other microcomputer operating system. And that's the key to the versatility of the SoftCard/Apple.

BASIC included. A powerful tool, BASIC-80 is included in the SoftCard package. Running under CP/M, ANSI Standard BASIC-80 is the most powerful microcomputer BASIC available. It includes extensive disk I/O statements, error trapping, integer variables, 16-digit precision, extensive EDIT commands and string functions, high and low-res Apple graphics, PRINT USING, CHAIN and COMMON, plus many additional commands. And, it's a BASIC you can compile with Microsoft's BASIC Compiler.

More languages. With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or

Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

Seeing is believing. See the SoftCard in operation at your Microsoft or Apple dealer. We think you'll agree that the SoftCard turns your Apple into the world's most versatile personal computer.

Complete information? It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us.

SoftCard is a trademark of Microsoft. Apple II and Apple II Plus are registered trademarks of Apple Computer. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc.

MICROSOFT

CONSUMER PRODUCTS

MICROSOFT Inc.

10700 Northup Way • Bellevue, WA 98004



JUDGE THE REST, THEN BUY THE BEST

Only GIMIX offers you **SOFTWARE SWITCHING** between MICROWARE's OS-9 and TSC's FLEX. Plus you get the power of the GMXBUG system monitor with its advanced debugging utility, and memory manipulation routines. A wide variety of languages and other software is available for these two predominant 6809 Disk Operating Systems.

You can order a system to meet your needs, or select from the 6809 Systems featured below.

JUDGE THE FEATURES AND QUALITY OF GIMIX 6809 SYSTEMS

GIMIX' CLASSY CHASSIS™ is a heavyweight aluminum mainframe cabinet with back panel cutouts to conveniently connect your terminals, printers, drives, monitors, etc. A 3 position keyswitch lets you lock out the reset switch. The power supply features a ferro-resonant constant voltage transformer that supplies 8V at 30 amps, + 15V at 5 amps, and - 15V at 5 amps to insure against problems caused by adverse power input conditions. It supplies power for all the boards in a fully loaded system plus two 5 1/4" drives (yes! even a Winchester) that can be installed in the cabinet. The Mother board has fifteen 50 pin and eight 30 pin slots to give you the most room for expansion of any S550 system available. 11 standard baud rates from 75 to 38.4K are provided and the I/O section has its own extended addressing to permit the maximum memory address space to be used. The 2 Mhz 6809 CPU card has both a time of day clock with battery back-up and a 6840 programmable timer. It also contains 1K RAM, 4 PROM/ROM/RAM sockets, and provides for an optional 9511A or 9512 Arithmetic Processor. The RAM boards use high speed, low power STATIC memory that is fully compatible with any DMA technique. STATIC RAM requires no refresh timing, no wait states or clock stretching, and allows fast, reliable operation. The system includes a 2 port RS232 serial interface and cables. All GIMIX boards use gold plated bus connectors and are fully socketed. GIMIX designs, manufactures, and tests in-house its complete line of products. All boards are twice tested, and burned in electrically to insure reliability and freedom from infant mortality of component parts. All systems are assembled and then retested as a system after being configured to your specific order.

56KB 2MHZ 6809 SYSTEMS WITH GMXBUX/FLEX/OS-9 SOFTWARE SELECTABLE

With #58 single density disk controller \$2988.59
 With #68 DMA double density disk controller \$3248.49
 to substitute Non-volatile CMOS RAM with battery back-up, add 300.00
 for 50 Hz export power supply models, add 30.00

Either controller can be used with any combination of 5" and/or 8" drives, up to 4 drives total, have data recovery circuits (data separators), and are designed to fully meet the timing requirements of the controller I.C.s.

5 1/4" DRIVES INSTALLED IN THE ABOVE with all necessary cables

	SINGLE DENSITY		DOUBLE DENSITY		
	Formatted	Unformatted	Formatted	Unformatted	
40 track (48TPI) single sided	199,680	250,000	341,424	500,000	2 for \$700.00
40 track (48TPI) double sided	399,360	500,000	718,848	1,000,000	2 for 900.00
80 track (96TPI) single	404,480	500,000	728,064	1,000,000	2 for 900.00
80 track (96TPI) double	808,960	1,000,000	1,456,128	2,000,000	2 for 1300.00

Chart shows total capacity in Bytes for 2 drives.

Contact GIMIX for price and availability of 8" floppy disk drives and cabinets; and 5" and 8" Winchester hard disk system.

128KB 2Mhz 6809 DMA Systems for use with TSC's UNIFLEX or MICROWARE's OS-9 Level 2

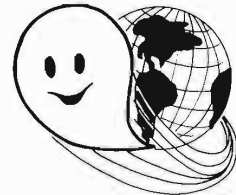
(Software and drives not included) \$3798.39
 to substitute 128KB CMOS RAM with battery back-up, add 600.00
 for each additional 64KB NMOS STATIC RAM board, add 639.67
 for each additional 64KB CMOS STATIC RAM board, add 988.64
 for 50 Hz export power supply, add 30.00

NOTE: UNIFLEX can not be used with 5" minifloppy drives.

GIMIX has a wide variety of RAM, ROM, Serial and Parallel I/O, Video, Graphics, and other S550 bus cards that can be added now or in the future. Phone or write for more complete information and brochure.

THE SUN NEVER SETS ON GIMIX USERS

GIMIX Systems are found on every continent, except Antarctica. (Any users there? If so, please contact GIMIX so we can change this.) A representative group of GIMIX users includes: **Government Research and Scientific Organizations** in Australia, Canada, U.K., and in the U.S.; NASA, Oak Ridge, White Plains, Fermilab, Argonne, Scripps, Sloan Kettering, Los Alamos National Labs, AURA. **Universities:** Carleton, Waterloo, Royal Military College, in Canada; Trier in Germany; and in the U.S.; Stanford, SUNY, Harvard, UCSD, Mississippi, Georgia Tech. **Industrial users** in Hong Kong, Malaysia, South Africa, Germany, Sweden, and in the U.S.; GTE, Becton Dickinson, American Hoechst, Monsanto, Allied, Honeywell, Perkin Elmer, Johnson Controls, Associated Press, Aydin, Newkirk Electric, Revere Sugar, HI-G/AMS Controls, Chevron. **Computer mainframe and peripheral manufacturers,** IBM, OKI, Computer Peripherals Inc., Qume, Floating Point Systems. **Software houses;** Microware, T.S.C., Lucidata, Norpak, Talbot, Stylo Systems, AAA, HHH, Frank Hogg Labs, Epstein Associates, Softwest, Dynasoft, Research Resources U.K., Microworks, Analog Systems, Computerized Business Systems.



GIMIX Systems are chosen by the Pros because of quality, reliability and features.

GIMIX inc.

The Company that delivers Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
 (312) 927-5510 • TWX 910-221-4055

TO ORDER BY MAIL

SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add \$5 handling if order is under \$200.00. Foreign orders add \$10 handling if order is under \$200.00. Foreign orders over \$200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, account #73-32033 Visa or Master Charge also accepted. GIMIX INC. reserves the right to change pricing and product specifications at any time without further notice. are registered trademarks of GIMIX inc. FLEX AND Uniflex are trademarks of Technical Systems Consultants Inc. OS-9 is a trademark of Microware Inc. See their ads for other GIMIX compatible software

FEELING A LITTLE OVERWHELMED



WITH APPLESOFT?

B.E.S.T. Our original Applesoft Optimizer (AOPT) has now been enhanced to bring to the Applesoft programmer the most comprehensive set of software tools available today—B.E.S.T. The BASIC ENHANCED SOFTWARE TOOLS is a machine language program to aid in writing, analyzing, and debugging Applesoft Basic programs. B.E.S.T. will certainly increase programmer efficiency for improved productivity. B.E.S.T. offers: 1) Variable Cross-Reference, 2) Line number Cross-Reference, 3) A Complete Cross-Reference, 4) Optimize variable names with recognition of both short (two character/ Applesoft standard) and long labels, 5) REM Optimizer—with an option to "protect" specified REM statements, 6) Line Optimizer to join short instruction lines of specified length, 7) Merging two programs, 8) Automatic line numbering, and 9) A powerful, yet easy to use Renumber function. Implementation of B.E.S.T. can reduce memory requirements by up to 50% and increase execution speed of Applesoft programs by up to 100%. B.E.S.T. is a low cost alternative to an Applesoft compiler.

B.E.S.T. requires a 48K Apple II/III +, Applesoft in ROM or Language card, and DOS 3.3..... **\$40.00**

EDIT-SOFT. A powerful, yet affordable, line editor for Applesoft. Using EDIT-SOFT, you can drastically cut your programming time. EDIT-SOFT not only contains the standard line editor features like inserting or deleting characters, moving to a specific character, entry of lower case letters, going to the beginning or end of a line, and displaying control characters, but it also has the advanced features that will prove indispensable:

- AUTO LINE NUMBERING
- SPLICING two lines together
- AUTOMATIC CHARACTER COUNTER for quoted strings makes screen formatting a snap
- EDITING LINES as they are being TYPED
- A STATUS LINE keeps you constantly aware of which options are currently being used
- Up to TWENTY MACROS available at any time
- MACRO EDITING AT ANY TIME
- HELP PAGES available when needed

Compare the features of EDIT-SOFT to other line editors, then compare the price. No other line editor has so many features at such a reasonable price!

EDIT-SOFT requires 48K of RAM, Applesoft in ROM (language and RAM expansion cards are fine), and DOS 3.3. ONLY **\$30.00**

PLUS. The original Structured Basic enhancement for Applesoft Basic. PLUS provides the tools that will increase your efficiency, simplify program rewriting, and make your program flow more

understandable. No longer do you have to remember the line numbers of subroutines and procedures; instead, just "DO GRAPHIT" and PLUS does the rest. PLUS adds the following structured programming commands to Applesoft basic: WHEN..ELSE..FIN, UNTIL, WHILE, CASE, SELECT, and OTHERWISE. Multi-line IF..FIN statements are also supported. PLUS includes functions to output indented listings to clarify logic flow and converting source code to standard Applesoft commands.

Requires Applesoft, 32K of RAM, and DOS 3.3... Only **\$25.00.**

BUILD USING. Do you have trouble printing charts, reports, or formatting numbers on the screen? If you do, BUILD USING can solve your problems. BUILD USING is a powerful Applesoft utility which provides a "print-using" type routine for numbers and strings. By creating simple "formats", you tell BUILD USING how to format the output. The output from BUILD USING are strings which may be printed, written to disk, saved for later usage, or even reformatted. With BUILD USING, you can choose how many digits should be displayed to the left and right of the decimal point, and even fill the leading positions with the character of your choice. For example, you can print the number '157.23' and '157.2' or '000157.230', or '....\$157. AND 23/100 DOLLARS'; or hundreds of other ways (including exponential formats). Working with strings is just as easy. Also included are three levels of error trapping, so you can correct numbers that cannot fit into your specified format.

Utilities like BUILD USING are usually difficult to use because they must be located in one memory location (usually between DOS and the DOS file buffers), they cannot be used with your favorite editor or other special routines. BUILD USING does not have this limitation, as it can be easily located in many different memory locations: 1) the "standard" between DOS and DOS file buffers, 2) at HIMEM, 3) APPENDED to your Applesoft program, or 4) anywhere else in memory. Appending BUILD USING to your program is as simple as EXECing a TEXT file. BUILD USING uses the "CALL" command thereby leaving the ampersand vector free for your own use.

BUILD USING requires Applesoft in ROM (Language cards are fine), DOS 3.3 and a minimum of 32K.... Only **\$30.00**



Sensible Software

6619 Perham Drive Dept. MO
West Bloomfield, Michigan 48033
(313) 399-8877

Please specify program desired.

Visa and Mastercard Welcome. Add \$1.25 postage and handling per diskette.

See you at Boston Applefest '82 — Booth 324

• Applesoft is a registered trademark of APPLE Computer Company •

MICRO™

THE 6502/6809 JOURNAL

STAFF

President/Editor-in-Chief
ROBERT M. TRIPP

Publisher
MARY GRACE SMITH

Senior Editor
LAURENCE KEPPLER

Editors
MARY ANN CURTIS
MARJORIE MORSE
TIM OSBORN
LOREN WRIGHT

Production Manager
PAULA M. KRAMER

Typesetting
EMMALYN H. BENTLEY

Advertising Manager
CATHI BLAND

Circulation Manager
CAROL A. STARK

Dealer Orders
LINDA HENS DILL

Comptroller
DONNA M. TRIPP

Bookkeeper
KAY COLLINS

Advertising
Sales Representative
KEVIN B. RUSHALCO
603/547-2970

Subscription/Dealer Inquiries
(617)256-5515

DEPARTMENTS

- 5 Editorial
- 6 Letterbox
- 33 Reviews in Brief
- 55 Microbes and Updates
- 89 Hardware Catalog
- 92 Software Catalog
- 99 6809 Bibliography
- 100 6502 Bibliography
- 103 Annual Index
- 107 Data Sheet
- 111 Advertiser's Index
- 112 Next Month in MICRO

MACHINE-LANGUAGE AIDS

- 9 **Memory Map Relocator**..... *Preston R. Black*
Use this handy program to prevent your programs from being overwritten
- 13 **General BASIC to Machine Language Interface**..... *Christer Engstrom*
Allow better communication between BASIC and machine language
- 19 **Memory Moves with the 6502 and the MC6809**..... *G. Walker and T. Whiteside*
Advantages of 6809's direct page addressing are discussed

APPLE PRINTER UTILITIES

- 37 **LISZT with Strings**..... *L.H. Anderson, D. Cohen, R.F. Searle*
Turn Applesoft listings into easy-to-understand structured formats
- 48 **Apple Graphics for Okidata Microline 80**..... *Charles F. Taylor*
Dump Apple hi-res screen graphics to Okidata printer

PET FEATURE

- 61 **Commodore and MICRO**..... *Loren Wright*
- 63 **PET Memory Protector**..... *Louis F. Sander, Victor H. Pitre*
Isolate 1K or more of memory from BASIC and resets
- 67 **Growing Knowledge Trees**..... *David R. Heise*
Artificial intelligence represented in tree diagrams
- 84 **PET Menu and Tape Timer**..... *Dale De Priest*
Advanced cassette control, extensive use of WAIT command

COLUMNS

- 25 **PET Vet**..... *Loren Wright*
PET SCREEN editing
- 29 **From Here to Atari**..... *James Capparell*
"Front Jacks" on the Atari 400 and 800
- 57 **The Single Life**..... *Brad Rinehart*
HDE Disk BASIC

THE CHIEFTAIN™ 5¼-INCH WINCHESTER HARD DISK COMPUTER



SO ADVANCED IN SO MANY WAYS . . .
AND SO COST-EFFECTIVE . . .
IT OBSOLETES MOST OTHER SYSTEMS
AVAILABLE TODAY AT ANY PRICE.

● HARD DISK SYSTEM CAPACITY

The Chieftain series includes 5¼- and 8-inch Winchesters that range from 4- to 60-megabyte capacity, and higher as technology advances. All hard disk Chieftains include 64-k memory with two serial ports and DOS69D disk operating system.

● LIGHTNING ACCESS TIME

Average access time for 5¼-inch Winchesters is 70-msec, comparable to far more costly hard disk systems. That means data transfer **ten-times faster** than floppy disk systems.

The Chieftain Computer Systems:

Here are the Chieftain 6809-based hard disk computers that are destined to change data processing . . .

- CHIEFTAIN 95W4**
4-megabyte, 5¼-inch Winchester with a 360-k floppy disk drive (pictured).
- CHIEFTAIN 95XW4**
4-megabyte, 5¼-inch Winchester with a 750-k octo-density floppy disk drive.
- CHIEFTAIN 98W15**
15-megabyte, 5¼-inch Winchester with a 1-megabyte 8-inch floppy disk drive.
- CHIEFTAIN 9W15T20**
15-megabyte, 5¼-inch Winchester with a 20-megabyte tape streamer.

● 2-MHZ OPERATION

All Chieftains operate at 2-MHz, regardless of disk storage type or operating system used. Compare this to other hard disk systems, no matter **how** much they cost!

● DMA DATA TRANSFER

DMA data transfer to-and-from tape and disk is provided for optimum speed. A special design technique eliminates the necessity of halting the processor to wait for data which normally transfers at a slower speed, determined by the rotational velocity of the disk.

● RUNS UNDER DOS OR OS-9

No matter which Chieftain you select . . . 5¼- or 8-inch floppy, or 5¼- or 8-inch

Winchester with tape or floppy back-up . . . they **all** run under DOS or OS-9 with **no need** to modify hardware or software.

● UNBOUNDED FLEXIBILITY

You'll probably never use it, but any Chieftain hard disk system can drive up to 20 other Winchesters, and four tape drives, with a single DMA interface board!

● SMOKE SIGNAL'S HERITAGE OF EXCELLENCE

This new-generation computer is accompanied by the same **Endurance-Certified** quality Dealers and end-users all over the world have come to expect from Smoke Signal. And support, software selection and extremely competitive pricing are very much a part of that enviable reputation.

20-Megabyte Tape Streamer Back-Up Option

Available with all Chieftain hard disk configurations. This cartridge tape capability provides full 20-megabyte disk back-up in less than five minutes with just one command, or copy command for individual file transfers. Transfers data tape-to-disk or disk-to-tape. Floppy back-up is also available in a variety of configurations.



Write or call today for details (including the low prices) on the Chieftain Series . . . and on dealership opportunities



SMOKE SIGNAL BROADCASTING®

31336 VIA COLINAS
WESTLAKE VILLAGE, CA 91362
TEL (213) 889-9340

Name _____

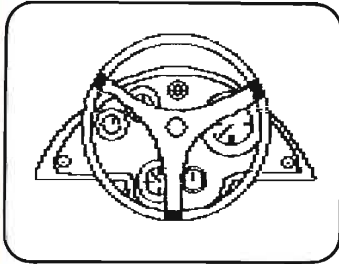
Company _____

Address _____

City _____ State _____ Zip _____

Telephone () _____

About the Cover



This month's cover gets MICRO off to a flying start at the races. Increasing software and hardware sophistication make simulations possible that put you in the driver's seat. The MIT Artificial Intelligence lab has produced a microcomputer-vidodisk combination that lets you simulate a drive through Aspen, Colorado, seeing on TV just what you would see through your windshield if you were actually there.

The cover graphic was generated on an Apple Graphics Tablet, and the output was produced on an IDS Color Printer by Susan Maras at Computerland of Nashua, New Hampshire.

Cover photo: Betsey Bolton
Art Alive! Gallery
200 Merrimack St.
Lowell, MA

MICRO™ is published monthly by:
MICRO INK, Inc., Chelmsford, MA 01824
Second Class postage paid at:
Chelmsford, MA 01824 and additional
mailing offices
USPS Publication Number: 483470
ISSN: 0271-9002

Send subscriptions, change of address, USPS
Form 3579, requests for back issues and all
other fulfillment questions to

MICRO
34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824
or call
617/256-5515
Telex: 955329 TLX SRVC
800-227-1617

Subscription Rates	Per Year
U.S.	\$24.00
	2 yr. / \$42.00
Foreign surface mail	\$27.00
Air mail:	
Europe	\$42.00
Mexico, Central America, Middle East, North Africa, Central Africa	\$48.00
South America, South Africa, Far East, Australasia, New Zealand	\$72.00

Copyright © 1982 by MICRO INK, Inc.
All Rights Reserved

MICRO™

Editorial

Present Shock

Turning on Commodore's new SuperPET is a startling experience. The system's introductory menu modestly offers access to: BASIC; Pascal; FORTRAN; APL; Assembly; and Monitor. Merely hit the appropriate key!

The development of the microcomputer is proceeding at breathtaking speed. A ferociously efficient combination of high technology, intense international competition, and ready venture capital is generating new hardware faster than society can absorb the old. How rapidly the aerospace industry progressed, we once thought, marvelling at the short span between Kitty Hawk and the Moon. But microcomputers are advancing much more swiftly. Incredibly, the arrival of the new SuperPET coincides with ongoing use of the KIM-1, a 1977 single-board system still covered by MICRO. As a special effect, such an eerie foreshortening of time belongs in a *Star Trek* episode, like Commodore's imagemaster, William Shatner.

As we admire the development of ever-more-sophisticated microcomputer hardware, we should remember that each new system requires of its users an enormous investment of time. A case in point: IBM's new Personal Computer. Announced last year to universal acclaim, the system almost entirely lacks software that exploits its 16-bit potential. While the software industry strains to fill the huge gap that appeared the day the Personal Computer was introduced, even more advanced machines are being developed.

We must learn to recognize how conservative we are with our most precious investment, time. Otherwise, costly mistakes will be hard to avoid. For example, those who promoted novel keyboard designs in hopes of replacing the standard QWERTY arrangement have convinced almost no one, the marketplace least of all. We have invested far too much time in learning QWERTY to leave it even for a significantly better keyboard design. The zeal with which microcomputer owners go

on developing systems that are technological antiques should warn us that these systems, like QWERTY, will be with us for a long time to come. And as more and more new systems are introduced, sopping up more and more available time and energy, the gap between a new system's potential and the availability of resources to develop that potential seems likely to widen.

The solution to this problem is certainly not to stop building more advanced computers. The limits of the microelectronics revolution are not yet in sight, and we can look forward to ever more powerful microcomputers. What we must do is understand that the most important component in a working computer system — people — cannot fully process change at such a fast rate. Nor can we afford to discard huge investments of our time. Therefore, we must find ways to keep that investment on board. The SuperPET, for example, includes both the older 6502 and the newer 6809 processor, and can therefore run older as well as newer software. Radio Shack's recently announced Model 16 incorporates the even newer 68000 processor, yet also uses a Z-80. All older Model II software can still be used. In fact, it will even be possible to upgrade Model IIs with the 68000 board.

MICRO supports design decisions that make software compatible with different generations of a system. The same generation of people will be using many generations of computers. To stay in touch with us, the microcomputer revolution must be made compatible with the need imposed by human limits to use our time wisely.

This issue of MICRO spotlights Commodore's PET. Europe's most popular microcomputer, the PET is steadily attracting more American users. The program accompanying David Heise's feature article, "Growing Knowledge Trees," was written especially for the PET. However, the insight it offers MICRO readers into the concept of artificial intelligence makes it must reading for all.

Laurence Kepple

MICRO™

Letterbox

Dear Editor:

I would strongly recommend against your readers' taking, at face value, the comments made in your December 1981 Letterbox, "Atari Ad Attacked."

Mr. Kirby does not define "adapting." If Mr. Kirby takes an Atari program and makes a "similar" program, he may end up in the Austin court he mentions in his letter. The same rules apply to a computer program as to books, etc. It is not necessary to make a 100% duplicate in order to be found guilty of copyright infringement. For example, if he were to use an unusual approach or algorithm in only a part of his new "adapted" program (assuming that the Atari program itself makes it obvious that the contents are copyrighted), the remainder of the "adapted program" could be totally different, yet a copyright infringement could easily be regarded by the court as having occurred. Mr. Kirby will then be subject to a number of possible actions, ranging from criminal penalties, damages, court injunctions... or all of them.

A helpful publication, costing about \$12.00, is "The Copyright Kit," published by the National Attorney's Publications, Inc., P.O. Box 150, East Setauket, N.Y. This book explains copyrighting in layman's terms and clears up the muddy waters created by December's "intellectual property law" expert.

Stephen C. Carpenter
Mondriaanstraat 14
3262 TH Oud-Beijerland
The Netherlands

Dear Editor:

Your magazine is a very good one. My opinion might be illustrated by my collection of your issues. I started reading your magazine in late 1979. I currently have 38 issues, one reprint collection covering six more issues, and am requesting a recently missed issue. When I receive this issue, I will have access to information from 45 of the 46 issues you have placed on the market at this time.

I own an Apple II with 48K of memory, an Applesoft language card, and one DOS 3.3 disk drive. I find your coverage of the Apple to be not only very large in quantity but fine in quality. I also get a lot of ideas from the articles dealing with the other 6502/6809 machines.

Are there any plans to publish articles which describe the other CPU boards which run in the Apple? Even though you are a 6502/6809 journal, an article describing how a 6502 works with a Z80, 6800, 6809, 8088, and other chips would be very interesting.

Also, are any of your readers familiar with the new MTU 6502 machine? I recently received some literature describing it and it isn't too far from a "dream machine" itself. It appears to have hardware 18-bit addressing (yes, 18 not 16) and great bit map graphics. It also has a very sophisticated operating system.

I do have one final problem: A few months back, you had an article which described an operating system for the 6809. I believe it was OS/9 or something similar. But I don't remember seeing a manufacturer's name or address (or price for that matter). Did I simply overlook these or were they missing? Could you re-supply them? Does this operating system come in a format for the Apple II's various 6809 boards?

Larry W. Virden
1207 Rosehill Rd., Apt. 104
Reynoldsburg, OH 43068

Editor's note: The MICRO staff is very interested in hearing from readers who have experience using any of these CPU boards. Since these boards use the 6502 to handle the I/O and other functions, it would be valuable to see how the two CPU's cooperate with each other. Possible areas of coverage could include how the dual CPU's deal with cycle stealing, address translations, interrupts, parameter passing, etc.

The OS-9 operating system is available for the MILL 6809 card through Stellation Two, P.O. Box 2342, Santa Barbara, CA 93120; (805) 966-1140.

Dear Editor:

I just finished reading the March issue of MICRO. As an OSI user (I have a C2-4P MF system) I wish to thank you for your editorial "Hello, OSI?" and also for making the March issue an OSI Feature.

Let's hope that the cover photo is not a group of OSI users watching the OSI personal computer division going up in flames.

After reading the notation about the cover photo, I looked through my collection of computer manuals and found a copy of the manual prepared by Professors J.G. Kemeny and T.E. Kurtz dated June 1965. A statement of interest in the manual is: "The language that you will use is BASIC (Beginner's All-purpose Symbolic Instruction Code) which is at the same time precise, simple, and easy to understand."

J. Edward Loeffler, Jr.
Elkins Lake, Box 278
Huntsville, TX 77340

Dear Apple Owners:

In conjunction with the release of The Graphics Magician and the updated Complete Graphics System II, Penguin Software is announcing a new policy with our applications software for the Apple. The Complete Graphics System II, Special Effects, and The Graphics Magician will all now be available on non-protected disks.

We've been torn between two points of view. As computer users, we appreciate the ability to have several working copies of our applications software, and even the ability to go in and modify the code, if desired. We'd use programs such as VisiCalc or DB Master for dozens of other applications if we could have them running off several separate disks and didn't have to guard our master copies with such extreme care. Disks are fragile; we handle thousands of them, and no disk is absolutely 100% error-proof. Being programmers also, occasionally we'd like to adapt a program slightly to our system or our needs. On locked disks, much of a software product's potential usage goes untapped. (Continued)

Letter Box *(Continued)*

But as publishers we've been drawn into the prevailing point of view that lack of copy protection means greatly decreased sales due to casual "piracy." This is not just a crazed overreaction; we've all been to user-group meetings, homes of acquaintances, and even some computer stores, where we've been aghast at the almost encouraging attitude toward copying copyrighted software, most of which took authors months, maybe years, to perfect. The real scare here is that many of us have decided to take a risk on a very new industry and trust our livelihoods to it. Suddenly, individuals out there become statistics, some of which say that for every non-protected program sold, there are at least a dozen "pirated" copies. Those kinds of numbers could really wreak havoc on paying the bills. Scary? Yes.

From these conflicting points of view, our desire to make a good product

better won, but not by much over our fear of tampering with something that is already going well. Our policies, from pricing to support, have always been very consumer-oriented. Ultimately, it is from that viewpoint that we decided to go ahead with removing the protection. We feel that you, the consumer, are entitled to software as useful as possible for the money you spend. Our hope is that the added convenience will result in more sales, not fewer, and that the software market has matured to the point where people realize that the result of illegal copying is less convenience for everyone with all software. We hope that people will think twice before accepting copies from friends, and we hope to be able to continue this policy and start a new trend toward improved usability of all applications software. Please don't abuse our trust in you.

Mark Pelczarski, President
Penguin Software
1206 Kings Circle
West Chicago, IL 60185

Dear Editor:

It would be extremely helpful if some of your readers could direct me to sources for two items: 1) a program in BASIC or machine language for OSI, Apple II, TRS-80, or PET, to score Gymnastics Meets; 2) a 16K dynamic RAM (4116) board to add to OSI Super-board II.

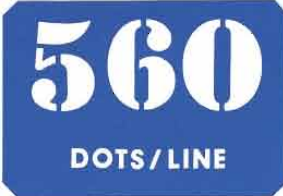
I have been looking for both of these for some time and have had no luck.

Bro. Felix Neussendorfer
Monasterio San Antonio Abad
Box 729
Humacao, PR 00661

If you have comments you'd like to share with MICRO's readers, why not send a letter to the editor?

Editor
MICRO
Box 6502
Chelmsford, MA 01824

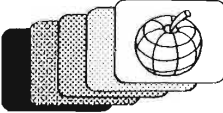
MICRO




Double Density HGR

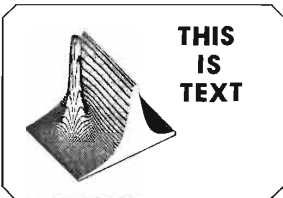
COSMOS

SCREEN MIXER





Highlight




Hi-Res Graphics+Text


SCREEN MIXER is a set of three modules for APPLE-II. Just plug-in these modules to your Apple-II, and you will have the Apple-II with more features you could not expect till now----

SCREEN MIXER provides:

- The mixed screen of any two of screens available for the Apple-II. Please note that all of HGR, LGR and Text screen has two pages. The mixing is done with hardware, not like Hi-Res Text Generator Programs, thus you need no software and the scroll speed is not reduced. Also, you can scroll the text without any effect to the graphic patterns.
- The Double Density High Resolution Graphics. Yes, you can plot 580 dots in one line. You have only 280 dots in one line on ordinary Apple-II. (Software is required)
- One of the most advanced character display. Besides Normal and Inverse characters which are already built-in you will have the choice of Half-Intensity and Highlighted characters. And more, you may Over-Write or Over-Type any character to other character if you want to do so! (Software is included)



Half Intensity



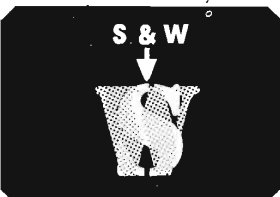
Lo-Res Graphics+Text

\$60 MANUAL — \$10.00, applied towards purchase.

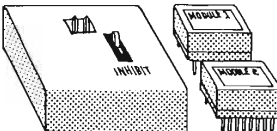
COSMOS™
WORLD WIDE COMPUTER SUPER SHOP

Dealers inquiries invited.
For more information call or write to ASTAR INTERNATIONAL CO.
5676 FRANCIS AVE., CHINO, CA 91710 Phone 714-627-9887

Apple-II is a registered trademark of Apple Computer Inc.



Over Write



A New Age Has Begun



SDS Presents "The Professional" Series

For the first time ever, three packages encompassing all major Apple operating systems:

- ASCII Express "The Professional"™ for Apple DOS
- P-Term "The Professional"™ for Apple Pascal
- Z-Term "The Professional"™ for Apple CPM

FORGET EVERYTHING YOU'VE EVER HEARD ABOUT COMMUNICATIONS SOFTWARE.

SDS now offers three new communication packages like no others on the market today. These packages are designed to operate alike, to look alike, and to be completely conversive with each other. An operator familiar with any one of these packages can easily operate the other two with minimal instruction. Absolutely no other packages can equal these outstanding features:

- All three packages are totally conversive with each other. ANY type of file may be transferred completely error-free within these or compatible systems utilizing the Christensen Protocol.
- Transfers may also include any ASCII based dial-up host computer. These flexible transfer capabilities are made possible by a wide variety of send options unique to SDS software.
- Support of ALL Apple-compatible modems and ALL display systems including any 80-column board or external terminal; 40-column too.
- Terminal emulation — allows your 80-column board or external terminal to "emulate" screen control codes

of other terminals. Ideal for screen-oriented editors or data bases.

- Fully unattended answer mode features password only access and error-free transfer of files, as well as standard ASCII transfer modes.
- All packages support up to 1200 baud. Z-Term and ASCII Express support up to 4800 baud when interrupt-driven.
- Greatly enhanced macro system now includes programmable "hand shaking" macros — entire log-on sequences may be automatically executed. Also, many internal program commands may be initiated from macro control. Multiple macro libraries may be stored for use with a variety of hosts.

A NEW STANDARD OF EXCELLENCE

"The Professional" series is so innovative and superior to other packages in the field that it is truly in a class by itself. Our comparison chart gives a detailed point-by-point analysis of these and other communication packages. Write for it. You'll be convinced — when it comes to communication software, SDS stands alone.

SDS
southwestern
data systems™

P.O. Box 582-M
Santee, CA 92071
714-562-3670

CP/M is a trademark of Digital Research.
Apple is a trademark of Apple Computers, Inc.

Memory Map Relocator

by Preston R. Black

This short program relocates a previous memory map program. Thus, even your longest program won't be written over by the memory map program.

MEMORY MAP Relocator
requires:
Apple II

The MEMORY MAP program by Peter Cook (MICRO 36:45) is a very elegant way to assess memory usage by the Apple II computer. I have found this program to be particularly useful as a tool in program development. It gives me an up-to-date account of the size of my program and the space reserved for variable storage.

One of the drawbacks of the program is its location in memory (\$3200). While this does not interfere with most of my programs, it is annoying to have to frequently reload a large program because parts of it have been written over by MEMORY MAP. This can be disastrous if the program you are developing has not been saved. Moreover, MEMORY MAP keeps reminding me that there is free space to use with my programs. Wouldn't it be great if MEMORY MAP were always loaded into that free space, leaving your program intact?

As Mr. Cook points out in his article, MEMORY MAP is not readily relocatable since there are frequent absolute references within the program. To relocate the program by hand would be most tedious. Obviously, a relocating loader for MEMORY MAP is what we need.

MEMORY MAP RELOCATOR is such a relocating loader. This program defines the free space in RAM and loads MEMORY MAP into that area. MEMORY MAP RELOCATOR then updates the relocated program to make it internally consistent and jumps to the beginning of the relocated program to run MEMORY MAP.

Memory Map Relocator

```

: *****
: *
: *
: *      MEMORY MAP RELOCATOR
: *      BY
: *      PRESTON R BLACK MD
: *      MAY 1981
: *
: * THIS PROGRAM WILL RELOCATE
: * THE MEMORY MAP PROGRAM
: * WRITTEN BY PETER A. COOK
: * (MICRO 36:45) TO RESIDE
: * IN RAM WHERE THERE IS FREE
: * SPACE AND WILL NOT WRITE
: * OVER THE BASIC PROGRAM
: * ALREADY IN MEMORY
: *
: *****
:
: EQUATES
:
: NADDRL EPZ #06
: NADDRH EPZ #07
: NENDL  EPZ #08
: NENDH  EPZ #09
: BUFFL  EPZ #19
: DIFF   EPZ #1B
: BUFFH  EPZ #1C
: LENGTH EPZ #2F
: FROMPT EPZ #33
: PGENDL EPZ #AF
: PGENDH EPZ #B0
: OLDADL EPZ #32
: OLDADH EPZ #39
: PGPTRL EPZ #CA
: PGPTRH EPZ #CB
: RSTPG3 EQU #9E25
: RSTRTS EQU #9E30
: INSD92 EQU #FB8E
: COUT   EQU #FDED
:
:      ORG #300
:
: 0300 4C1303  START JMP INIT
: 0303
: ; THIS ROUTINE RESTORES PAGE THREE DOS POINTERS
: 0303 ; WHICH ARE WRITTEN OVER BY THIS ROUTINE
: 0303
: 0303 A960   RSTORE LDA #60 ; PLACE 'RTS' WHERE
: 0305 8D309E STA RSTRTS ; WE WANT IT BEFORE
: 0308 20259E JSR RSTPG3 ; WE JUMP TO RESTORE ROUTINE
: 030B
: 030B A9AD   LDA #AD ; RESTORE CHANGED DOS BYTE AND
: 030D 8D309E STA RSTRTS ; JUMP TO RELOCATED MEMORY MAP
: 0310 8C1900 JMP (BUFFL)
: 0313
: ; THE TRUE BEGINNING OF THE PROGRAM
: 0313
: 0313 A533   INIT LDA #33 ; WHICH LANGUAGE?
: 0315 C9BE   CMP #">" ; INTEGER BASIC?
: 0317 F00C   BEQ INIT1
: 0319 E680   INC PGENDH ; NO, THEN ADD ONE TO H.O. BYTE
: 031B A5B0   LDA PGENDH ; APPLESOFT EOP POINTER
: 031D 8507   STA NADDRH ; TO MAKE SURE WE ARE OVER
: 031F 851A   STA BUFFL+1 ; THE PROGRAM
: 0321 C6B0   DEC PGENDH
: 0323 D009   BNE INIT2
: 0325 38     INIT1 SEC ; YES, USE INTEGER BASIC
: 0326 A5CB   LDA PGPTRH ; POINTERS
: 0328 E90C   SBC #6C ; ALLOW ENOUGH SPACE BELOW
: 032A 8507   STA NADDRH ; PROGRAM FOR MEMORY MAP

```

(Continued)

How it Works

The first step in relocating MEMORY MAP is to define the area of free space in RAM. Both Applesoft and Integer BASIC have pointers to the end of the program stored in memory. Unfortunately, they are not the same bytes. In addition, programs are not stored the same way in the two languages. Applesoft begins storing programs at \$801 and succeeding bytes are added above this. Integer BASIC begins storing programs at HIMEM and places all succeeding bytes below this. Thus, the pointers to the end of the program in the two languages must be treated differently.

For Applesoft we must load MEMORY MAP above the program already in memory. If we take the high order byte of the address of the end of the program and add one to it, we can be certain that we are above the program in memory. MEMORY MAP requires slightly less than \$C00 bytes of memory if we include the area used by the printing routine. Therefore, in Integer BASIC we must go at least this far below the program to load MEMORY MAP. Otherwise we will overwrite the BASIC program already in memory.

During initialization of MEMORY MAP RELOCATOR we can determine the current language by checking which prompt is used. Appropriate adjustments must be made to the high order byte of the program end. We now have a starting address within the free space to place MEMORY MAP. Calculate the ending address of the relocated program by adding the length of MEMORY MAP to the new starting address.

Next load MEMORY MAP into the free area. This is done by constructing a string consisting of "BLOAD MEMORY MAP,A\$xx00". The xx is the high order byte for the new starting address that we determined during initialization. But before we can place this number into our string, it must be converted into the ASCII representation of that number. This is done by first dividing the number into two nibbles (a nibble is equal to four bits) and converting the nibbles into the Apple ASCII code for the respective numbers. The Apple ASCII codes for the numbers from 0 to 9 are \$B0 to \$B9 respectively. Thus, to convert these numbers, we simply add \$B0 to them. (The numbers \$A to \$F must have \$B7 added to them to convert them into ASCII.) Once the numbers have been converted to ASCII, they are added to our string to complete it. We then use COUT (\$FDEE) to pass the string to DOS to be executed.

Memory Map Relocator (Continued)

```

032C 851A      STA BUFFL+1
032E 18       INIT2  CLC
032F 6906     ADC  #$06      ; FIND END OF
0331 8509     STA NENDH     ; RELOCATED PROGRAM
0333 851C     STA BUFFH     ; AND SAVE IN BUFFERS
0335 A9E0     LDA  #$E0
0337 8508     STA NENDL
0339 A900     LDA  #$00
033B 8506     STA NADDRL
033D 8519     STA BUFFL
033F A507     BLOAD  LDA NADDRH     ; TAKE H.O. BYTE OF NEW START
0341 290F     AND  #$0F     ; LOOK AT L.O. NIBBLE
0343 C90A     CMP  #$0A     ; IS IT <10
0345 9006     BLT  BLOAD1
0347 18       CLC          ; NO, CONVERT TO ASCII
034B 69B7     ADC  #$B7     ; FOR 'A'-'F'
034A 4C5003   BLOAD1 JMP  BLOAD2
034D 18       CLC          ; YES, CONVERT TO ASCII
034E 69B0     ADC  #$B0     ; FOR '0'-'9'
0350 A201     BLOAD2 LDX  #$01
0352 9DF803   STA  LOAD1,X  ; STORE IN STRING
0355 A507     LDA  NADDRH
0357 4A       LSR          ; NOW LOOK AT H.O. NIBBLE
035B 4A       LSR          ; AND CONVERT TO ASCII
0359 4A       LSR          ; AND STORE IN STRING
035A 4A       LSR
035B C90A     CMP  #$0A
035D 9006     BLT  BLOAD3
035F 18       CLC
0360 69B7     ADC  #$B7
0362 4C6803   BLOAD3 JMP  BLOAD4
0365 18       CLC
0366 69B0     ADC  #$B0
0368 CA       BLOAD4 DEX
0369 9DF803   STA  LOAD1,X
036C BDE203   BLOAD5 LDA  LOAD,X  ; USE COUT TO
036F F006     BEQ  UPDATE  ; PASS BLOAD COMMAND TO
0371 20EDFD   JSR  COUT     ; DOS WITH RELOCATING
0374 E8       INX          ; STARTING ADDRESS
0375 D0F5     BNE  BLOAD5
0377 38       UPDATE SEC      ; FIRST FIND OFFSET
037B A507     LDA  NADDRH  ; BETWEEN ORIGINAL
037A E932     SBC  #OLDADL  ; AND RELOCATED PROGRAM
037C 851B     STA  DIFF     ; AND SAVE
037E A000     UPDAT1 LDY  #$00
0380 B106     LDA  (NADDRL),Y
0382 208EFB   JSR  INDS2   ; FIND LENGTH OF
0385 A52F     LDA  LENGTH  ; OP CODE
0387 AB       TAY
038B C902     CMP  #$02     ; IF < 3 THEN NEXT ONE
038A D01D     BNE  UPDAT3
038C B106     LDA  (NADDRL),Y
038E C932     CMP  #OLDADL ; IF = 3 THEN
0390 9017     BLT  UPDAT3  ; SEE IF ADDRESS IS
0392 C939     CMP  #OLDADH ; WITHIN PROGRAM
0394 B013     BGE  UPDAT3  ; BOUNDARIES
0396 C93B     CMP  #OLDADH-1 ; IF NOT, THEN
0398 D008     BNE  UPDAT2 ; GOTO NEXT OP CODE
039A 88       DEY
039B B106     LDA  (NADDRL),Y
039D CB       INY
039E C9E0     CMP  #$E0
03A0 8007     BGE  UPDAT3
03A2 18       UPDAT2 CLC      ; IF ADDRESS WITHIN PROGRAM
03A3 B106     LDA  (NADDRL),Y ; ADD OFFSET TO CHANGE
03A5 651B     ADC  DIFF     ; TO RELOCATED ADDRESS
03A7 9106     STA  (NADDRL),Y
03A9 CB       UPDAT3 INY
03AA 98       TYA
03AB 18       CLC
03AC 6506     ADC  NADDRL  ; UPDATE TO NEXT
03AE 8506     STA  NADDRL  ; OP CODE
03B0 9002     BCC  UPDAT4
03B2 E607     INC  NADDRH
03B4 38       UPDAT4 SEC      ; END OF THE
03B5 A50B     LDA  NENDL  ; RELOCATED PROGRAM?
03B7 E506     SBC  NADDRL
03B9 A509     LDA  NENDH
03BB E507     SBC  NADDRH
03BD 9003     BCC  UPDAT5
03BF 4C7E03   JMP  UPDAT1
03C2 A51C     UPDAT5 LDA  BUFFH ; NO, CONTINUE UPDATE
03C4 A06F     LDY  #$6F    ; YES, REPLACE $6F
03C6 9119     STA  (BUFFL),Y ; WITH PROPER ADDRESS
03C8 ;
03C8 ; NOW CORRECT THE ADDRESSES OF THE STARTING PAGES
03C8 ; FOR PRINTING THE MOVED TEXT PAGE
03C8 ;
03C8 ;
03C8 A51C     LDA  BUFFH
03DA 8509     STA  NENDH
03CC A900     LDA  #$00
03CE 8508     STA  NENDL

```

Memory Map Relocator (Continued)

```

03D0 A0B3      LDY #B3
03D2 1B        UPDAT6 CLC
03D3 B108      LDA (NENDL),Y
03D5 651E      ADC DIFF
03D7 9108      STA (NENDL),Y
03D9 C8        INY
03DA C8        INY
03DB C0E0      CPY #E0
03DD 90F3      BLT UPDAT6
03DF          :
03DF          : WHEN FINISHED RESTORE PAGE THREE POINTERS
03DF          : BEFORE RUNNING RELOCATED MEMORY MAP
03DF          :
03DF 4C0303     JMP RSTORE
03E2 8DBD84    LOAD  HEX 8DBD84
03E5 C2CCCF    ASC  "BLOAD MEMORY MAP,A#"
03E8 C1C4A0
03EB CDC5CD
03EE CFD2D9
03F1 A0CDC1
03F4 D0ACC1
03F7 A4
03FB 0000B0    LOAD1 HEX 0000B0B0B000
03FB B0B000
03FE          LENTH EQU *-START
    
```

Once the program has been loaded into memory, we must update it to make internal calls consistent. The algorithm for this is as follows: First, the offset between the original program and the relocated program is calculated. This is the amount that must be added to the original addresses to make them compatible with the relocated program. Using the monitor routine INSDS2, we determine how many bytes are used by each op code. If the op code requires only one or two bytes, then any addressing will be relative and will not require updating. If, however,

the op code is three bytes long, then all addresses used must be absolute.

We must also check to see if that address is within the boundaries of the original program (i.e., from \$3200 to \$38E0). If it is, then we add the offset to the high order byte. If it is not, we go to the next op code. We continue in this fashion until we reach the end of the relocated program. When the relocated program has been completely updated, an indirect jump to the beginning of the relocated program will run MEMORY MAP.

How to Use the Program

MEMORY MAP RELOCATOR resides on page three of memory. Since it is longer than \$D0 bytes long, it overwrites important DOS vectors located on page three. To insure proper function of DOS after the program is run, a short routine to restore these pointers begins the program. It is placed at the beginning so it will not be destroyed while the pointers are restored.

The routine to restore the pointers makes use of the part of DOS which places the pointers onto page three during the bootstrap. I place an 'RTS' (\$60) in the place that suits my purposes and restore the byte to what it was before performing the indirect jump to run MEMORY MAP.

Once the program has been entered and saved, BRUNning it will place MEMORY MAP into the available free space and run it. Remember that this program is written to run with a program named MEMORY MAP which is normally stored from \$3200 to \$32E0. With minor modifications, this program can be converted to run with a program beginning at any address, and of any length.

Please contact Mr. Black at 16 Durham St., Boston, MA 02115.

MICRO

"master diagnostics +plus"

Apple Computer Co. has provided you with the best equipment available to date. **master diagnostics +plus** will examine every functional operation of your computer, detect errors, and report any malfunctions. Besides the diagnostic's part of the analysis, **master diagnostics +plus** provides routines for complete internal and external maintenance of your computer, it's peripherals and monitor.

CALIBRATING THE DISK DRIVE SPEED is made simple and accurate. HEAD CLEANING is made even easier with our special head cleaning kit and programmed routine.

The forty page manual provides in depth help, trouble shooting guides, and detailed information on everything needed to diagnose and maintain the entire Apple II or Apple II+ computers and peripherals. *It not only tells you what's wrong, but what to do about it.* Our monitor routines are actual programmed tests, not just "test patterns". Absolute video clarity can be obtained with a little time and knowhow!

Our product sets the standard in Apple diagnostics and maintenance. We are proud to say that it is being used by hundreds of colleges and technical schools world wide. Fortune five-hundred companies and even Apple dealers have been using our product for over one year and we have received nothing but praise for the thoroughness, ease of use, and technical guidance from our product.

Price:
master diagnostics \$49.95
master diagnostics +plus \$69.95

REQUIRES: 48K, FP IN ROM
 1 DISK DRIVE, DOS 3.3

Our unlimited warranty insures the diagnostic's diskette against any reasonable mis-hap.

© **NIKROM**

Technical Products, Inc.

25 PROSPECT STREET • LEOMINSTER, MA 01453

APPLE is Registered Trademark of Apple Computer Co. MICROMODEM II is Registered Trademark of D.C. HAYES

THE TESTS INCLUDE:

MOTHERBOARD ROM TEST	MONITOR TEST PATTERN
APPLESOFT CARD ROM TEST	MONITOR & TV YOKE ALIGNMENT
INTEGER CARD ROM TEST	LO-RES COLOR TESTS
MOTHERBOARD RAM TESTS	HI-RES COLOR TESTS
PARALLEL CARD PROM TESTS	RANDOM HI-RES COLOR GENERATOR
DISK DRIVE SPEED CALIBRATION	LISSAJOUS PATTERNS
DISK HEAD READ/WRITE TEST	SPEAKER FUNCTION TESTS
DISK WRITE PROTECT SWITCH TEST	SQUARE WAVE MODULATION
DISK DRIVE HEAD CLEANING ROUTINES	PADDLE & SPEAKER TESTS
DISK DRIVE MAINTENANCE	PADDLE & BUTTON TEST
DC HAYES MICROMODEM II TEST	PADDLE DRIFT TEST
(EPROM & HARDWARD)	INTERNAL MAINTENANCE
MONITOR SKEWING TESTS	GENERAL MAINTENANCE
MONITOR & MODULATOR CALIBRATION	ON BOARD HELP
MONITOR TEXT PAGE TEST	40 PAGE PROCEDURE MANUAL

the +plus

master diagnostics +plus provides everything needed to maintain your computer. The entire package is housed in our own patented, molded diskette case to protect against static electricity, x-ray and other contaminants.

Included in the kit is:

- THE MASTER DIAGNOSTICS DISKETTE
- PROCEDURE MANUAL
- HEAD CLEANING KIT WITH TWO HEAD CLEANING DISKETTES
- BOTTLE OF CRT SCREEN CLEANER
- BOTTLE OF COMPUTER HOUSING CLEANER
- CHAMOIS TIPPED WANDS WHICH ARE SPECIALLY DESIGNED FOR LINT-FREE & STATIC-FREE CLEANING ANY PART OF THE MACHINE, INTERNAL & EXTERNAL

ORDER TOLL-FREE ANYTIME • Dealer Inquiries Invited

C.O.D., Master Charge & Visa users call: 1-800-835-2246

Kansas Residents call: 1-800-382-2421

Add: \$2.50 for shipping • USA. FOREIGN: \$5.00

Note: "BRAIN SURGEON" owners may update to master diagnostics +plus. Return your diskette with \$24.95 plus \$2.50 shipping.



BOX 120
ALLAMUCHY, N.J. 07820
201-362-6574

HUDSON DIGITAL ELECTRONICS INC.

THE TASK* MASTERS

HDE supports the *TIM, AIM, SYM and KIM (TASK) with a growing line of computer programs and peripheral components. All HDE component boards are state-of-the-art 4½" x 6½", with on board regulation of all required voltages, fully compatible with the KIM-4 bus.

OMNIDISK 65/8 and 65/5

Single and dual drive 8" and 5¼" disk systems. Complete, ready to plug in, bootstrap and run. Include HDE's proprietary operating system, FODS (File Oriented Disk System).

DM816-M8A

An 8K static RAM board tested for a minimum of 100 hours and warranted for a full 6 months.

DM816-UB1

A prototyping card with on-board 5V regulator and address selection. You add the application.

DM816-P8

A 4/8K EPROM card for 2708 or 2716 circuits. On board regulation of all required voltages. Supplied without EPROMS.

DM816-CC15

A 15 position motherboard mounted in a 19" RETMA standard card cage, with power supply. KIM, AIM and SYM versions.

DISK PROGRAM LIBRARY

Offers exchange of user contributed routines and programs for HDE Disk Systems. Contact Progressive Computer Software, Inc. for details.

HDE DISK BASIC

A full range disk BASIC for KIM based systems. Includes PRINT USING, IF . . . THEN . . . ELSE. Sequential and random file access and much more. \$175.00

HDE ADVANCED INTERACTIVE DISASSEMBLER (AID)

Two pass disassembler assigns labels and constructs source files for any object program. Saves multiple files to disk. TIM, AIM, SYM, KIM versions. \$95.00

HDE ASSEMBLER

Advanced, two pass assembler with standard mnemonics. KIM, TIM, SYM and KIM cassette versions. \$75.00 (\$80.00 cassette)

HDE TEXT OUTPUT PROCESSING SYSTEM (TOPS)

A comprehensive text processor with over 30 commands to format and output letters, documents, manuscripts. KIM, TIM and KIM cassette versions. \$135.00 (\$142.50 cassette)

HDE DYNAMIC DEBUGGING TOOL (DDT)

Built in assembler/disassembler with program controlled single step and dynamic breakpoint entry/deletion. TIM, AIM, SYM, KIM AND KIM cassette versions. \$65.00 (\$68.50 cassette)

HDE COMPREHENSIVE MEMORY TEST (CMT)

Eight separate diagnostic routines for both static and dynamic memory. TIM, AIM, SYM, KIM and KIM cassette versions. \$65.00 (\$68.50 cassette)

AVAILABLE DIRECT OR FROM THESE FINE DEALERS:

Progressive Computer Software
405 Corbin Road
York, PA 17403
(717) 845-4954

Johnson computers
Box 523
Medina, Ohio 44256
(216) 725-4560

Falk-Baker Associates
382 Franklin Avenue
Nutley, NJ 07110
(201) 661-2430

Perry Peripherals
P.O. Box 324
Miller Place, NY 11764
(516) 744-6462

Lux Associates
20 Sunland Drive
Chico, CA 95926
(916) 343-5033

Laboratory Microcomputer Consultants
P.O. Box 84
East Amherst, NY 14051
(716) 689-7344

BASIC to Machine Language Interface

by Christer Engstrom

Since the AIM lacks a SYS or CALL function, it is difficult to communicate between BASIC and machine language. This interface routine makes the USR(W) function more flexible and allows entry of the machine language address directly in hexadecimal through the BASIC variable AD%.

Interface

requires:

AIM with 4K
BASIC
AIM Assembler

I own an AIM with 4K RAM and BASIC and assembler ROMs. The BASIC interpreter is slow, but the machine itself is very fast. Furthermore, there is a lack of cooperation between BASIC and machine language. The only way to jump out from BASIC to execute other code in memory, is to use the USR(W) command. This command lets you jump to a subroutine whose address is given in locations \$04 and \$05, and also pass a floating point value in locations \$B1 - \$B6.

A frequent use of this method results in many POKEs, which are done byte-by-byte in decimal. This is not good for readability. I think we need an easier way to jump out, a better way to define where to jump, and a method to pass parameters. We need a general interface.

The best way to define an address is with (ASCII) hex characters. Here's my solution to the problem: every time BASIC executes the USR(W) command, the machine enters the general interface. The integer AD% is supposed to contain the address to jump to, in high-low order. The interface scans the variable table and searches for AD% (which in the machine is interpreted as \$C1C4 - ASCII of 'A' and 'D', each ORed with \$80). If it is found and contains an address > \$00FF, the interface converts the value W specified in the

USR(W) statement, to a signed binary value in locations \$AC to \$AD (subroutine \$BEFE).

Next it loads the byte at location \$AD (LSB of the value) into the accumulator, and jumps to the subroutine. On returning from the subroutine, the accumulator is stored into \$AD, and the signed binary value in \$AC-\$AD is converted back to the floating-point register (subroutine \$COD). Finally, a return to BASIC is made.

Examples

1. You want to jump to a subroutine at location \$0400.

```
10 AD% = X"0400"  
20 L = USR(0)
```

2. Take advantage of the monitor routine at location \$E97A. Don't forget that the accumulator must be loaded with a value:

```
10 AD% = X"E97A"  
20 L = USR(A)
```

I know, you're thinking that the X"0400" and the X"E97A" are not conforming to general BASIC syntax. But AD% = 1024 and AD% = -576 are! So what we now need is a way to translate all X".." expressions to their decimal equivalents before execution. That is done by the hex converter. If the general interface doesn't find AD% or if AD% is zero, all X".." and Z".." expressions are converted to decimal.

You can see that the hex converter is entered *via* the general interface. This means that it is easy to modify the interface so that it can execute more functions (with a function code in AD%). You may even want to modify the whole interface. Maybe it is better when used this way:

```
10 AD$ = "E97A"  
20 L = USR(A)
```

Let's get back to the hex converter. If you want to assign an unsigned value > \$7FFF to a BASIC Integer, you must consider this: the interval \$8000 to \$FFFF equals the decimal interval -32768 to -1. This means that \$8000 = -[\$10000 - \$8000] = -(65536 - 32768) = -32768.

We don't always want to translate the hex string to a signed value, so another type must be defined. This leads us to two different syntaxes. To get a signed decimal value, precede the hex string in quotes with an X. For positive (unsigned) values, use a Z instead.

Example 3:

```
X"9000" = -(65536 - 36864)  
          = -28672 BUT  
Z"9000" = 36864
```

If you define an address, use the X type. Note that only the program part, not the variable part, is hex-converted. Also note that the string within the quotes must consist only of the hex characters (0 - 9, A - F), and have a length of 0 - 4 characters. Left-fill with zeroes is done automatically. For example:

```
10 L = USR(X"A")  
20 A = X"CC"/B
```

After hex conversion,

```
10 L = USR(0010)  
20 A = 00204/B
```

no compression is done. The string beginning with X or Z is replaced by a decimal value of the same length. If the hex string is not enclosed within quotation marks, BASIC will attempt to interpret some strings to function codes during input phase; "DEF" for example.

Program Description: The Interface

In my version, the interface consists of two parts: the interface and the hex converter. Since only relative branches are made, both parts are relocatable. The interface and the hex conversion need

not cooperate — simply remove one of them. In my version, the interface must know the real start addresses of functions it should handle. The BASIC input buffer (\$14 - \$50) is used as a work area. The interface starts with a lookup of the variable table.

Here are some valuable points:

1. The start address for the table is in locations \$75 and \$76.
2. An integer variable name has \$80 added to the first and second character of its name (thus making AD% = \$C1C4, not \$4144).
3. Every entry in the table consists of seven bytes, the first two for its name.
4. An integer variable has its value [signed] in the two bytes following the name.
5. "End-of-table" is flagged by \$AA in the first location of an entry.

Here is a description of what is done at each label:

ENTRY—Stores the address of the variable table start in a work area.

SCAN—A search for \$C1C4 or end-of-table is done.

NEW—Get address for next entry, scan again.

CHECK—Tests value in leftmost byte of integer AD%.

JUMP—Jumps to subroutine after floating point conversion. At return, converts integer back to floating point array.

OUT—Clears A,X,Y registers and returns back to BASIC.

FUNC—Jumps to functions by testing the rightmost byte of AD%. Invalid functions are ignored.

FUNC.—The functions.

Hex Converter

This is a fairly long and space-consuming part; the readability is worth more than smart programming.

1. A scan for all strings beginning with X" and Z" is done in the program part (page "CHECK PROG").
2. If such a string is found, and the rest of it conforms to the above given

syntax, it is converted to the decimal equivalent (pages "MOVE RIGHT JUST" and "STRING TO HEX").

3. If it is an X" string with a value > = 32768, a sign-conversion [see example 3] is done (page "SIGN CONVERSION").
4. Finally, the converted value is edited back to the program (pages "EDIT NOW" and "MOVE TO PROG").
5. When the program part is scanned through, a return to the interface is done (page "MAIN LOOP").

Conclusion

We now have an extended and more flexible way to use BASIC with the rest of the machine. Even some monitor routines can be used without specially written routines. The hex converter allows us to specify constants in hexadecimal mode. This method also cooperates better with the rest of the machine. Finally, the interface can help during the editing of a program (function codes could be used).

Please address the author at Gotlandsgatan, 71A, 2½ TR, S-11638, Stockholm, Sweden.

Olympic Sales Company

SERVING YOU SINCE 1947

Main Showroom & Offices:
216 South Oxford Avenue
Los Angeles, CA 90004

WE HONOR
VISA and MASTERCARD

TELEX: 67 34 77

ORDER DESKS open 7 Days a Week!
7:00 AM to 7:00 PM Mon thru Sat
Sunday Noon to 5:00 PM

Order Desks: (213) 739-1130

TOLL-FREE (outside Calif.) TOLL-FREE (within Calif.)

800-421-8045 800-252-2153

NO ONE UNDERSELLS OLYMPIC SALES

Write & request our new 112 pg catalog—please include \$1.00 to defray postage & handling—includes many more items—from TV's to Watches!

All goods subject to availability; this ad supersedes all previous ads; we are not responsible for typographical errors; we will meet or beat any advertised prices if the competition has the goods on hand.
Minimum shipping and handling \$4.95.
All orders subject to verification and acceptance.

We are an authorized servicing Apple dealer for Apple II & III. Immediate delivery on all models—we carry an enormous inventory of Apple products at all times!

Immediate delivery
16K—32K—48K—64K—Plus or Integer in stock!
NEW, IMPROVED APPLE III 128K VERSION

Large Inventory of:
Disk with controller DOS 3.3
Second Disk Drives
Personal Fortran—Label languages
Dow Jones & Quote reporter
Graphics Tablets
Visicalc for Apple II & III
Smarterm 80 column card
Micromodem II by DC Hayes and more.

ATTENTION: Immediate delivery
WE ALSO CARRY SOFTWARE!
Personal Software
Peach Tree Software
Microsoft
Microsoft
Innovative
American
System Plus
and more.

NOW IN STOCK!

HEWLETT PACKARD
2 NEW DELUXE CALCULATORS FROM HP!
Slim, shirt-pocket styling

NEW! HP-11C
Advanced Programmable Scientific LCD Retail \$130.00
Your Cost: \$114.95

NEW! HP-12C
Advanced Programmable Financial LCD Retail \$150.00
Your Cost: \$129.95

HP-125 New Microcomputer
64K CPU/Terminal/Keyboard Retail Your Cost
3150 00 3195 00

HP-85 Microcomputer—built-in printer/monitor
3250 00 2499 00

HP-83 Microcomputer—built-in monitor
2750 00 1799 00

HP-26318 Printer, dot mtr x (ask for optns) 3950 00 3199 00

HP82905A 80 col printer, dot matrix 945 00 755 00

HP-2681A Letter quality printer, daisy wheel 345 00

HP-82910 Dual master (256K disk drive) 2500 00 1999 00

Call us for the lowest prices on 7 disk drives Call us Call us

HP 41CV New! 2.2K bytes of memory
325 00 256 00

Card Reader for 41CV
215 00 171 00

Printer for 41CV
365 00 294 00

Quad Ram
95 00 84 95

Optical Wand for 41CV
125 00 99 95

HP 41C Calculator
250 00 188 95

Memory mod. for 41C
25 95

HP-87 Programmable printer
750 00 579 95

HP-34C Programmable scientific
150 00 117 95

HP-38C Programmable business R/E
150 00 119 95

We have the complete line of accessories, etc.

Texas Instruments Home Computer

New—1982 Model with full typewriter-style keyboard, TI-99/4A UI/L case & more!

\$369.95 Retail Your Cost \$950.00 \$369.95

New keyboard
650 00 319 95

32K Exp. mem. module
399 95 314 95

Extended Basic, a MUST for 32K module
100 00 75 00

Speech synthesizer
150 00 129 95

We carry a large inventory of software, & accessories

TI-30 2 LCS Stu Slide NEW 18.95
TI-35SP LCD SCIENTIFIC 22.50
TI-40 LCD Sci/NEW 28.95
TI-57 Prog. Scientific 39.95
TI-58C 480 Step. Prog. 89.95
TI-59 960 Step. Prog. 179.95
PC-100C Print/Plot 149.95
LCD-Programmer/NEW 59.95

XEROX
Model 8020 (15K) 8" \$3750.00

64K COMPUTER & WORD PROCESSOR
AS LOW AS \$2995.00

Special discount available to Schools & Institutions—Inquire! Required software add!

NEW—FAMOUS CORVUS DISK DRIVES—5, 10, 20 MEGABYTES with fantastic new OMNINET Network System

Call us for the best prices in the USA!

ATARI Computer Retail Your Cost

400 SPECIAL PRICE! 16K 595 00 339 95

No language inc., opt'l basic, 54 95

800 16K Computer 1080 00 759 95

SPECIAL! ATARI 800 48K Computer 1250 00 889 95

OHIO SCIENTIFIC
C&PDF—48K Retail: \$3495.00/Y.C. \$3195.00

• Dual 8" Drives • 64 col x 32 line/color
• 7 MIPS—FAST! • Many more std features
Fortran & Pascal available

Many other OSI products—at discounted prices

NEW! From TI—Series 10 Personal Information Terminal Retail 995.00 Your Cost 795.00

PRINTERS

• DIABLO (Letter Quality) Retail Your Cost
630 R102 bi-directional/tractors 2995 00 2699 00
1640K 109 keyboard, tractors 3077 00 2899 95
630 R0 Receive only 2710 00 2499 95
1650K 136 keyboard/tractors 3220 00 2999 95

• CENTRONICS dot matrix
700-9 Parallel, heavy duty 1460 00 1199 95
704-9 Serial, heavy duty 1795 00 1599 95
737-1 Parallel 935 00 799 95
737-3 Serial 1045 00 899 95
704-11 Parallel 1870 00 1695 00
P-1 Electrostatic 495 00 189 95

• PAPER TIGER
460 995 00 895 00
460G graphics 1094 00 969 95
560 1295 00 1059 00
560G graphics 1394 00 1195 00
445 884 00 695 00
445G 884 00 789 00

EPSON PRINTERS Retail Your Cost

MX80 645.00 539.95
Optional Graftrax Chip 80 95.00

MX80 FT 745.00 659.95
MX80 + GRAFTRAX 80 695.00 579.95
MX80 FT + GRAFTRAX 80 795.00 689.95
MX100 895.00 789.95

WE ALSO HAVE . . .

• MODAVION Modems Retail Your Cost
CAT 199 95 159 95
D-CAT 199 95 159 95
APPLE-CAT Direct connect 349 95 314 95

SANYO MONITORS High resolution

13" Color (new) high quality 550 00 419 95
12" Green phosphorous 360 00 259 95
12" Black & white 340 00 239 95
15" Black & white 370 00 259 95
9" Black & white (the best seller) 235 00 169 95

AMDEK(Leedex) High Quality Monitors

100 12" B/W, 12 MHz 179 00 139 95
100-G 12" Green, 12 MHz 199 00 174 95
300-G 12" Green, 18 MHz 249 00 199 95
Color I 13" Color, NTSC comp. input, 449 00 339 95
audio amp & speaker
Color II 13" Color, RGB input, hi res graphics, speaker 999 00 699 95

• HAZELTINE Video Display Terminals
• SHUGART Disk Drives
• DEC VT100 & VT103
Call us for your DISCOUNTED price TODAY!

Listing 1: The Interface

```

;*****
;*
;* BASIC INTERFACE *
;*
;* BY CHRISTER ENGSTROM *
;*
;*****
;
; ORG $F81
ENTRY LDA $75
STX $14
LDA $76
STX $15
SCAN LDY #$00
LDA ($14),Y
CMP #$AA
BEQ FUNCO
CMP #$C1
BNE NEW
INY
LDA ($14),Y
CMP #$C4
BEQ CHECK
NEW CLC
LDA $14
ADC #$07
STA $14
LDA $15
ADC #$00
STA $15
CLC
BCC SCAN
CHECK INY
LDA ($14),Y
BEQ FUNC
JUMP JSR $BEFE
LDA #$4C
STA $16
LDY #$02
LDA ($14),Y
STA $18
INY
LDA ($14),Y
STA $17
LDA $AD
JSR $0016
STA $AD
LDA $AC
LDY $AD
JSR $COD1
OUT LDA #$00
TAX
TAX
RTS
FUNC INY
LDA ($14),Y
CMP #$00
BEQ FUNCO
CMP #$01
BEQ FUN''
RTS
FUNCO JSR $OE00
CLC
BCC OUT
FUNCL NOP
CLC
BCC OUT

```

Listing 2: Hexadecimal Converter

```

;*****
;*
;* HEX CONVERTER *
;*
;* BY CHRISTER ENGSTROM *
;*
;*****
; ORG $OE00
START LDA $73
STA $14
LDA $74
STA $15
;
; START LOOP IN PROG
;
LOOP LDA $15
CMP $76
BMI GO
LDA $14
CMP $75
BMI GO
RTS
;
GO JSR SUBR
CLC
LDA $14
ADC #$01
STA $14
LDA $15
ADC #$00
STA $15
CLC
BCC LOOP
;
; CHECK PROGRAM
;
SUBR CLD
LDY #$00
LDA 'X
CMP ($14),Y
BEQ CONT
LDA 'Z
CMP ($14),Y
BNE NEXT
;
CONT STA $16
LDA ''
INY
CMP ($14),Y
BNE NEXT
LDA '0
STX $18
STX $19
STX $1A
STX $1B
LDA $200
LDA $B34
LDA $B18
STX $33
NOP
;
IN INY
LDA ($14),Y
CMP ''
BEQ UT
CPY #$06
BEQ NEXT

```

(Continued)

Listing 2 (Continued)

```

OE5E C930      CMP 'O
OE60 300C      BMI NEXT
OE62 C93A      CMP #$3A
OE64 30ED      BMI IN
OE66 C941      CMP 'A
OE68 3004      BMI NEXT
OE6A C947      CMP 'G
OE6C 30E5      BMI IN
OE6E           ;INVALID/NO STRING
OE6E 60        NEXT RTS
OE6F           ;
OE6F           ;MOVE RIGHT JUST
OE6F           ;
OE6F 8417      UT STY $17
OE71 A203      LDX #$03
OE73 8636      STX $36
OE75 8435      STY $35
OE77 C635      IN2 DEC $35
OE79 A435      LDY $35
OE7B B114      LDA ($14),Y
OE7D C922      CMP ""
OE7F F009      BEQ UT2
OE81 A436      LDY $36
OE83 9133      STA ($33),Y
OE85 C636      DEC $36
OE87 18        CLC
OE88 90ED      BCC IN2
OE8A           ;
OE8A           ;STRING TO HEX
OE8A           ;
OE8A A003      UT2 LDY #$03
OE8C B133      UT2A LDA ($33),Y
OE8E 207DE2    JSR $EA7D
OE91 9133      STA ($33),Y
OE93 88        DEY
    
```

Listing 2 (Continued)

```

OE94 10F6      BPL UT2A
OE96           ;START COUNT LOOP
OE96 A200      LDX #$00
OE98 8622      STX $22
OE9A 8623      STX $23
OE9C 8624      STX $24
OE9E F8        SED
OE9F A940      LDA #$40
OEA1 8525      STA $25
OEA3 A996      LDA #$96
OEA5 8526      STA $26
OEA7 A518      LDA $18
OEA9           ;TEST BIT 1
OEA9 C908      CMP #$08
OEAB 300D      BMI NOMI
OEAD A516      LDA $16
OEAF C958      CMP 'X
OEB1 D007      BNE NOMI
OEB3 A9A5      LDA #$A5
OEB5           ;FLAG < 0
OEB5 8516      STA $16
OEB7 18        CLC
OEB8 9004      BCC IN3
OEBA A930      NOMI LDA '0
OEBB           ;FLAG >= 0
OEBB 8516      STA $16
OEBE C618      IN3 DEC $18
OEC0 3005      BMI UT3
OEC2 20F60E    JSR ADD
OEC5 90F7      BCC IN3
OEC7 A902      UT3 LDA #$02
OEC9 8525      STA $25
OECB A956      LDA #$56
OECD 8526      STA $26
OECF C619      IN4 DEC $19
OED1 3005      BMI UT4
OED3 20F60E    JSR ADD
OED6 90F7      BCC IN4
OED8 A900      UT4 LDA #$00
OEDA 8525      STA $25
OEDC A916      LDA #$16
OEDE 8526      STA $26
OEE0 C61A      IN5 DEC $1A
OEE2 3005      BMI UT5
OEE4 20F60E    JSR ADD
OEE7 90F7      BCC IN5
OEE9 A901      UT5 LDA #$01
OEEB 8526      STA $26
OEED C61B      IN6 DEC $1B
OEEF 301A      BMI UT6
OEF1 20F60E    JSR ADD
OEF4 90F7      BCC IN6
OEF6           ;ADD TO RESULT
OEF6 18        ADD CLC
OEF6           ;ADD TO RESULT
OEF7 A524      LDA $24
OEF9 6526      ADC $26
OEFB 8524      STA $24
Oefd A523      LDA $23
OEFF 6525      ADC $25
OF01 8523      STA $23
OF03 A522      LDA $22
OF05 6900      ADC #$00
OF07 8522      STA $22
OF09 18        CLC
OFOA 60        RTS
OF0B           ;
OF0B           ;SIGN CONVERSION
    
```

(Continued)

POWER TO YOUR AIM

Treat your AIM to a quality power supply:

1. Designed to Rockwell's specifications for the AIM-65 (5 volts at 2 amps, regulated; 24 volts, .5 amps avg., 2.5 amps peak, unregulated).
2. *Overtoltage protection* to protect the expensive circuits in your AIM (5 volt output).
3. Handsome all metal case (two tone blue).
4. Fuse (externally accessible), switch, pilot light, line cord, cable from power supply to AIM — all included.
5. Conservative thermal design for long life.

\$64.95 ppd. (continental U.S.)

CA residents add 6 % sales tax.
 VISA/MC, cashier's or registered check.
 Personal check (allow 2 weeks to clear).

CompuTech
 Box 20054
 Riverside, CA 92516

Listing 2 (Continued)

```

OFOB
OFOB A516      ;
OFOD           UT6   LDA $16
               ;CHECK MINUS
OFOD C9A5      CMP #5A5
OFOD D013      BNE IN7
OF11 38        SEC
OF12 A936      LDA #36
OF14 E524      SBC $24
OF16 8524      STA $24
OF18 A955      LDA #55
OF1A E523      SBC $23
OF1C 8523      STA $23
OF1E A906      LDA #06
OF20 E522      SBC $22
OF22 8522      STA $22
OF24           ;
OF24           ;EDIT NOW
OF24           ;
OF24 D8        IN7   CLD
OF25 EA        NOP
OF26 A516      LDA $16
OF28 8535      STA $35
OF2A A522      LDA $22
OF2C 20560F    JSR LEFT
OF2F 8536      STA $36
OF31 A522      LDA $22
OF33 205C0F    JSR RIGHT
OF36 8537      STA $37
OF38 A523      LDA $23
OF3A 20560F    JSR LEFT
OF3D 8538      STA $38
OF3F A523      LDA $23
OF41 205C0F    JSR RIGHT
OF44 8539      STA $39
OF46 A524      LDA $24
OF48 20560F    JSR LEFT
OF4B 853A      STA $3A
OF4D A524      LDA $24
OF4F 205C0F    JSR RIGHT
OF52 853B      STA $3B
OF54 9014      BCC UT7
OF56 29F0      LEFT  AND #5F0
OF58 6A        ROR
OF59 6A        ROR
OF5A 6A        ROR
OF5B 6A        ROR
OF5C 290F      RIGHT AND #50F
OF5E 18        CLC
OF5F 6930      ADC '0
OF61 C93A      CMP #3A
OF63 1001      BPL MORE
OF65 60        RTS
OF66 6907      MORE  ADC #507
OF68 18        CLC
OF69 60        RTS
OF6A           ;
OF6A           ;MOVE TO PROG
OF6A           ;
OF6A A935      UT7   LDA #35
OF6C 8533      STA $33
OF6E A006      LDY #06
OF70 8416      STY $16
OF72 B133      IN8   LDA ($33),Y
OF74 C616      DEC $16
OF76 A417      LDY $17
OF78 9114      STA ($14),Y
OF7A A416      LDY $16
OF7C C617      DEC $17
OF7E 10F2      BPL IN8
OF80 60        RTS
    
```

MICRO



WHEN DID WE ORDER IT?
 WHERE IS IT LOCATED?
 HOW MANY DID WE BUILD?
 WHAT IS MY INVENTORY WORTH?
 WHAT ARE MY PRODUCTION COSTS?

ANSWER THESE QUESTIONS
 AND MORE
 WITH
**KEYSTONE DATA'S
 INVENTORY MANAGEMENT SYSTEM**

- BILL OF MATERIALS PROCESSING
- AUTOMATIC OR MANUAL REORDERING
- REVIEW OF DAILY ACTIVITY
- MATERIAL & LABOR COSTING
- DISTRIBUTED PROCESSING CAPABILITY
- REMOTE SITE COMMUNICATIONS
- LINE ITEM CROSS REFERENCE BY -
 ASSEMBLY
 VENDOR
 PART TYPE
- COMPREHENSIVE USER'S MANUAL

AVAILABLE FOR
 HDE DISK SYSTEMS

CONTACT -



KEYSTONE
 DATA CONSULTANTS, INC.

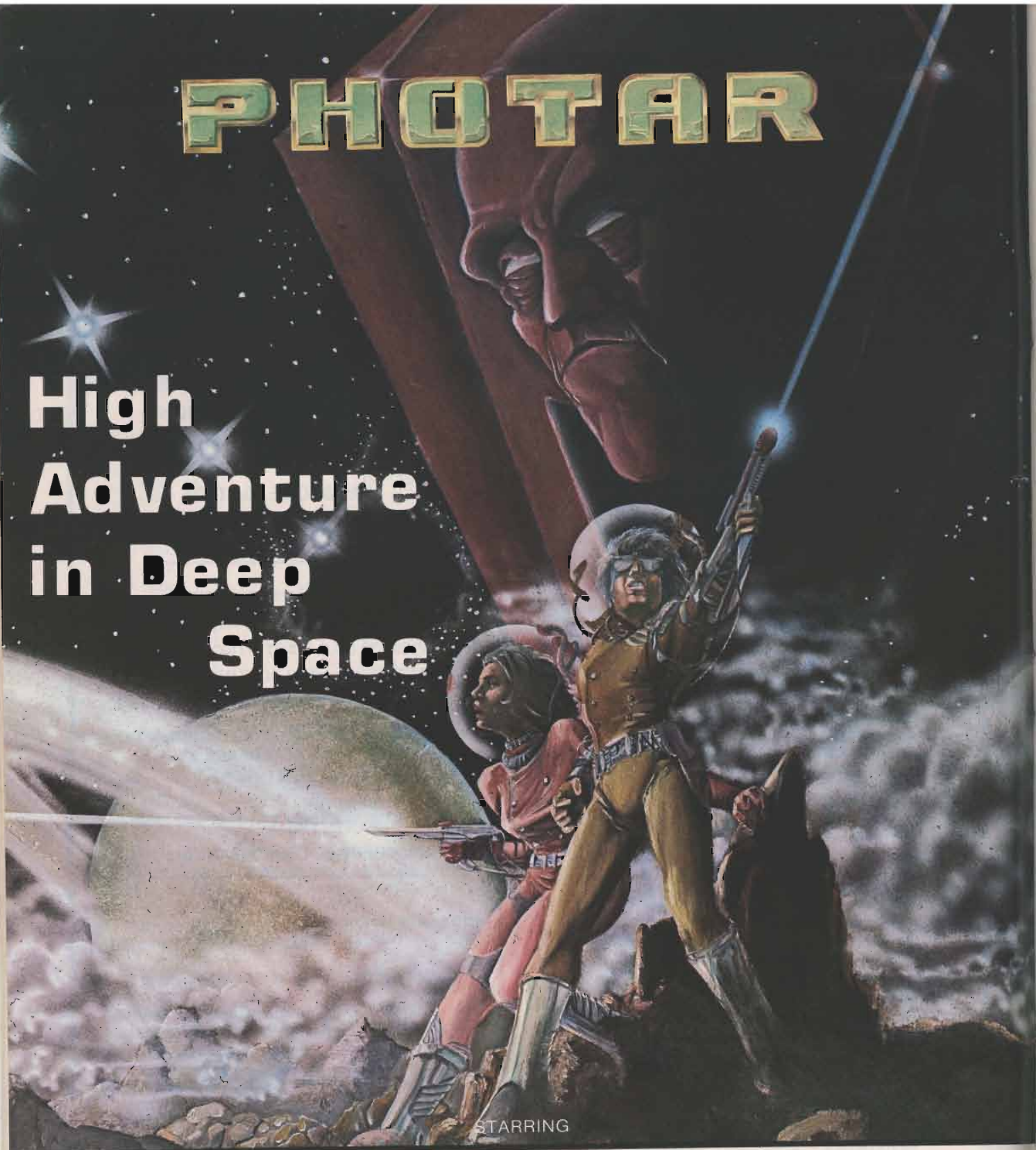
CUSTOMER SERVICE
 P.O. BOX 606
 YORK, PA 17405

717-848-5666

MEMBER:
 York Area Chamber of Commerce
 National Federation Independent Businesses (N.F.I.B.)

PHOTAR

High
Adventure
in Deep
Space



STARRING

PHOTAR ■ THE RINGS OF SATURN ■ PULSARS ■ KILOSTARS ■ BLACK STARS
IN

The Ultimate Experience in Outer Space Adventure

Produced by

SOFTAPE

An Artsci Company

The Ultimate Experience in Outer Space Adventure
The Ultimate Experience in Outer Space Adventure

Screenplay by

SAB

Distributed through Softape, Softsel, and other computer stores world-wide

10432 BURBANK BOULEVARD ■ NORTH HOLLYWOOD, CALIFORNIA 91601 ■ 213/985-5763

Memory Moves with the 6502 and 6809

by Gregory Walker and Tom Whiteside

The authors demonstrate the advantages of the 6809's direct page addressing and 16-bit index registers with a comparison of 6502 and 6809 memory moves.

In a previous article (MICRO 47:57) we illustrated the advantages of programming the 6809 over the 6502 with a comparison of multiprecision arithmetic routines. We continue in this article with a comparison of the two processors' capabilities in solving memory move problems.

With these two articles, we hope to demonstrate that the MC6809 is not only faster and more byte-efficient than the 6502, but also more straightforward to program. Because the most useful kinds of 6502 indirect addressing must be done through page zero, it is important, particularly with larger operating systems, that page-zero RAM be used wisely. In addition, the 6502 index registers are only eight bits long, limiting indexing to a 256-byte range. These limitations will show themselves especially well in these memory move examples.

Memory moves have a number of practical applications, including word processing, EPROM programming, and program relocation. Similar techniques are involved with string manipulation and table searches.

Figure 1 is a 6502 memory move for fewer than 256 bytes from a fixed absolute address. This routine is not all that useful, since it can only work on two fixed pages due to the limited range of the 6502 index registers. However, it illustrates the real power of the 6502 in terms of byte efficiency and speed over *small* ranges of memory.

The program uses the fastest form of 6502 indexed addressing — absolute indexed. The Y register will be used

both as a loop index for the move and as a counter for the number of bytes to be moved. The Y register is initialized to the number of bytes to be moved and is decremented each time through the loop. When the Y register decrements to zero, the branch conditions are not met and the loop terminates. This use of the Y register eliminates the need for a CPY immediate instruction in the loop and speeds up the code. A "CNT" value of zero will move 256 bytes.

In these examples, the "LNG" column in figure 1 represents the number of bytes required per instruction. The "TIM" column is the number of machine cycles per instruction. The 6502 memory move for fewer than 256 bytes of memory required only 11 bytes of code and approximately 14 machine cycles per byte moved.

Figure 2 shows the same memory move written in MC6809 code. In this example, the 16-bit X register points to the "FROM" address and the U register points at the "TO" address. The MC6809 addressing mode used is indexed with accumulator offset. The effective address is formed by summing the two's complement contents of the B accumulator with the contents of the index register used. You will notice that the B accumulator is being used in the same manner as the 6502 Y register was in figure 3. Because the offset is two's complement, the MC6809 example is limited to 127 bytes. We included this example to show how similarly the two processors can be used to solve the same problem. The MC6809 took 15 bytes and 15 machine cycles per byte moved.

While the 6502 wins this round by four bytes and one machine cycle per

Figure 1: 6502 program to move fewer than 256 bytes of memory. Timing = 2 + 14 * N where N is the number of bytes to move.

		[LNG TIM]		
	LDY #CNT	2	2	INITIALIZE THE BYTES TO MOVE COUNT
LOOP	LDA FROM-1, Y	3	4	LOOP: GET BYTE TO MOVE
	STA TO-1, Y	3	5	MOVE BYTE
	DEY	1	2	DECREMENT LOOP COUNTER
	BNE LOOP	2	3	LOOP UNTIL ZERO COUNT
		11		

Figure 2: MC6809 program to move fewer than 128 bytes of memory. Timing = 8 + 15 * N.

	LDX #FROM-1	3	3	INITIALIZE "FROM" POINTER
	LDU #TO-1	3	3	INITIALIZE "TO" POINTER
	LDB #CNT	2	2	INITIALIZE BYTES TO MOVE COUNT
LOOP	LDA B, X	2	5	LOOP: GET BYTE
	STA B, U	2	5	AND MOVE IT
	DECB	1	2	DECREMENT LOOP COUNTER
	BNE LOOP	2	3	LOOP UNTIL COUNT IS ZERO
		15		

byte over the MC6809, the MC6809 code is more versatile. If this were a subroutine, "LOOP" could be called with X and U pointing anywhere in memory, while the 6502 example would be limited to the 256-byte range of its index registers. Because the MC6809 code holds the pointers in registers instead of memory locations, it is re-entrant and could be used in a real-time operating system.

Since it appears that the 6502 can keep up with the MC6809 on a short memory move, let's try another memory move. This time the code must be able to move any number of bytes. A real measure of a processor's power is how much its performance degrades as the complexity of its task increases. In this example, complexity is measured in terms of address range.

Figure 3 shows a 6502 program to move any size block of memory. "CNT" bytes will be moved from address "FROM" to address "TO". The bytes will be moved starting at address "FROM" plus "CNT," with "CNT" decremented each time through the loop. Since the 6502 index registers are only eight bits wide, it is necessary to use indirect indexed addressing to move more than 256 bytes. [We do not count self-modifying code as an option, but as an abomination!]

The first part of the program sets up the two 16-bit zero-page pointers "FROM" and "TO". "CNT" is a 16-bit number stored in the X register (least significant byte) and "COUNT" (most significant byte). The X register is used to store the least significant byte of "CNT" to save three machine cycles per byte moved over using a zero-page variable. The "CNT" most significant byte is initialized to one count higher than desired to eliminate the need for a load/compare step that would have used time and bytes. The pointer least significant bytes are incremented rather than the Y register, since the "FROM" and "TO" least significant bytes might not be the same.

If the above "tricks" seem confusing to you, you are not alone. Tricks cost money in terms of debug time and the time required to recode the routine when some performance requirement is changed. However, we tried to write the best 6502 code possible. The result is typical of commercial practice. Even with all these tricks, the 6502 code was 47 bytes long and approximately 32 machine cycles per byte. This is more than four times the size, and twice the cycles per byte over the short memory move. Now you see what we meant about performance degradation with increased task complexity!

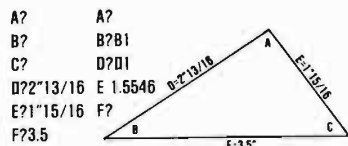
Figures 4 and 5 show two versions of MC6809 memory moves which can handle memory ranges of more than 256 bytes. Both examples are identical except that one (figure 4) moves memory one byte at a time while the other (figure 5) moves two bytes at once. Both require 18 bytes of code, but the second program is eight machine cycles per byte [40%!] faster than the first.

In both these programs, the X register acts as a pointer to the "FROM" address and the U register acts as a pointer to the "TO" address, just like the program in figure 2. In these routines, however, the index registers are incremented each time through the loop as indicated by the "+" beside the indexed loads and stores. Since the second program moves two bytes at a time, the MC6809 double increment ("++") mode is used to advance to the next word. In both figures, the Y register is a counter to the number of bytes remaining to be counted. The "LEAY" instruction has no 6502 equivalent and indicates that the Y register is to be loaded with the "effective" address indicated in the operand field. In figure 4, the operand field of the LEAY instruction means to load Y with the contents of Y minus 1 like a 16-bit 6502 "DEY".

THE TRIANGULATOR

Solve triangles on your Apple* and get rid of your calculator and pencil!!

This program will solve any right or oblique triangle. Results can be printed or used on further calculations. Previous results are recalled with a simple letter/number pair and can be added or subtracted from each other or from new data. For example, the height of the triangle below is calculated as follows:



Entry can be fraction, decimal, or DMS. Results are rounded to 4 places. Requires 48k Apple II+. DOS 3.2 or 3.3. parallel printer.

Send \$39.95 + \$1.50 postage and handling to:

Arrow Data Systems
1224 E. Harmont
Phoenix, AZ 85020

*Apple is a registered trademark of Apple Computer, Inc.

Dealer inquires invited. (602) 997-6638

Figure 3: 6502 program to move any number of bytes of memory. Timing = 31 + (35 + 28/256) * N.

	(LNG)	(TIM)	
LDA #FROML	2	2	INITIALIZE INDIRECT "FROM" PTR
STA FROM	2	4	
LDA #FROMH	2	2	
STA FROM + 1	2	4	
LDA #TOL	2	2	INITIALIZE INDIRECT "TO" PTR
STA TO	2	3	
LDA #TOH	2	2	
STA TO + 1	2	3	
LDA #CNTH + 1	2	2	INIT BYTES TO MOVE COUNT MSB
STA COUNT	2	3	TO (COUNT / 256) + 1
LDX #CNTL + 1	2	2	INIT X TO THE COUNT LSB
LDY #0	2	2	INITIALIZE INDIRECT POINTER
LOOP LDA (FROM), Y	2	5	LOOP: GET A BYTE
STA (TO), Y	2	6	AND MOVE IT
INC TO	2	5	INCREMENT 16-BIT "TO"
BNE NOINC1	2	4	POINTER
INC TO + 1	2	5	
NOINC1 INC FROM	2	5	INCREMENT 16-BIT "FROM"
BNE NOINC2	2	4	POINTER
INC FROM + 1	2	5	
NIINC2 DEX	1	2	DECREMENT 16-BIT "CNT"
BNE LOOP	2	4	
DEC COUNT	2	5	
BNE LOOP	2	4	LOOP UNTIL "CNT" IS ZERO
		47	

Figure 4: MC6809 program to move any length of memory. Timing = 10 + 20 * N.

		(LNG TIM)	
LDX #FROM	3	3	INITIALIZE 16-BIT "FROM" POINTER
LDU #TO	3	3	INITIALIZE 16-BIT "TO" POINTER
LDY #CNT	4	4	INITIALIZE BYTES TO MOVE COUNT
LOOP LDA , X +	2	6	LOOP: GET BYTE TO MOVE; BUMP
STA , U +	2	6	POINTER; MOVE WORD; BUMP
LEAY -1, Y	2	5	POINTER; DECREMENT COUNT
BNE LOOP	2	3	BY ONE UNTIL COUNT IS ZERO
	18		

Figure 5: MC6809 program to move any length of memory. Timing = 10 + 12 * N.

		(LNG TIM)	
LDX #FROM	3	3	INITIALIZE 16-BIT "FROM" POINTER
LDU #TO	3	3	INITIALIZE 16-BIT "TO" POINTER
LDY #CNT	4	4	INITIALIZE BYTES TO MOVE COUNT
LOOP LDD , X + +	2	8	LOOP: GET WORD TO MOVE; BUMP
STD , U + +	2	8	POINTER; MOVE WORD; BUMP
LEAY -2, Y	2	5	POINTER +2; DECREMENT
BNE LOOP	2	3	COUNT BY TWO UNTIL COUNT IS ZERO
	18		

Since the second program moves words, Y gets decremented with the contents of Y minus 2.

The program in figure 6 combines the code from figures 4 and 5 to produce a fast, general-purpose memory move for the MC6809 which moves any number of bytes, a word (two bytes) at a time. This routine uses the powerful double-byte move code of figure 5, only without the even-byte restriction. The way this is achieved is straightforward. The "CNT" word is tested for odd length by first using the "TFR" instruction to move the "CNT" to the D register. This is followed by a "LSRB" [logical shift right B] which sets the carry bit if "CNT" is odd. If the length is even, the routine branches directly to the double-byte move routine. Otherwise, the "odd" byte is moved first using the figure 2 code. This routine is 29 bytes long and takes approximately 12 machine cycles per byte moved. The general purpose routine takes almost twice the bytes of the MC6809 short move but requires 20% less time per byte!

Figure 7 summarizes the results for the memory moves discussed in figures 1 through 6. The byte ratio column is the number of 6502 bytes for a given comparison. The cycles per byte ratio column is the 6502 cycles required, per byte moved, divided by the MC6809 cycles per byte. For example, the row labeled "<= 256 bytes" shows that the 6502 program from figure 1 used 11 bytes and needed about 14 cycles per byte moved. The MC6809 program in figure 2 needed 15 bytes and used 15 cycles per byte moved. The "byte ratio" is then 11/15 or 0.73. The "cycles per byte ratio" is 14/15 or 0.93.

As the table in figure 7 shows, the 6502 is good at moving small blocks of memory with fixed addressing. The MC6809 code for a move of fewer than 256 bytes comes close to keeping up with the 6502, but requires over a third more bytes. Our general-purpose double-byte move routine is slightly faster than the 6502 but is much more costly in terms of bytes. Since the MC6809 general purpose routine is

MICRObits

6800/6809 Software

Includes compatible single-user, multi-user and network-operating systems, compilers, accounting and word processing packages. Free catalog.

Software Dynamics
2111 W. Crescent, Sta. G
Anaheim, CA 92801

OSI Superboard Cabinet and Accessories

Pre-cut pine cabinet kit \$27.95 ppd; RS-232 interface kit \$9.95 ppd; 600 baud/band kit \$8.95 ppd., complete with hardware and well-illustrated instruction sets. Free catalog of software, hardware, Molex, joysticks, cassettes, kits and more.

Dee Products
Department M
150 Birchwood Road
Lake Marion, IL 60110

Expand Apple Memory!

Memory boards from 16K to 256K by Microsoft, Saturn, Legend, SVA, etc. at lowest prices. Also, 8" floppy drives and hard disks by LOBO, CORVUS, and SVA at big discounts!

Connecticut Informations Systems
218 Huntington Road
Bridgeport, CT 06608
(203) 579-0472

PET/CBM Owners

Real world software at low cost. 2114 RAM adapter and 4K Memory Expansion for "old" 8K PETs. Write for free catalog!

Optimized Data Systems
Dept. M, Box 595
Placentia, CA 92670

Lessons in Algebra

An easy and fun way to learn the basic elements of high school algebra. Apple computer diskette \$29.95. 30-day money guarantee if not satisfied.

George Earl
1302 South General McMullen Dr.
San Antonio, TX 78237

Telex Hard Disk Drive

Capacity - 100 megabytes; compatible with IBM-3330; excellent condition. Asking \$2500.00 plus shipping.

J. Hoogstraat
Box 20, Site 7, SS 1
Calgary, Alberta
T2M-4N3
Canada 403-239-0900

(Continued on page 87)

A harvest of savings from



SOFTWARE

APPLE • ATARI • TRS80 • IBM
A full line of software for business, games and education **up to 35% off!**

MCISE	ICIS
VISICORP	STONEWARE
ON LINE	SYNERGISTIC
EDU-WARE	HAYDEN
HOWARD	AND MANY MORE

HARDWARE

AMDEK • HAYES • MICROSOFT

	List	Our Price
32K RAM card	\$293.00	\$205.00
Video Term	\$345.00	\$279.00
Lazer Products	--- 20% off ---	

DISKS

Maxell Box of 10, 5 1/4", SS-DD \$35.00
Verbatim Box of 10, 5 1/4", SS-DD \$29.00

MONITORS

LE MONITORS	List	Our Price
9" Green	\$189.00	\$159.00
12" Green	\$199.00	\$169.00
ZENITH		
12" Green	\$179.00	\$129.00

Plus a full line of **AMDEK** Monitors

PRINTERS

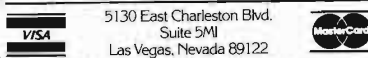
PAPER TIGER	List	Our Price
460G	\$1,094.00	\$950.00
560G	\$1,394.00	\$1,250.00
EPSON		
MX 70	\$449.00	\$395.00
MX 80FT	\$745.00	\$595.00
MX 100FT	\$945.00	\$795.00

CALL FOR THIS MONTHS SPECIAL!

1-800-835-2246 EXT. 211

OR

702-452-5589



5130 East Charleston Blvd.
Suite 5M1
Las Vegas, Nevada 89122

Phone orders welcome. Mail orders may send charge card number (include expiration date), cashiers check, money order or personal check (allow ten business days for personal or company checks to clear). Add \$3.00 for shipping, handling and insurance. Nevada residents add 5.75% sales tax. Please include phone number. All equipment is in factory cartons with manufacturers warranty. Equipment subject to price change and availability. Call or write for price list.

Figure 6: General MC6809 program to move any length of memory. Timing = 41 + 12 * N.

		[LNG TIM]	
LDX #FROM	3	3	INITIALIZE "FROM" POINTER
LDU #TO	3	3	INITIALIZE "TO" POINTER
LDY #CNT	4	4	INITIALIZE BYTES TO MOVE COUNT
			CHECK FOR ODD COUNT
TFR Y, D	2	6	MOVE COUNT LSB TO B
LSRB	1	2	REGISTER; SET CARRY IF COUNT IS ODD; TO LOOP IF COUNT IS EVEN
BCC LOOP	2	3	
			ELSE GET ODD BYTE; BUMP POINTER; MOVE IT; BUMP POINTER; DECREMENT LOOP COUNT; QUIT COUNT IS ZERO
LDA ,X+	2	6	
STA ,U+	2	6	
LEAY -1, Y	2	5	
BEQ DONE	2	3	
LOOP LDD ,X++	2	8	LOOP: GET NEXT WORD; BUMP POINTER+2; MOVE IT; BUMP POINTER+2; DECREMENT LOOP COUNT BY TWO UNTIL COUNT IS ZERO
STD ,U++	2	8	
LEAY -2, Y	2	5	
BNE LOOP	2	3	
DONE EQU *		29	

Figure 7: 6502/MC6809 byte and cycles per byte ratios for figures 1 through 6.

Class of Move	Byte Ratio	Cycles/Byte Ratio
< = 256 bytes		
6502-Fig 1 / MC6809-Fig 2	11 / 15 = 0.73	14 / 15 = 0.93
6502-Fig 1 / MC6809-Fig 6	11 / 29 = 0.38	14 / 12 = 1.17
> 256 bytes		
6502-Fig 3 / MC6809-Fig 4	47 / 18 = 2.61	35 / 20 = 1.75
6502-Fig 3 / MC6809-Fig 5	47 / 18 = 2.61	35 / 12 = 2.92
6502-Fig 3 / MC6809-Fig 6	47 / 29 = 1.62	35 / 12 = 2.92

easily made into a subroutine, the extra byte cost might be lessened by sharing the code with other parts of a program. The 6502 code example lacks this versatility since it is limited to fixed 256 byte ranges.

The more complex memory move of more than 256 bytes is where the MC6809 really asserts itself. MC6809 versions were presented for a single byte move, an even-length-only double-byte move, and a general-purpose "any length" move. For "byte tight" applications, the MC6809 byte mover runs 1.75 times faster than the 6502, while the 6502 uses 2.6 times the bytes of the MC6809. While the MC6809 double-byte mover from figure 5 is restricted to even-byte moves only, it rips along at almost three times the rate of the 6502 with no more code than the single-byte version.

The MC6809 general-purpose double-byte mover (figure 6) maintains the blazing speed of figure 5 without being restricted to even-byte moves. The 6502 move uses 1.6 times the bytes of the MC6809 general purpose mover.

These results show clearly the degradation of speed and code size of the 6502 for memory moves across page boundaries. We feel that the MC6809 has also been easier to program. There has been no need to set up and manipulate indirect pointers with registers of only eight bits, as was necessary on the long 6502 memory move.

The authors may be contacted at Motorola, Inc., Microprocessor Design, Maildrop M2880, 3501 Ed Bluestein Blvd., Austin, Texas 78721.

MICRO

CLASSIFIED
(APPROVED FOR RELEASE TO APPLE USERS)

"You put *what* in your Apple computer?"

Investments. Account data. Wills. Phone lists. Sooner or later you will create a file into which you will write some very confidential information. And that's why Passage Research has developed a special utility that will encrypt that DOS 3.3 file through software routines rather than the expensive (and conspicuous) black-box approach. Now you have the means to keep your private affairs private. Totally private.

The bits and bytes of your file will get hopelessly mixed up according to a cipher "key" that you keep as secret as you want. Over one hundred million billion different keys are available for selection. (This many keys may sound like a lot, but it's typical of a modern encryption process.)

To make your encrypted file understandable again, all you have to do is reactivate the appropriate key schedule and run the file back through. It's easy.

With this utility (and a few easy-to-learn mnemonic commands) you can create a personalized encryption algorithm that is specially tailored to your needs. If you want, you can execute successive encryptions with different keys and then strip off the outer layers, one by one, to reveal the original text. And you can call many routines from your own application programs to do "codebook" encryption (documentation included).

Passage Research

945 Turquoise St., Suite G
San Diego, California 92109

\$39.50 postpaid. California residents add 6% sales tax. Specify Software Package No. U-17 for Apple II or Apple II+. Memory requirement: standard 48K. Due to time required for checks to clear financial institutions, please allow 3-5 weeks for disk delivery. Software documentation shipped immediately upon receipt of order.

(Apple is a registered trademark of Apple Computer, Inc.)

6809 Small-C

More bang, less buck! WW Small-C 1.0, with separate optimizer. Has all C control structures, including do/while, for, and switch/case. Generates relocatable modules for either TSC absolute assembler or SSB MACRO. (Version 2.0 is also planned for OS-9.) FLEX9 version includes RLOAD 3.0 on separate disk. Run-time library source included. 48K recommended.

For FLEX9 (with loader)	\$52.50
(If you already have RLOAD)	\$47.50
RLOAD 3.0 separately	\$17.50
For DOS69D (specify assembler and CPU)	\$47.50

Shipping included. Prices good until version 2.0 release (about July). Liberal upgrade policy. Add \$2/disk for 8". Add \$2 handling for Visa/MC. Allow 4 weeks for non-certified check. Please don't mail cash. Payment must accompany PO. Texas residents: add \$0.25 sales tax each 5" disk, \$0.35 each 8" disk.

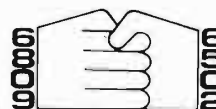
DOS69D is a trademark of SSB. FLEX is a trademark of TSC. OS-9 is a trademark of Microware.

word's worth

P.O. Box 28954
Dallas, Texas 75228

EXCEL-9

The Ultimate 6809 Board for Apple



- EXCEL-9 FLEX, a famous DOS, Assembler and Editor included.
- Also able to use Apple DOS.
- 8KB versatile monitor contains 35 commands including 6809.
- Can handle all Apple slot I/O routine from EXCEL-9.
- On-board programmable timer for both 6809 and 6502 systems allows printer spooling, multitask, etc.
- 50 page well documented manual.
- 64K RAM area expandable for multi-MPU operation.
- Able to switch MPU from 6809 to 6502 and vice versa in both machine code routine and BASIC.
- TSC 6809 BASIC, EXTENDED BASIC, PRECOMPILER, SOFT/MERG, etc., are coming soon.

Ask your nearest dealer
or

ESD LABS CO., LTD.

c/o AbCom
P.O. Box 5203
Mission Hills, CA 91345

Dealer Inquiries are Invited.

Introductory Price: **\$399.95** for Board & FLEX diskette
(Sales tax not included)

- FLEX is a trade mark of Technical Systems Consultants, Inc.



ATARI 800
16K ... \$679
32K ... \$749
48K ... \$819

410 Recorder	\$76.00
810 Disc Drive	\$449.00
822 Printer	\$269.00
825 Printer	\$629.00
830 Modem	\$159.00
820 Printer	\$269.00
850 Interface	\$169.00
New DOS 2 System	\$29.00
CX30 Paddle	\$18.00
CX40 Joy Stick	\$18.00
CX853 16K RAM	\$89.00
Microtek 16K RAM	\$75.00
Microtek 32K RAM	\$159.00
Ramdisk (128K)	\$539.00
One year extended warranty	\$50.00
INTEC 48K Ram	\$249
481 Entertainer	\$83.00
482 Educator	\$130.00
483 Programmer	\$57.00
484 Communicator	\$344.00

ATARI 400



16K. . . . \$329
32K. . . . \$478
48K. . . . \$555

Visicalc	\$169.00
Letterperfect	\$109.00
Ricochet	\$14.50
Crush Crumble & Chomp	\$24.00
Star Warrior	\$29.00
Rescue at Rigel	\$24.00
Datstones	\$16.00
Invasion Orion	\$18.50
Mission Asteroid	\$22.00
MouskATTACK	\$31.00
The Next Step	\$34.00
Softporn	\$27.00
Wizard & Princess	\$29.00
K-BYTE Crazy Shoot Out (ROM)	\$39.00
Protector (Disk 32K)	\$32.00
Jaw Breaker (on line disk)	\$27.00
Ghost Hunter (cassette)	\$24.00
Ghost Hunter (disk)	\$30.00
PAC MAN	\$35.00
Centipede	\$35.00
Caverns of Mars	\$32.00

We carry crystal software and Compumax Programs



TI-99/4A \$299

Telephone Coupler	\$169.00
RS 232 Accessories Interface	\$169.00
Disk Drive Controller	\$239.00
Disk Memory Drive	\$389.00
Memory Expansion (32K RAM)	\$319.00
R F Modulator	\$43.00
Wired Remote Controllers(Pair)	\$31.00
32K Expansion	\$329.00
PHP Printer Solid State	\$319.00
Home Financial Decisions	\$26.00
Personal Record Keeping	\$43.00
Mailing List	\$60.00
Checkbook Manager	\$18.00
Tombstone City 21st Century	\$34.00
Munch Man	\$34.00
TI Invaders	\$34.00
Car Wars	\$34.00



Xerox 820	
System I 5 1/4"	\$2450.00
System II 8"	\$2950.00
CPM 5 1/4"	\$169.00
Word Processing	\$429.00
Super Calc	\$269.00

Monitors

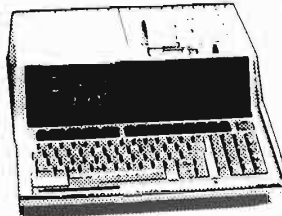
Amdc 12" B&W	\$129.00
12" Green	\$139.00
13" Color	\$349.00
NEC	
12" B&W	\$169.00
12" Color	\$339.00
TI 10" Color	\$349.00

Terminals

Televideo	
910	\$579.00
912C	\$699.00
920C	\$749.00
950	\$939.00
Call for computers	
Zenith Z19	\$749.00
Adds	\$549.00

Modems

Novation Auto	\$239.00
D Cat	\$169.00
Cat	\$159.00
Hayes Smart	\$239.00
Livermore Star	\$119.00



HP-85 \$1999

80 Column Printer	\$799.00
HP-125	\$1999.00
HP-83	\$1699.00
HP-85 16K Memory Module	\$169.00
5 1/4" Dual Master Disc Drive	\$1929.00
Graphics Plotter (7225B)	\$2079.00
NEW! HP-87	\$1999.00
Hard Disk w/Floppy	\$4349.00
Hard Disk	\$3440.00
"Sweet Lips" Plotter	\$1149.00

HP41CV Calculator \$249

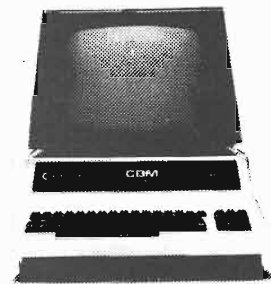
41 C	\$189.00
11 C	\$119.00
12 C	\$129.00
34 C	\$117.00
38 C	\$119.00
HP-41 Printer	\$340.00

HPIL CALCULATOR PERIPHERALS

IL Modul	\$104.00
Digital Cassette	\$449.00
Printer/Plotter	\$419.00
Card Reader	\$164.00
Optical Wand	\$99.00

PRINTERS

Centronics 739-3	\$619.00
Centronics 739-1	\$519.00
Dialbo 630 Special	\$1799.00
Epson	
MX70	\$359.00
MX80	\$469.00
MX80FT	Call
MX100	Call
NEC	
8023	\$549.00
7730	Call
7720	Call
7710	Call
Okidata	
82A	\$499.00
83A	\$769.00
84	\$1129.00
Citih Starwriter	
F10-40 CPS	\$1469.00
F10-55 CPS	Call
Paper Tiger	
445G	\$699.00
460G	\$899.00
560G	\$1129.00
Talley	
8024-7	\$1399.00
8024-L	\$1629.00



8032 \$1069

4032	\$969.00
4016	\$769.00
8096 Upgrade Kit	\$399.00
Super Pet	\$1599.00
2031	\$529.00
8050	\$1299.00
4040	\$969.00
8300 (Letter Quality)	\$1799.00
8023	\$769.00
4022	\$599.00
Pet to IEEE Cable	\$37.00
IEEE to IEEE Cable	\$46.00
Tractor Feed for 8300	\$240.00
8010 Modem	\$229.00

SOFTWARE

COMMODORE	MAGIS	AVALON	HILL
BPI	PROFESSIONAL	SOFTWARE	
VISICORP	CREATIVE	SOFTWARE	

VIC 20



\$259

Commodore Catassette	\$69.00
Disk Drive	\$499.00
VIC Graphic Printer	\$339.00
3K Memory Expander	\$32.00
8K Memory Expander	\$53.00
RS232C Terminal Interface	\$43.00
VIC IEEE-488 Interface	\$86.00
VIC 20 Super Expander	\$53.00
Programmers Reference Guide	\$15.00
Introduction to Computing	\$19.00
Introduction to BASIC Programming	\$19.00
Household Finance	\$27.00
VIC Games	\$19.00
VIC Home Inventory	\$13.00
VIC Rec/Ed II	\$13.00
Terminal	\$13.00
Un Word	\$13.00
Grafix Menagerie	\$11.00
VIC PICS	\$15.00
Ticker Tape	\$13.00
Banner Headliner	\$13.00
RS 232	\$39.00

 — CALL FOR PRICES AND AVAILABILITY OF APPLE PRODUCTS —

east **computer mail order** **west**
800-233-8950 **CALL TOLL FREE** **800-648-3311**

477 East Third Street
 Williamsport, PA 17701
 (717) 327-9575
 Patricio Habla Espanol

HOW TO ORDER:
 Phone orders invited or send check or money order and receive free shipping in the continental United States. PA residents add 6% sales tax. Add 3% for VISA or MC. Equipment subject to price change and availability without notice.

P.O. Box 6689
 State Line, Nevada 89449
 (702) 588-5654
 Franco Habla Espanol

By Loren Wright

PET's Powerful Screen Editing — Are you getting the most from it?

When it comes to BASIC programming, most computers employ what I call a 'teletype' mentality. You type in the line number, followed by the line contents. Then you send all the characters you typed to the computer by pressing the RETURN key. Before you hit RETURN, you can correct errors by deleting back to the error, making the correction, and then retyping the rest of the line. If you have made the mistake of hitting RETURN before you notice the error, the only way to correct it is to start over. The characters you just typed in are still sitting there on the screen, but you can't do anything with them. The computer has forgotten all about what you typed. Sure, there's a copy in your BASIC program, and you can list the line to the screen, but you can't do anything with that either. Some of these computers allow some primitive editing — if you can remember the obscure control codes and are willing to copy characters back into memory. It is usually faster to retype your BASIC line.

The PET BASIC input works quite differently. Instead of keeping track of the stream of characters you're typing, the PET just puts them up on the screen. You can move the cursor anywhere you want, draw pictures, check your disk directory, or list another line. All that counts is where the cursor is when you hit RETURN. If it's on a line that begins with a number, then the system reads the line as a BASIC statement. If the line doesn't begin with a number and the line isn't a valid direct command, then the PET will respond with a ?SYNTAX ERROR.

This system offers a lot of advantages to the PET BASIC programmer. To correct an error in the line you're typing, all you have to do is move the cursor, make the correction, move the cursor back to the end of the line, and continue typing. If you want to correct a line you've already entered into the program, just list it to the screen (if it isn't already there), make the correction, and hit RETURN with the cursor anywhere on the line to enter the new version.

Tricks of the Trade

It seems simple enough, doesn't it? If you have never tried to write a BASIC program on another computer, you probably take it all for granted. There are, however, a few traps you can fall into, and there are a few little tricks you can use to make the system work even better.

1. Clear the screen before you list lines you're going to edit. If you don't, garbage left over from your program run will appear on the same lines as your BASIC lines and those characters will be put into memory when you hit RETURN.
2. The cursor does not have to be at the end of the line when you hit RETURN. As soon as you have completed your change, you can hit RETURN.
3. If you're at the left end of a line and you want to be at the right end, the fastest way to get there is not to go forward, but rather to back up to the end of the previous line and move down one line. If you're at the right end and you want to be at the left, then the opposite holds true.
4. Don't forget the HOME key! If you're at the bottom of the screen, it's much faster to hit HOME than to move the cursor up all those lines.
5. Shifted RETURN is not the same as RETURN! It will move the cursor to the beginning of the next line, but it will not send the line to the PET for processing.
6. If you need to move a line to make room for others, just list it, change the number and hit RETURN. Remember, though, that the old copy is still there at the old number until you delete it or replace it. This technique is also particularly handy when you are writing a program that is very repetitive (e.g., a series of subroutines, where several lines are identical in each routine). Just type the line once, and for each copy, change the line number and hit RETURN.
7. If a listed line exceeds two lines, the overflow is not considered as part of the line when you try to re-enter it. This happens because you used abbreviations for BASIC keywords (like '?' for PRINT) when you originally entered the line. Using the keyword abbreviations is fine, but try to avoid using such long lines.
8. Be careful with BASIC lines that occupy only one screen line. Under some circumstances it is possible to get the next line listed on the screen entered as part of your current line. The cure is to list only one such line at a time.
9. Use the screen as a temporary storage device! This one takes some care. Let's say you have just typed in 30 lines, and you suddenly decide that only eight of them are good. You could delete each unwanted line by typing its number, but it is faster to list the lines you want to save, type NEW and RETURN. Then position the cursor on the first line, hit RETURN, and keep hitting RETURN until all the lines are restored. If any of them scroll off the screen before you re-enter them, they will have to be retyped.

Most of these tricks work fine for direct commands, too. For instance, if you misspell the file name in a LOAD command, just stop the search, move the cursor to the command line, make the change, and hit RETURN.

Programmed Cursor Mode

Another powerful feature of the PET is its character-programmable cursor commands. Cursor moves can be included as special characters in a BASIC string so that when the string is printed, the cursor moves are executed. To get these characters into the string, the PET has something called "programmed cursor mode," where pressing a cursor key causes the appropriate special character to appear on the screen instead of the cursor move itself. The programmer loses control of the cursor while in programmed cursor mode (PCM), and if you don't know what's going on, it's easy to get

PET VET *(Continued)*

frustrated. PCM is entered under only two circumstances:

1. When you type a double quote, you enter PCM; when you type another, you exit. The PET keeps track of the number of quotes in a line, but it can be fooled.
2. When you use the INSERT key, the PET counts the number of times you press it, and for that number of characters it is in PCM. The assumption is that most insertions will be within strings.

Quite often you want to be in PCM when the PET isn't, and *vice versa*. To get in or out, just type a quote and then delete it if you don't need it. The PET only recognizes when you type quotes, not when you delete them! If you've done an insertion, just type spaces for the number of characters you inserted and you will regain control of the cursor. The spaces can then be deleted.

In other instances, things get completely out of hand and you just want to start over. The answer is shift-

RETURN! It will bail you out of PCM and it will preserve the original version of the line you're editing.

It also helps to know what the cursor control characters look like when they're included in strings. This depends both on which model PET you have and on which character set you're in. A few experiments, and perhaps a little crib sheet taped to your PET will help.

Editing Improvements

If you do a lot of BASIC programming even these powerful features may not be enough. Autounumber, renumber, delete functions, and repeating keys are probably the most useful enhancements. List scrolling and programmable function key capability are also useful. These functions are available in a number of commercial ROMs, such as Programmer's Toolkit, Disk-O-Pro, Command-O, POWER, EZAID, and others. Not all offer all of these editing features, but all include other capabilities.

Fat 40, 8000 Series, and VIC

These recent Commodore machines incorporate repeating keys and an ES-

CAPE key to get out of programmed cursor mode. The 8000 series computers have additional special characters for window, delete line, insert line, scrolling and other commands. The VIC has special characters for color commands and its eight programmable function keys.

Commodore's New Computers

With three new computers added to its existing line, Commodore will have an iron in just about every part of the microcomputer fire. The Ultimax (\$149.95) is a color-and-sound computer that hooks up to any home TV set. It will compete very favorably with the Sinclair ZX-81, Mattel Intellivision, and Atari VCS. The Ultimax will support joysticks, paddles, light pens, cartridges, and cassette storage. To achieve such a low price, Commodore has provided only a limited amount of RAM and a flat membrane keyboard.

The Commodore-64 (\$599) is designed to compete with the Atari 800 and Apple II with its full-size keyboard, 64K of memory, function keys, and sophisticated sound capabilities. Also announced was a 16K VIC — the SuperVIC.

MACRO

(LABEL), Y (LABEL,X) LABEL + INDX-1

6502 Assembler/Editor

- APPLE
- ATARI
- PET
- KIM
- SYM

Before you buy that off-brand Assembler/Text Editor, note that EHS is the only company that provides a line of compatible ASM/TED's for the PET/APPLE/ATARI/SYM/KIM and other microcomputers.

When you make the transition from one of these 6502-based microcomputers to another, you no longer have to relearn peculiar Syntax's, pseudo ops, and commands. Not only that, EHS ASM/TED's are the only resident 6502 Macro Assemblers available and they have been available for several years. Thus you can be sure they work. Our ASM/TED's may cost a little more but do the others provide these powerful features: Macros, Conditional Assembly, String Search and Replace, or even up to 31 characters per label? Before you spend your money on that other ASM/TED, write for our free detailed spec sheet.

MACRO ASM/TED

- For APPLE/ATARI/PET/SYM/KIM
- Other than our MAE, no other assembler is as powerful.
- Macros/Conditional Assembly.
- Extensive text editing features
- Long Labels
- Designed for Cassette-based systems.

\$49.95

MAE ASM/TED

- For APPLE/ATARI/PET
- The most powerful ASM/TED
- Macros/Conditional and Interactive Assembly
- Extensive text editing features
- Long Labels
- Control files
- Designed for Disk-based Systems.

\$169.95



EASTERN HOUSE SOFTWARE
3239 Linda Drive
Winston-Salem, N. C. 27106 USA
(Dealer Inquiries Invited)

PHONE ORDERS
(919) 924-2889
(919) 748-8446



.EN .BY .OS .BA .DE .CE

LDA STA LDX STX LDY STY

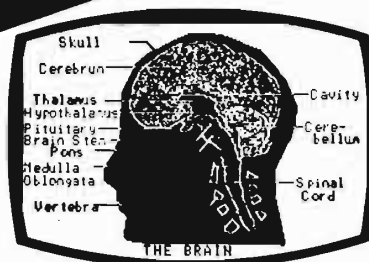
SET IFM IFP IFN IFE

Versa Computing

PRESENTS



VersaWriter DRAWING TABLET



COMPLETE HARDWARE / SOFTWARE GRAPHICS SYSTEM - \$299

- Hi-Res & Med-Res Draw
- Paint Brush-5 Sizes
- Point to Point / Line Draw
- Air Brush
- Color Fill-In
- Change Color Hue & Intensity
- Reverse Picture
- Scaling
- Split / Full Screen
- Save / Load / Erase
- Text Writer
- Fix X or Y Axis

Requires: Atari®300, 32K RAM, Basic Language Cartridge, Disk Drive

GRAPHICS COMPOSER

PADDLE / JOYSTICK GRAPHICS SOFTWARE - \$39.95

- Draw on Hi-Res Screens 7 or 8
- Save Pictures on Disk or Cassette
- Create Player / Missile Shapes Automatically
- Geometric Figures Program
- Add Text to Screen

Requires: Atari®800, 32K RAM, Basic Language Cartridge, Disk or Cassette

GLOBE MASTER

COMPLETE HI-RES GEOGRAPHY GAME - \$29.95

- 8 Hi-Res Color Maps
- U.S.-Europe-World-Asia-Africa-Australia
- Countries-Cities-Capitals-Oceans-Rivers-Mountains, Etc.
- Several Skill Levels

Requires: Atari®800, 32K RAM, Basic Language Cartridge, Disk

ATARI® is a registered trademark of Atari Inc.

MIND BOGGLERS

THREE STRATEGY GAMES

- Capture
- Mystery Box
- Simon Says

CASSETTE \$15.95 DISK \$19.95

Requires: Atari®400, 16K RAM, Cassette
Atari 800, 24K RAM, Cassette or Disk



Versa Computing, Inc.

3541 Old Conejo Road, Suite 104
Newbury Park, CA. 91320 (805)498-1956

ROCKWELL Microcomputers from Excert, Inc.

THE AIM 65/40 Single Board or Smorgasbord



- A full size terminal style keyboard w/8 special function keys
- A smart, 40 character display with its own microprocessor
- A 40 column printer w/text and graphic output
- Up to 64K of on-board RAM and ROM
- On-board interfaces include RS232, dual audio cassette and 2 user I/O R6522 devices
- Firmware includes interactive monitor and text editor w/options of Assembler, BASIC, FORTH and PL/65

THE AIM 65 Take-Out Order



- A full size terminal style keyboard w/3 special function keys
- A 20 character display
- A 20 column printer w/text and graphic output capability
- Up to 4K RAM and 20K ROM on-board
- On-board interfaces include 20MA TTY, dual audio cassette and 1 user I/O R6522 device
- Firmware includes interactive monitor and text editor w/options of Assembler, BASIC, FORTH, PASCAL, & PL/65

And if the above isn't enough,
Try the RM65 — a product line filled with embellishments including:

32K DRAM Board
CRT Controller
Floppy Disk Controller
PROM Programmer

ACIA Board
IEEE-488Board
CPU/SBC Board
4-16 Slot Card Cages

Prototype cards
Adaptor Buffer Modules
General Purpose I/O Board
PROM/ROM Board

NEW LOWER PRICES AND A CASH DISCOUNT* TO BOOT!

A65/40-16 (16K RAM)	\$1225
A65/40-32 (32K RAM)	\$1295
A65/40-A (Assembler)	\$ 85
A65/40-B (BASIC)	\$ 65

A65-1 (1K RAM)	\$420
A65-4 (4K RAM)	\$445
A65-4B (4K RAM w/BASIC)	\$495
A65-PS (PASCAL)	\$100
A65-F (FORTH)	\$ 65
A65-A (Assembler)	\$ 35

Mail Order to:

Educational Computer Division EXCERT INCORPORATED

- SALES
- SERVICE
- INSTALLATION
- CONSULTING

P.O. Box 8600
White Bear Lake, MN 55110
(612) 426-4114

Higher quantities quoted upon request, COD's accepted, shipping will be added. *Deduct 5% cash discount on prepaid orders. Minnesota residents add 5% sales tax. Prices subject to change without notice.

From Here to Atari

By James Capparell

Editor's Note: This is the last "From Here to Atari" that will appear in MICRO. We thank Jim Capparell for his efforts and wish him success with his new Atari magazine, ANTIC. And we say to the Atari Community: send us your work! We are very interested in publishing Atari material.

The "front jacks" on the Atari 400 and 800 are used for everything from game controller ports to printer and graphics ports. To get the most out of them it is necessary to understand something about the hardware. This month, I'll provide a description of the pin configuration of these jacks, the memory locations assigned to the jacks, and an example of how each pin may be individually configured as input or output.

The hardware controlling these jacks is a 6820 PIA (Peripheral Interface Adapter). This chip consists of two ports labeled "A" and "B". Port A controls Jacks 1 and 2. Jacks 3 and 4 (on your right) are controlled by Port B. The jacks are numbered left to right as you face the console. There is a numbering discrepancy: BASIC numbers them 0 to 3. The BASIC statement,

```
10?STICK(0),STICK(1),STICK(2),STICK(3)
```

will print the values read at Jack 1, Jack 2, Jack 3, and Jack 4.

Each port consists of three registers — the Control Register, the Data Direction Register (DDR), and the Data (buffer) Register (DR). The PIA is a 40-pin chip. Due to a limitation on available pins, the DDR and the DR share the same address. (See table 1.)

Bit 2 of the control registers determines whether the DDR or the DR is addressed. When set to 0, bit Z addresses the DDR, but when set to 1 the DR is addressed. The data register simply holds data. When the jacks are configured as input ports, the DR holds the data for the Atari to read.

When the jacks are configured as output ports, the DR holds data to be written to an external device. The DDR determines for the PIA which lines are input and which are output.

To configure Jack 1 as input and Jack 2 as output, it is necessary to tell the PIA the direction for each of the eight bits in Port A. To accomplish this, perform the following steps:

1. Set bit 2 of PACTL(\$D302) to 0. This allows us to address the DDR.
2. Write 00001111 = 15 to address \$D300 (note a 1 bit indicates the associated line is output).
3. Set bit 2 of PACTL to 1. This restores address \$D300 to the data register.

At this point, Jack 1 can be read normally with a STICK[0] statement. Jack 2 can't be read since it is configured as an output jack. Try the following:

```
10 POKE 54018,0 :REM Go talk to
Data Direction Register
20 POKE 54016,15 :REM Jack 1 is input,
Jack 2 is output
30 POKE 54018,4 :REM Reset to data
register
40 REM connect joystick to Jacks 1 and 2
50 ?STICK(0),STICK(1) :REM Print out
values from Jacks 1 and 2
60 GOTO 50 :REM Loop forever
70 REM Move Joysticks 1 and 2, only
Joystick 1 will register a change.
```

Whenever your system is turned on all jacks are configured as input. That is, the operating system writes a 0 to the Data Direction Registers in Ports A and B. The values returned at these jacks are always a 1 when there is no input — logical 1 is false. This helps explain why a 15 is read even when there is no

Table 1	
I/O Address	O.S. Shadow Address
\$D300 {54016}	\$278 {632}
Port A data register or data direction register when bit 2 of PACTL is 0. This address corresponds with Jack 1 and Jack 2. BASIC statements STICK[0] and STICK[1] read this port.	
\$D301 {54017}	\$279 {633}
Port B data register or data direction register when bit 2 of PBCTL is 0. This address corresponds with Jack 3 and Jack 4. BASIC statements STICK[2] and STICK[3] read this port.	
\$D302 {54018}	\$27A {634}
Port A control register. Insert a value of 4 (bit 2 = 1) and \$D300 becomes the Data Register.	
\$D303 {54019}	\$27B {635}
Port B control register. Insert a value of 4 (bit 2 = 1) and \$D301 becomes the Data Register.	
The shadow registers are updated at Stage 2 of Vertical blank processing — no more frequently than every 1/60 second. If your program requires more accurate data, read the associated hardware registers at addresses \$D300 and \$D301.	

From Here To Atari *(Continued)*

input from a joystick. Look at the diagram in figure 1 for correspondences between bits in DDR and bits in data buffer.

When a jack is configured to input and the following BASIC statement is executed:

```
10 ?STICK(0):GOTO 10
```

the following values will be printed as the joystick is manipulated:

```
1111 (15) = stick neutral
1110 (14) = forward
1101 (13) = backward
1011 (12) = left
0111 (11) = right
```

Combinations (diagonal)

```
1010 (10) = forward/left
1001 (9) = backward/left
0101 (7) = backward/right
0110 (6) = forward/right
```

The pin configuration for each jack is as follows:

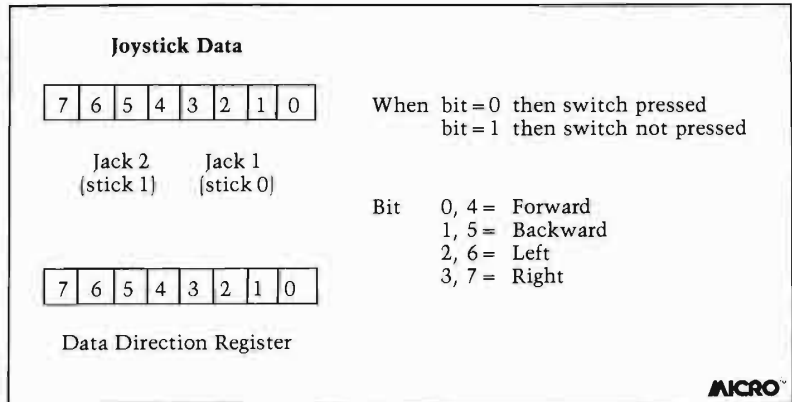
1	2	3	4	5
6	7	8	9	

Console (male)

Pin 1 = forward
Pin 2 = backward
Pin 3 = left
Pin 4 = right

Pin 5 = pot (paddle control)
Pin 6 = joystick trigger at \$D010-\$D013 (CTIA)
Pin 7 = +5V
Pin 8 = gnd
Pin 9 = pot (paddle control)

These front jacks are versatile and easy to use. I've connected a Hewlett Packard Bar code reader to my 800. Others have used them for graphics printer interface and 10-key pad for business use as well.



BREAK THE TIME BARRIER!

Speed-up and expand Apple II functions
with *TWO* all-new, low-cost utilities. . . *

1. UNIVERSAL BOOT INITIALIZER (UBI) \$49.95

• Initializes DOS 3.3 or 3.2.1 disks that load RAM Card on boot in 1.7 seconds. • UBI-initialized disks will RUN (BRUN or EXEC) one of two GREETING programs supplied by the user. • Built-in error handling capacity ensures execution of correct GREETING program whether one BASIC or both BASICs are available. • Obsoletes use of Apple's BASICS disk and DOS 3.3 System Master disk for fast single-disk direct booting. • The DOS 3.2.1 UBI-initialized disk boots on either 13 or 16 sector PROMS. • Multiple copies of UBI-initialized disks can be made with standard copy programs.

2. AMPER-SORT/MERGE (A-S/M) \$49.95

• Can sort 1000 records in seconds. • General purpose Disk Sort/Merge Utility for Apple DOS 3.3 text files. • M/L Read, Sort and Merge for speed. • Sorts up to 5 input files into 1 output file. • Sorts random or sequential files alphanumerically (Ascending or Descending) up to 5 fields. • Merges 2 to 5 pre-sorted files into a single file. • Compatible with most data base programs that create standard DOS 3.3 text files [e.g., VisiCorp's VisiFiles (formerly Personal Software's CCA DMS files)].

SYSTEM

REQUIREMENTS: 48K Apple II or II+, ROM or RAM Card, DOS 3.3 (or DOS 3.2.1 for UBI) and one or more disk drives.

COST: \$49.95 (+\$3.00 Postage and Handling) for each Utility Disk (with Training Disk and Manual).

FOR ONLY \$15: You can get the Training Disk and Instruction Manual for either utility.

CREDIT CARDS: [VISA, MC or AMER EXP] Call **cybertronics international, inc.**
Clearinghouse: (212) 532-3089.

MAIL ORDER: Send check (no Purchase Orders or C.O.D.'s) to S&H Software, Box 5, Manvel, ND 58256.

***LICENSED FROM APPLE:** All Apple Computer Inc. programs — FPBASIC, INTBASIC, DOS 3.3 & DOS 3.2.1 — used in S & H Software's utility programs are licensed from Apple Computer Inc. by S & H Software. The unique UBI license to software vendors allows their use of these Apple Computer copyrighted programs to execute their programs on UBI-created disks, without further licensing from Apple Computer Inc. Software vendors' inquiries invited for both utilities.

S&H Software



Box 5 Manvel ND 58256
(701) 696-2574



THE BUG

A sophisticated, yet easy to use diagnostic aid for getting "the bugs" out of your assembly language programs.

If you are a novice just getting started with assembly-language programming, you will find The BUG helpful in developing your understanding of how the Apple's 6502 internal processor operates. The many display options of The BUG will permit you to try out your assembly-language programs at the speed that is most comfortable for you. The BUG will also make it easy for you to see the effect of your program on the Apple as it executes.

If you are a professional programmer, you will also find that The BUG can improve your efficiency by reducing the time you spend identifying and solving complex, assembly-language programming errors. You will particularly appreciate the fact that The BUG offers the easiest to use and most extensive breakpointing capability of any "debugger" available for the Apple. Up to 13 different breakpoints can be specified to halt program execution when either: 1) a particular program location is reached, 2) one of the 6502 registers reaches a specified value, or 3) one of the bits in the 6502 status register reaches a specified value.

Another key feature of The BUG that serious programmers will appreciate is the ability to AUTOMATICALLY run lower-level subroutines at FULL SPEED. You no longer have to keep debugging the portions of your program that you already have working.

This is not the least expensive "debugger" program for the Apple, but we challenge you to find more capability for less money!

The BUG is supplied with a 40+ page user guide and is designed for use with DOS 3.3 on either the Apple II or Apple II Plus computer. **only \$50.00**

BUILD USING

... provides an easy to use print-using routine plus similar functions for strings. Creating charts, reports and general screen formatting becomes a simple task. BUILD USING is written entirely in machine language and provides a simple means of avoiding garbage collection (those unnecessary delays that slow down your programs). With BUILD USING, you can choose how many digits should be displayed to right and left of the decimal point, and you can even fill the leading positions with the character of your choice. For example, you can print the number '157.23' as '157.2', or '0000157.230', or '***** \$157. AND 23/100 DOLLARS', or hundreds of other ways (including exponential formats). Working with strings is just as easy; it's a snap to convert names from 'John' and 'Doe' to 'Doe, J.'. Also included are three levels of error trapping, so you can trap and correct numbers or strings that cannot fit in your specified format.

Utilities like BUILD USING are usually difficult to use because they must be located in one memory location (usually between DOS and the DOS file buffers); they cannot be used with your favorite editor or other special routines. BUILD USING does not have this limitation, as it can be easily located in many different memory locations: 1) the "normal" between DOS and DOS file buffers, 2) at HIMEM, 3) APPENDED to your Applesoft program, or 4) anywhere else in memory. Appending BUILD USING to your program is as simple as EXECing a TEXT file. BUILD USING uses the "CALL" command thereby leaving the ampersand vector free for your own use.

BUILD USING requires Applesoft in ROM (Language cards are fine), DOS 3.3 and a minimum of 32K. **only \$30.00**

IMAGE PRINTER SERIES

Sensible Software is proud to introduce our new series of high resolution screen dumps. IMAGE PRINTERS provide a simple way to transfer high resolution graphic images onto paper. Each program in the series has unique features that give you full control of the printing. Some of the included options are:

- Full control over the area of the HIRES screen to be printed. You graphically pick the area for the utmost ease and accuracy.
 - One-step printout of the picture with the ability to pause or abort the printing at any time.
 - Menu-driven. All options are invoked with single keystrokes. IMAGE PRINTERS are extremely easy to use.
 - Multiple image sizes, 6 different sizes for letter quality printers, 4 sizes for other printers.
 - Creation of an inverse (negative) image for reverse printing.
 - The ability to save the compressed and inverse images to disk.
 - One time configuring for your printer and interface card.
- Why answer all those questions about your printer each time you want to print a picture?

- The images may be printed anywhere on the page.
- IMAGE PRINTERS support most popular interface cards, such as cards from Apple, California Computer Systems, Epson, and Mountain Computer. (The SSM AIO Serial Card and user-written 'driver' routines may be used with the letter quality printers.)

There are three separate versions of IMAGE PRINTERS, each one tailored to take full advantage of a different printer.

IMAGE PRINTER—LETTER QUALITY. For all popular letter quality printers (Diablo, NEC, Qume, etc.)

IMAGE PRINTER—EPSON. For the popular Epson MX-70, MX-80 and MX-100.

IMAGE PRINTER—NEC PC-8023A. For the NEC dot-matrix printer.

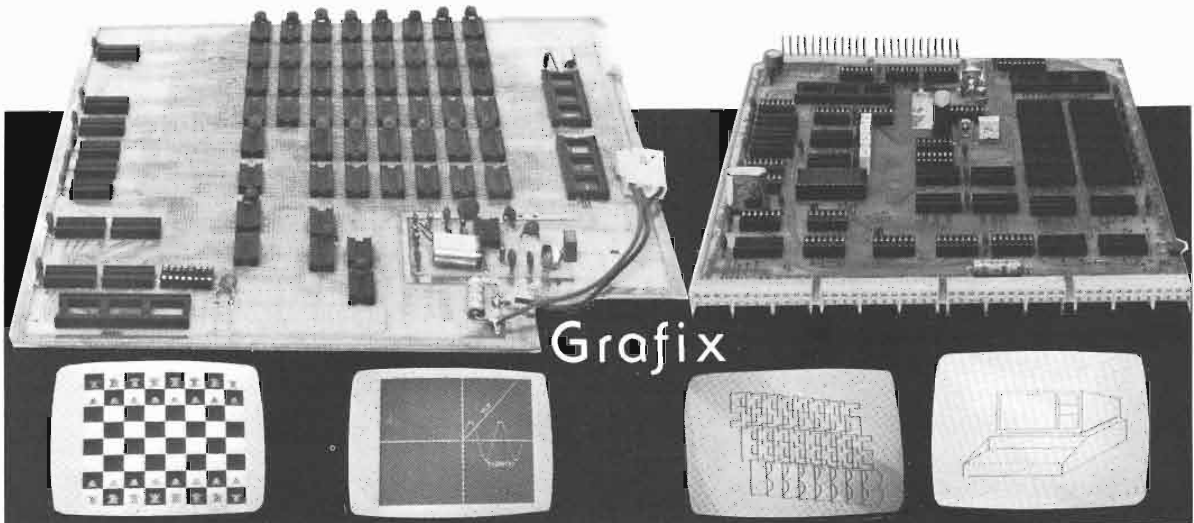
All versions are available for \$40.00 ea.

Please specify version desired.



Sensible Software

6619 Perham Drive Dept. M
West Bloomfield, Michigan
48033 • (313) 399-8877
Visa and Mastercard
Welcome
Please add \$1.25
postage and
handling per
diskette



SUPER OSI EXPANSION BOARDS

If your Challenger can't generate displays like those shown above **WHAT ARE YOU WAITING FOR?** The SEB-1 High Resolution Graphics and Memory Board (for C1P and Superboard II) and the SEB-2 High Resolution Graphics and Disk Controller Board (for C2/4/8) simply 'plug-in' to your computer and give you instant access to over 49000 individually addressable pixels in up to 8 colors! Your Hi-Res screen can go from 32 x 16 alphanumeric to 256 x 192 point graphics in 11 software selectable modes. The standard video of your computer is left intact, so that none of your current software library is outmoded. Use the graphics for Business, Scientific, Education, or Gaming displays that were impossible — until now!

Installation of either board requires absolutely NO modification of your computer—they just 'plug-in'. Nor do they preclude you using any other OSI-compatible hardware or software. In addition to the Hi-Res Graphics the SEB-1 gives C1 & Superboard II users 16K of additional user memory (over and above that memory devoted to the graphics), two 16 bit timers/counters, an on-board RF modulator, and a parallel port with handshaking. The SEB-2 gives OSI 48-pin BUS users an OSI hardware/software compatible Disk controller, and an RF modulator that can be user-populated.

FOR OSI 1P, 2-4P, 2-8P, C4P, C8P

	SEB-1	SEB-2
Assembled and Tested	\$249.00 (5K RAM)	\$239.00 (1K RAM)
Kit	\$165.00 (No RAM)	\$199.00 (No RAM)

	SEB-1	SEB-2
Bare Board & Manual	\$ 59.00	\$ 59.00
Manual only	\$ 5.00	\$ 5.00

SOFTWARE

TARG-IT! \$11.95

An action-packed, super fast arcade game where you try to destroy Ram ships and the Evil Spectre inside the Crystal city. 8K.

PLANETARY LANDER \$11.95

Your computer generates contour maps of the surface of a strange planet as you descend looking for the landing pad. As you make your final approach, the computer displays side and top views of the terrain and your ship as you try to land while avoiding the swarms of asteroids above the planet's surface. 8K.

PROGRAMMA-TANK \$9.95

Programma-Tank is a complex simulation of a battle between two robot tanks as well as a fairly painless way to learn assembly language. The opponents program their tanks in 'Smalltank' a miniature programming language, which is much like assembly language. The two strategies are then pitted against one another until a victor emerges. 8K.

RUINS OF TALAMARINE \$16.95

This is the first of our "Epic Quests" programs. Epic Quests could be described as graphic adventures, but they are much more. Can you retrieve the stolen Jewels from the ruined Talamarine castle? This series of two programs allows you to create a character with the qualities that you want to do battle with the forces of darkness. 8K.

CODENAME: BARRACUDA \$19.95

Another of our "Epic Quests", in which you are an agent working for the US Government trying to rescue a defecting Soviet professor. The problem is that a middle-eastern government has kidnapped the professor and is forcing him to develop atomic weapons for them. The three programs (8K ea.) in this series allow you to create, train and send your character on his mission.

DUNGEONS OF ZORXON \$19.95

Can you explore the Dungeons of Zorxon in this "Epic Quest"? Step into a nether world of monstrous beasts and fortunes in gold where magic reigns supreme. Three programs 8K each.

COMING: SEB-3 80 x24 Video/Disk Controller (C2/4/8), SEB-4 48K Memory RAM/ROM (C2/4/8), SEB-5 8K RAM/Disk/Sound/Clock/Voice (C1 & Superboard).

Write for FREE catalog
International Requests please
supply 2 International Response Coupons

ORION



SOFTWARE ASSO.

147 MAIN ST, PO BOX 310, OSSINING, NY 10562



Reviews in Brief

Product Name: Mittendorf High-Resolution Graphics Board

Equip. req'd: OSI
Price: \$40 bare board; \$185 kit
Manufacturer: Mittendorf Engineering
905 Villa Nueva Dr.
Litchfield Park, AZ 85340

Description: A 6" x 6" circuit board which adds 256 x 256 black and white high-resolution graphics to OSI systems. The same kit works with the superbard or 540 video board. The Mittendorf board contains 8K of 2114 memory which can be used for program storage when not using graphics.

Pluses: Combines with the present video signal to give hi-res graphics and the OSI character set on the same monitor. The graphics memory can be wired at one of several addresses.

Minuses: The Mittendorf board requires 16 connections into the OSI video circuits. The 540 version requires additional jumpers to all the bus lines. The superbard version requires removal of the data buffers which prevents further use of the 40-pin expansion port. All 256 x 256 dots are not visible. Dots are lost to overscan in the same ratio that are lost from OSI's nominal "32 x 32" characters.

Documentation: Several sets of construction plans dependent on system, software to add graphics commands to BASIC, demonstration examples.

Skill level required: Experienced builder; modification of present video circuits required.

Reviewer: Earl D. Morris

Product Name: Visiterm
Equip. req'd: Apple II or Apple II Plus and communications device: Apple Communications card, CCS card plus modem or D.C. Hayes Micromodem
Price: \$149.95
Manufacturer: Personal Software Inc.
1330 Bordeaux Drive
Sunnyvale, CA 94086
(408) 745-7841
Author: Tom Keith
Copy Protection: Yes
Language: 6502 Machine Language

Description: A communication package for interfacing the Apple with other computers, permitting the transfer and receipt of sequential text files.

Pluses: Visiterm uses one of two high-resolution character sets and the hi-res screen to provide the Apple user with a seventy-column display when communicating with other systems. This feature is particularly valuable when accessing a mainframe computer since up to eighty columns are often transmitted to the user. The character sets are programmable, permitting the user the valuable resource of redefining keys to permit the generation of mainframe-dependent control characters. One of the most technically

challenging aspects of using a micro as an intelligent terminal is the problem of required control keys such as the Break and X-on and X-off signals. The very extensive manual contains a thorough discussion of data communications. The main part of the manual contains almost 100 pages. The appendices, glossary, and detailed index contain almost as many pages. A print utility is provided that allows the user to obtain a hard copy of received data. The utility has many useful options which permit printer control and output formatting. A very powerful package with features useful for more than just communications.

Minuses: One limitation is that only sequential text files can be exchanged. However, stand-alone utility programs, outside of Visiterm, are provided to convert BASIC and binary files to and from text files. Visiterm does not provide the user with the ability to edit the data buffer exchanged. A separate text file editor may be needed by the user. [This is mentioned because at least one competing package does provide this feature.] An abbreviated summary of the Visiterm options would be helpful. For those familiar with VisiCalc II, also manufactured by Personal Software, a flow chart of commands is included which provides the user an excellent reference. It is sometimes difficult to locate the discussion of a particular topic. For example, it is possible to obtain a CATALOG of text files on a disk from within Visiterm. However, the section describing this is found under lesson Three, "File Transfer Mode." You may find it necessary to read most of the three lessons provided before you are comfortable using the package.

Skill level required: For the moderately sophisticated user, preferably with some understanding of communications problems.

Reviewer: David R. Morganstein

Product Name: A2-3D1, A2-3D2 and A2-GE1 Graphics Package
Equip. req'd: 48K Apple II or Apple II with disk drive
Price: \$119.85
Manufacturer: SubLogic Communications Corp.
713 Edgebrook Drive
Champaign, IL

Description: Programs to help the user produce, project, and maneuver three-dimensional shapes on the two-dimensional screen medium. Contains impressive features for recording motion sequences and replaying them. Individual snapshots or slides of a motion sequence can also be recorded for later display. Provision is made for interfacing routines to Applesoft programs.

Pluses: Either eye or object movement can be commanded, thereby adding flexibility to sequence definition. Exceptional ease in interfacing to BASIC programs.

Minuses: Extensive memory and disk space is required. The included demo disk inadequately demonstrates the

(Continued on next page)

Reviews in Brief *(Continued)*

package's considerable capabilities. The potential buyer should be aware of this and not underjudge the product.

Documentation: Superb documentation leads the user through a continuing example that eventually opens a 3-D garage door. Along the way, all capabilities are presented and an example of each is given. Surprisingly well-written in a style that lends itself to use as a tutorial or a reference.

Skill level required: Competent BASIC programmer with some exposure to assembly language.

Reviewer: Chris Williams

Product Name: **56K CMOS Static Memory Board**
Equip. req'd: OSI 48-pin bus
Price: 4K \$200; 24K \$450; 56K \$850
Manufacturer: Micro-Interface
3111 So. Valley View Blvd.
Suite I-101
Las Vegas, Nevada 89102

Description: The Micro-Interface board puts 56K of memory, an expanded monitor ROM and a parallel printer port all into a single bus slot. The board enable can be set at each 2K address selection, allowing any combination of 6116 CMOS RAM and/or 2716 EPROM to populate any portion of the 56K memory space. The use of CMOS RAM reduces the power requirements for 48K to less than ½ amp, allowing memory expansion without a new power supply. Decoding is also provided for a 1.75K enhanced ROM monitor between \$F800 and \$FFFF. Micro-Interface sells several such monitors, or you can program your own into a 2716 EPROM.

Pluses: Very low power RAM rated for 2 MHz operation. Combines functions of several boards into one bus slot. Provision is made for multi-user or memory banking. The parallel port supports either a 6821 PIA or 6522 VIA. The board is available assembled with any amount of memory between 4K and 56K. Additional memory chips are easily installed.

Minuses: For 8/16/24K the Micro-Interface board is more expensive than the same memory assembled from D&N.

Documentation: Instructions for installing jumpers, memory addresses, chip types, jumper locations, and functions are printed on the circuit board.

Reviewer: Earl Morris

Product Name: **Cer-Comp Color Computer Editor**
Equip. req'd: TRS-80C Color Computer
with 16K memory
Price: \$19.95
Manufacturer: Cer-Comp
5566 Ricochet Ave.
Las Vegas, Nevada

Description: A screen editor based on line numbers; resides in R/W memory, distributed on cassette tape using the Color Computer format. The editor has 21 commands that modify text produced in a BASIC-like format. Two edit modes allow spaces or characters to be inserted or deleted from existing lines, and allow forward and reverse scrolling through existing text. Cursor control is either single space per keystroke (forward or back) or single keystroke to reach either end of a line. Block move and copy, search and replace, list to screen or printer with or without line numbers, load and save tapes, append a second tape to existing text, and some special commands for

BASIC files are available. In addition, line numbers can be removed from a file to save space, or added to files from other editors to allow editing.

Pluses: Low price, good versatility, easy to learn, does not require Extended BASIC. Works with machine-language monitors.

Minuses: Instructions and documentation lacking, no listing supplied. Although cursor control is adequate, a repeat key function for continuous cursor scroll would be advantageous.

Skill level required: Normal typing skills, ability to visualize final page format.

Reviewer: Ralph Tenny

Product Name: **Color Computer Disk System**
Equip. req'd: TRS-80 Color Computer,
16K w/Extended BASIC
Price: \$600
Manufacturer: Tandy Radio Shack
P.O. Box 2625
Fort Worth, TX 76113

Description: A 35-track, double-density disk operating system for the Color Computer. Capacity is 156,672 user-available bytes, and 68 maximum files, on a standard 5¼ inch soft-sectored diskette. The system includes a single drive, a disk controller ROM pak, and a connecting cable that allows two drives at a time on line. A four drive cable is optional. System utilities include BACKUP, COPY, and FORMAT. The operating system requires 2K of RAM and no disk space (except for directory tracks). Files are cataloged with an eight-character file name, and three-letter extension. VERIFY, LSET, RSET, MKNS\$, and CVN\$ are typical commands available to the system which are used in other DOS systems.

Pluses: Because the operating system is on ROM, it requires very little extra memory from the machine. There is no DOS to learn, as disk commands are an extension of BASIC. As there is no DOS with the COLOR disk system, all disk commands can be executed from BASIC, inside or outside a program. The Microsoft disk BASIC includes the "WRITE" command, which allows easier formatting and creation of serial data files, and random access variable length files. The disk BASIC is simple, and easy to learn.

Minuses: Utilities are lacking in sophistication, compared to TRSDOS. Backups require pre-formatted destination disks, and there are no file protection capabilities (other than the write protect tab). BACKUP also copies all bytes on a disk, whether there is one small file, or a full disk. Auto start is not supported, and there are no DO files to provide a turn-key system. This could be partially offset by running a standard file upon power-up. This file could load any machine-language routines, and finally load the desired program from a MENU. The CHAIN command is not supported, although it is possible to load and run a program from inside another program. Another useful command that is missing is the ON ERROR GOTO statement.

Documentation: The owner's manual is written to the same high standards of the other two Color Computer manuals. Instructions pre-suppose no previous experience with disk systems or programming. The style is very readable, and some fine demonstration file programs are included. Missing is the usual TRS-80 programmers card.

Skill level required: Novice.

Reviewer: John Steiner

MICRO

OSI

TRS-80

COLOR-80

OSI

GALAXIAN - 4K - One of the fastest and finest arcade games ever written for the OSI, this one features rows of hard-hitting evasive dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. Specify system - A bargain at \$9.95 OSI

LABYRINTH - 8K - This has a display background similar to MINOS as the action takes place in a realistic maze seen from ground level. This is, however, a real time monster hunt as you track down and shoot mobile monsters on foot. Checking out and testing this one was the most fun I've had in years! - \$13.95. OSI

THE AARDVARK JOURNAL

FOR OSI USERS - This is a bi-monthly tutorial journal running only articles about OSI systems. Every issue contains programs customized for OSI, tutorials on how to use and modify the system, and reviews of OSI related products. In the last two years we have run articles like these!

- 1) A tutorial on Machine Code for BASIC programmers.
- 2) Complete listings of two word processors for BASIC IN ROM machines.
- 3) Moving the Directory off track 12.
- 4) Listings for 20 game programs for the OSI.
- 5) How to write high speed BASIC - and lots more -

Vol. 1 (1980) 6 back issues - \$9.00

Vol. 2 (1981) 4 back issues and subscription for 2 additional issues - \$9.00.

ADVENTURES!!!

For OSI, TRS-80, and COLOR-80. These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are interactive fantasies. It's like reading a book except that you are the main character as you give the computer commands like "Look in the Coffin" and "Light the torch".)

Adventures require 8K on an OSI and 16K on COLOR-80 and TRS-80. They sell for \$14.95 each.

ESCAPE FROM MARS (by Rodger Olsen)

This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

PYRAMID (by Rodger Olsen)

This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

TREK ADVENTURE (by Bob Retelle)

This one takes place aboard a familiar starship. The crew has left for good reasons - but they forgot to take you, and now you are in deep trouble.

DEATH SHIP (by Rodger Olsen)

Our first and original ADVENTURE, this one takes place aboard a cruise ship - but it ain't the Love Boat.

VAMPIRE CASTLE (by Mike Bassman)

This is a contest between you and old Drac - and it's getting a little dark outside. \$14.95 each.

OSI NEW-NEW-NEW TINY COMPILER

The easy way to speed in your programs. The tiny compiler lets you write and debug your program in Basic and then automatically compiles a Machine Code version that runs from 50-150 times faster. The tiny compiler generates relocatable, native, transportable machine code that can be run on any 6502 system.

It does have some limitations. It is memory hungry - 8K is the minimum sized system that can run the Compiler. It also handles only a limited subset of Basic - about 20 keywords including FOR, NEXT, IF THEN, GOSUB, GOTO, RETURN, END, STOP, USR(X), PEEK, POKE, =, *, /, <, >. Variable names A-Z, and Integer Numbers from 0-64K.

TINY COMPILER is written in Basic. It can be modified and augmented by the user. It comes with a 20 page manual.

TINY COMPILER - \$19.95 on tape or disk OSI

SUPERDISK II

This disk contains a new BEXEC* that boots up with a numbered directory and which allows creation, deletion and renaming of files without calling other programs. It also contains a slight modification to BASIC to allow 14 character file names.

The disk contains a disk manager that contains a disk packer, a hex/dec calculator and several other utilities.

It also has a full screen editor (in machine code on C2P/C4) that makes corrections a snap. We'll also toss in renumbering and program search programs - and sell the whole thing for - SUPERDISK II \$29.95 (5 1/4") OSI

BARE BOARDS FOR OSI C1P

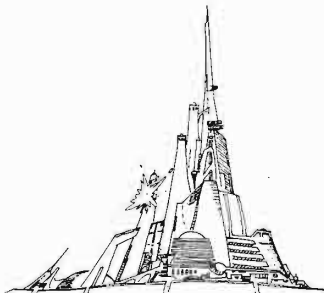
MEMORY BOARDS!!! - for the C1P - and they contain parallel ports!

Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel port! It sells as a bare board for \$29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P - Burns single supply 2716's. Bare board - \$24.95.

MOTHER BOARD - Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - \$14.95.

16K RAM BOARD FOR C1P - This one does not have a parallel port, but it does support 16K of 2114's. Bare Board \$39.95.



WORD PROCESSING THE EASY WAY - WITH MAXI-PROS

This is a line-oriented word processor designed for the office that doesn't want to send every new girl out for training in how to type a letter.

It has automatic right and left margin justification and lets you vary the width and margins during printing. It has automatic pagination and automatic page numbering. It will print any text single, double or triple spaced and has text centering commands. It will make any number of multiple copies or chain files together to print an entire disk of data at one time.

MAXI-PROS has both global and line edit capability and the polled keyboard versions contain a corrected keyboard routine that make the OSI keyboard decode as a standard typewriter keyboard.

MAXI-PROS also has sophisticated file capabilities. It can access a file for names and addresses, stop for inputs, and print form letters. It has file merging capabilities so that it can store and combine paragraphs and pages in any order.

Best of all, it is in BASIC (OS65D 51/4" or 8" disk) so that it can be easily adapted to any printer or printing job and so that it can be sold for a measly price.

MAXI-PROS - \$39.95. Specify 5 1/4" or 8" disk.

SUPPORT ROMS FOR BASIC IN ROM MACHINES - C1S/C2S. This ROM adds line edit functions, software selectable scroll windows, bell support, choice of OSI or standard keyboard routines, two callable screen clears, and software support for 32-64 characters per line video. Has one character command to switch model 2 C1P from 24 to 48 character line. When installed in C2 or C4 (C2S) requires installation of additional chip. C1P requires only a jumper change. - \$39.95

C1E/C2E similar to above but with extended machine code monitor. - \$59.95 OSI

ARCADe GAMES FOR OSI, COLOR-80 AND TRS-80 (8K OSI, 16K TRS-80 AND COLOR-80)

TIMETREK - A REAL TIME, REAL GRAPHICS STARTRECK. See your torpedoes hit and watch your instruments work in real time. No more unrealistic scrolling displays! \$14.95.

STARFIGHTER - This one man space war game pits you against spacecruisers, battlewagons, and one man fighters, you have the view from your cockpit window, a real time working instrument panel, and your wits. Another real time goody. \$9.95

BATTLEFLEET - This grown up version of Battleship is the toughest thinking game available on OSI or 80 computers. There is no luck involved as you seek out the computers hidden fleet. A topographical toughie. \$9.95

QUEST - A NEW IDEA IN ADVENTURE GAMES! Different from all the others, Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. 16K COLOR-80 OR TRS-80 ONLY. \$14.95

Please specify system on all orders

This is only a partial listing of what we have to offer. We offer over 120 games, ROMS, and data sheets for OSI systems and many games and utilities for COLOR-80 and TRS-80. Send \$1.00 for our catalog.

AARDVARK TECHNICAL SERVICES, LTD.
2352 S. Commerce, Walled Lake, MI 48088
(313) 669-3110



OSI



COLOR-80

Adventure of the Month



The Adventure is Waiting for You.....

How would you like to go back in time to 19th century London to match wits with Jack the Ripper? Out into space to brave the swirling vortex of a black hole? Into the depths of the ocean, or on a quest to rescue a beautiful princess from the clutches of evil monsters?

You never know where **SoftSide Magazine's Adventure of the Month** might take you. But you can be sure that each month you will experience new delights and new challenges as you receive an original adventure on tape or disk, ready to load into your computer.

The cost? A six-month membership is only \$29 for the adventures on tape (\$4.83 each) or \$49 on disk (\$8.16 each). If you're not sure you can handle six full months of excitement, you can order a single adventure on tape for \$7 or on disk for \$10. You can choose from:

Arabian Adventure	Black Hole Adventure
Alien Adventure	Windsloe Mansion Adventure
Treasure Island Adventure	Klondike Adventure
Jack the Ripper Adventure	James Brand Adventure
Crime Adventure	Witches' Brew Adventure
Around the World in 80 Days	

To order, use coupon provided or write to:

Adventure of the Month
6 South Street
Milford, NH 03055

ADVENTURE OF THE MONTH

■ **Six month subscription:**

- Cassette — \$29
 Disk — \$49

■ **Individual adventures (please specify)**

- Cassette — \$7 each
 Disk — \$10 each

■ **Three adventures on one super disk (\$26 each):**

- Arabian, Alien, and Treasure Island Adventures
 Jack the Ripper, Crime, and Around the World Adventures
 Black Hole, Windsloe Mansion, and Klondike Adventures

Please specify which computer:

- Apple™ (req. 24K for tape, 32K for disk)
 ATARI® (req. 32K for tape, 40K for disk)
 TRS-80® (req. 16K for tape, 32K for disk)

Name _____

Address _____

City/State _____ Zip _____

MasterCard VISA Payment enclosed

Name of Cardholder _____

MC# and Interbank#/VISA# _____

Exp. Date _____ Signature _____

Prices subject to change without notice. Apple™, ATARI® and TRS-80® are registered trademarks of The Apple Computer Company, Warner Communications and The Tandy Corporation respectively.

LISZT with Strings

by Leonard H. Anderson, Donald Cohen, Richard F. Searle

LISZT turns your Applesoft program listing into an easy-to-understand structured format. The program is designed to be flexible and works with a variety of printers.

LISZT

requires:

Apple II with Applesoft
Disk Drive
Printer

Can you understand a program you wrote six months ago? Do you remember some of those special tricks imbedded in a concatenated line of code? The "LISZT" (Logical Interpreter Statement Zeugmatic Tabulator) can help you understand BASIC source code listings by structuring printouts in a clear, orderly form with a minimum of extra characters. Written for the Apple II Plus, it can be modified for other BASIC dialects.

Credit is due Mark Capella for the first listing program.¹ Since then, two others have been published.^{2,3} Not completely satisfied, we decided to start fresh with the following rules:

1. Print results so they are easy to read.
2. Make the program adaptable to various printers.
3. Gather statements in strings for flexibility.
4. Separate REMs from printed code.
5. Omit the concatenation colon and "LET."
6. Split over-long print lines at a logical character.
7. Indent FOR-NEXT loops globally.
8. Indent IF-THEN statements locally.
9. Minimize disk operations.

The main program, LISZTER, was written in linear form to accommodate different printers and to allow easy deletion or addition of special features. This article is both a program description and a partial history of program development.

Applesoft Source Code Structure

Source code structure rules the program. One line of Applesoft BASIC is shown in figure 1. Each line contains five overhead bytes: two for a pointer to the next line, two more for the number, and an end-of-line null (binary zero) byte. The last line number source code ends in three null bytes to indicate end-of-program.

All variable names, strings, and punctuation not a function are expressed as 7-bit ASCII with most-significant-bit (MSB) set false or zero. All function words (IF, NEXT, REM, etc.) are stored as one-byte "tokens" with MSB set true or high. There are 107 Applesoft tokens.⁴

Starting the Program Organization

Figure 2 is the initial flow chart. Each program byte is examined, beginning with decimal memory location 2049. ("Standard" ROM Applesoft code begins here. It can be changed and

will vary for other BASICs.) String variables hold the line number in N\$, statement text in a "gather" string, G\$, and the "combination" printout string, C\$.

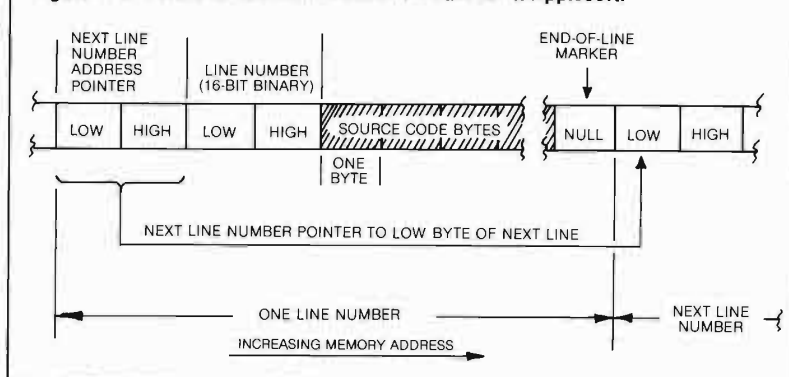
A remarks flag is set if a REM token is encountered. The first decision separates remarks by blank print lines but groups successive remarks without blank lines. Remarks are highlighted without appearing to be part of the main coding.

ASCII characters and token bytes are parsed next with tokens reconverted to the original function word. This section and the print line formatting section receive the most attention. A prime example is separating concatenated statements and allowing indication of over-long text lines.

Holding Two BASIC Programs in Memory

Applesoft reserves two bytes in page zero (first 256 bytes) for the starting address. Start location is normally decimal 2049 for ROM BASIC, stored in locations 103 (low byte) and 104 (high byte). End-of-program in memory is in locations 175 (low) and 176 (high). Either can be changed from the keyboard or program in memory.

Figure 1: Structure of one source-code line number in Applesoft.



Apple's DOS allows the simulation of keyboard commands with an EXEC Text File. An EXEC file loads statements into the keyboard buffer. Each statement is then executed as if it were a keyboard command.

The program to be listed is loaded first. The EXEC file is called next by typing "EXEC LISZT." LISZT then changes normal program start address to the end of program plus two, loads and runs the LISZTER working program. Loading LISZTER will automatically set the new end-of-program address.

Although two programs are now in memory, Applesoft will only execute LISZTER as indicated by the starting address changed by EXEC file LISZT. Original start and end addresses are held in page zero scratchpad locations; LISZTER resets start and end from these scratch locations on completion of printout.

EXEC file LISZT is generated by the short program in listing 1. MAKE LISZT may be deleted after generating LISZT. LISZT EXECution commands are those indicated within quotes in MAKE LISZT line numbers 225 through 265.

LISZTER start location is set slightly higher than normally expected. This and the extra nulls will insure that the listed program can be RUN normally after LISZTER resets start and end addresses on printout completion. Normal source code ending must be three successive null bytes.

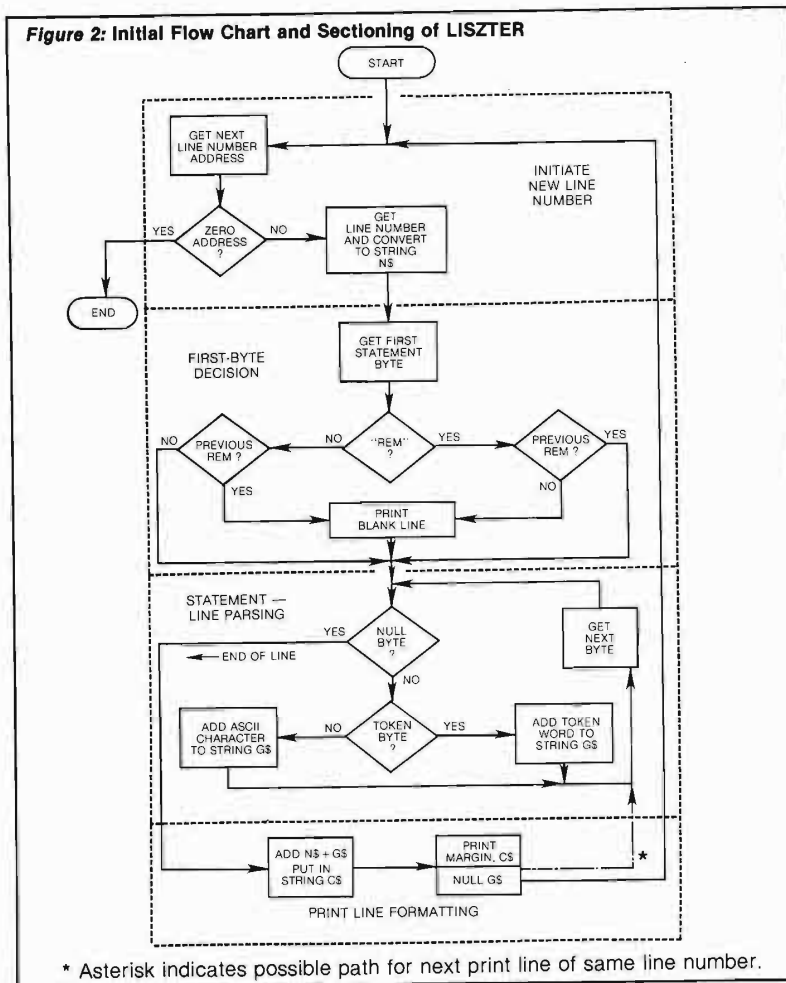
Setting Up LISZTER

LISZTER begins execution at line number 82 by initializing the variables. Initializing will speed up execution, especially with string variables in Applesoft. Token array T\$ contains the 107 function words expressed as literal strings in the DATA statements. Direct expression as strings allows spaces to be added for clarity in gathering and converting the tokens.

The REM token word was changed to an asterisk. It is left as an isolated DATA declaration for those desiring another symbol or word. LET appears as a null string in line 88 to permit completion of the array; token parsing will skip over a LET.

Screen prompts in lines 94 to 100 are optional. Printed page length is normally 60 lines per page including the header. Indent spacing is normally four column spaces, fitting the REM asterisk with three following blanks.

Figure 2: Initial Flow Chart and Sectioning of LISZTER



* Asterisk indicates possible path for next print line of same line number.

Listing 1: MAKE LISZT "EXEC" file generator printed with the LISZTER program in listing 2.

```

MAKE LISZT
LEONARD H. ANDERSON
PRINTOUT ON 20 AUGUST 1981
Page 1

200 * "MAKE LISZT" GENERATOR FOR "LISZT" EXEC FILE
205 * LEONARD H. ANDERSON 7/5/81
210 D$ = Chr$(4)
    C$ = Chr$(13)+D$
215 Print C$"OPEN LISZT"
220 Print C$"WRITE LISZT"
225 Print "POKE208,PEEK(103)"
    Print "POKE209,PEEK(104)"
    Print "POKE210,PEEK(175)"
    Print "POKE211,PEEK(176)"
230 Print "POKE104,PEEK(211)"
235 Print "IF PEEK(210)<254 THEN POKE103, (PEEK(210)+1)"
240 Print "IF PEEK(210)>253 THEN
    POKE103, (PEEK(210)-254);POKE104, (PEEK(211)+1)"
245 Print "POKE (PEEK(210)+PEEK(211)*256-2),0"
250 Print "POKE (PEEK(210)+PEEK(211)*256-1),0"
255 Print "POKE (PEEK(210)+PEEK(211)*256),0"
260 Print "POKE (PEEK(210)+PEEK(211)*256+1),0"
265 Print "POKE (PEEK(210)+PEEK(211)*256+2),0"
270 Print "RUN LISZTER"
275 Print D$"CLOSE"
280 End

End of Listing

```

Program Length = 642 Bytes, Total of 17 Line Numbers
19 Total Non-Rem Statements, 2 Total Remarks

Line 104 assumes the printer interface located in peripheral slot 1. The "POKE 33,30" sets screen window width to 30 columns, necessary only for certain Apple I/O interfaces. Line 106 is the main printer control statement.

Subroutines

"GET BYTE" simply advances program byte pointer B and fetches a decimal value of the byte in B. "BLANK LINE PRINT" is used mainly in REM separation. String S\$ is a single blank space and required by Centronics 737 printers; most other printers will accept a single PRINT without a variable.

"TEST PAGE" (lines 6 through 14) does several things: line count (LC) advance and test, header printing and a "continued" indicator print on the current page if another page is to be printed. Since concatenated statements are separated, some may be printed on the following page. Lines 13 and 14 place brackets around the line number and print that on the following page for clarity.

Upper and Lower Case

The mixture of upper and lower case characters is due primarily to the TS array DATA declarations. We decided that program statements would appear better with the familiar function words in a mixture, non-token statements in normal upper-case only. The choice is up to the user.

Since none of us has direct lower-case control, an available utility program was used to "zap" the desired characters directly on the disk. You should do this on a copy file in disk; strange results occur if a token is accidentally changed!

First-Byte Decision

In the section occupying lines 29 through 33 on listing 2 remarks are separated and grouped. A single, first-byte colon in Applesoft is equivalent to a REM; first-byte colons are changed to REMs since the asterisk-equivalent voids the original source code separation intent.

Statement Line Parsing

Line 34 begins a number of definitions. A null program byte indicates end-of-line in Applesoft and jumps to the print line formatting routine at line

LISTER 'UNIVERSAL' VERSION
LEONARD H. ANDERSON
24 JULY 1981
Page 1

```

0 goto BZ
1 * "GET BYTE" SUBROUTINE
2 P = P+1
3 B = Peek(P)
4 D = 0
5 * "BLANK LINE PRINT" SUBROUTINE
6 Print B$
7 GOSUB 111
8 LC = 6
9 PC = PC+1
10 Print B$
11 Print B$;Label$;"(continued)"
12 * "FORM-FEED TO GET TOP OF NEXT PAGE AFTER 'CONTINUED',
13 NOTICE, HOTEL PRINTERS WITH F-F COMMAND CAN USE 'PRINT
14 CORRECT' IN PLACE OF FOR-NEXT LOOP, THE SINGLE-SPACE
15 'PRINT S$' (THROUGHOUT) IS REQUIRED BY CENTRONICS 737."
16 For K = LP To 63
17 Print B$
18 For K = LP To 63
19 * PRINT THE HEADER, BRACKET LINE NUMBER FOR STATEMENTS
20 SPLIT OVER TWO PRINT PAGES
21 H$(4) = "Page "+Str$(PC)
22 For K = 1 To 4
23 E = Int((LL-Len(H$(K)))/2)+1
24 Print H$(1+3*(B$(E)+1));H$(K)
25 Next
26 K = P+(0)
27 Print B$
28 If Not D Then
29 Return
30 * PUT LINE NUMBER IN BRACKETS AS A STATEMENT
31 IDENTIFICATION ON NEXT PAGE
32 NO = Str$(Val(NO))
33 K = Len(NO)
34 * NO IS NON WITHOUT SPACES; BRACKET N$ AND ATTACH TO
35 STATEMENT CHARACTERS
36 C$ = Right$(Left$(LB$, (6-K))+Chr$(91)+N$+Chr$(92)+B$, 61)
37 K = P+(0)
38 Right$(C$, Len(C$)-B1)
39 Return
40 * GET THE TWO-BYTE POINTER TO NEXT LINE NUMBER; NULL
41 INDICATES END.
42 GOSUB 2
43 D = B
44 GOSUB 2
45 D = Z$68B+D
46 If D> 255 goto Z7
47 * END OF PROGRAM, PRINT NOTICE
48 GOSUB 111
49 GOSUB 4
50 GOSUB 4
51 GOSUB 6
52 Print B$;Label$;"End of Listing"
53 (continued)

```

54. A decimal value between 1 and 127 is an ASCII character byte; any value above 127 is a token.

The double quote test at line 36 allows colons *within* quotes or remarks. Any other colons are treated as delimiter characters and tested at line 37. A delimiter forces a new print line but not a new line number as in the case of a null value byte.

Control characters are converted to upper case equivalents. Besides making control characters visible, conversion allows a printer to continue without suddenly switching to a new mode! We enclosed control characters in vertical bars because that print character has little use in normal printing.

Token byte values are changed to allow you to gather them from the T\$ array. A token value out of normal range is made into a distinctive word at line 40. A test-true here would indicate an error.

The REM flag set at line 43 is primarily for concatenated remarks. The remarks counter is optional and used only for end-of-listing statistics. REM spacing variable RS is set to one for indenting remarks. While remarks are highlighted, we also wanted their appearance out of the normal program flow.

The FOR flag sets up the start of global FOR-NEXT indenting. The FOR spacing counter is advanced in print line formatting to allow completion of the entire FOR statement. The NEXT test at line 48 removes one FOR indent space. This space is held at zero in case an intermediate (but legal) NEXT is used with the loop.

Conditional tests add an indent space on completion of a THEN. Anything following a THEN, even if only a line number, is considered a separate statement. An IF-GOTO is considered a single statement. The choice was arbitrary to reduce total code.

A LET token is ignored by choice. Omitting line 47 allows you to print a LET.

DATA flag (DF) is used solely in print formatting. When set, it allows splitting an over-long print line only on commas. This is useful when DATA declarations contain strings with spaces as in LISZTER itself.

Listing 2 (Continued)

```

19 * OPTIONAL STATISTICS
20 Gosub 4
   Gosub 4
   Gosub 6
   Print M$;"Program Length = _
     ";(Peek(211)-Peek(209))*256+Peek(210)-Peek(208);" Bytes, _
     Total of ";TN;" Line Numbers"
21 Gosub 4
   Gosub 6
   Print M$;(TB-TR);" Total Non-Rem Statements, ";TR;" Total_
     Remarks"
   Gosub 4
   Gosub 6
   Print M$;"END"
22 * TURN OFF PRINTER, DISPLAY END PROMPT ON SCREEN
23 Pr# 0
   Poke 33,40
   Home
   VTab 12
   HTab 11
   Inverse
   Print " END OF LISTING "
   Normal
24 * RESET PAGE 0 POINTERS FOR THE LISTED PROGRAM
25 Poke 105,Peek(210)
   Poke 106,Peek(211)
   Poke 107,Peek(210)
   Poke 108,Peek(211)
   Poke 109,Peek(210)
   Poke 110,Peek(211)
   Poke 111,Peek(115)
   Poke 112,Peek(116)
   Poke 103,Peek(208)
   Poke 104,Peek(209)
   Poke 175,Peek(210)
   Poke 176,Peek(211)
   End
26 * MAKE THE LINE NUMBER STRING
27 TN = TN+1
   Gosub 2
   D = B
   Gosub 2
   K = B*256+D
   D = Len(Str$(K))
   N$ = Right$(Left$(LB$(7-D))+Str$(K)+" ",B)
28 * BEGIN LINE PARSING WITH FIRST-BYTE DECISION
29 TB = TB+1
   D = 0
   Gosub 2
   If B = 5B Then
     B = 17B
     * CONVERT "SIMPLE REM" (A "1" FIRST-BYTE) TO ORDINARY
       "REM"
30 If B = 17B And Not RF Then
   Gosub 4
   Goto 34
   * "REM" FLABS ARE SET AFTER SEPARATION OF TOKENS;
     REM-GROUPS SEPARATED BY BLANK PRINT LINES.
31 If B = 17B And RF Goto 34
   * BYPASS RF RESET
32 If RF Then
   RF = 0
   Gosub 4
33 * RE-ENTRY POINT FOR NEXT BYTE IN STATEMENT DECISION FLOW
34 If B = 0 Goto 54
   * FORCE A NEW LINE ON THE END-OF-LINE NULL MARKER
35 If B>127 Then
   B = B-127
   Goto 41
   * BYTE IS A TOKEN; REMAINDER ARE CHARACTERS (continued)

```

Listing (Continued)

```

36 If B = 34 Then
    QF = -QF
    * TOGGLE QUOTE FLAG FOR COLON-PRINT TEST IN NEXT LINE

37 If B = 58 And Not RF And QF<1 Then
    TS = TS+1
    Goto 54
    * OMIT THE CONCATENATION "1" AND FORCE A NEW LINE, ELSE
    PRINT THE COLON AS A CHARACTER

38 If B<32 Then
    B = B+64
    G$ = G$+Chr$(124)+Chr$(B)
    B = 124
    * PRINT CONTROL CHARACTERS AS UPPER-CASE BETWEEN
    VERTICAL BARS; INDICATOR OF CONTROL CHARACTER
    OPTIONAL.

39 G$ = G$+Chr$(B)
    Gosub 2
    Goto 34

40 * INDICATE UNUSED TOKENS AND CONTINUE

41 If B>107 Then
    G$ = G$+" ?! "
    Gosub 2
    Goto 34

42 * ACCEPTABLE TOKENS...

43 If B = 51 Then
    TR = TR+1
    RF = 1
    RS = 1
    * SET BOTH FLAGS AND TOTAL-COUNT ON "REM"

44 If B = 2 Then
    FF = 1
    * A "FOR" IS STARTED

45 If B = 69 Then
    CF = 1
    G$ = G$+T$(B)
    Goto 54
    * FORCE A NEW LINE AFTER PRINTING A "THEN"

46 If B = 4 Then
    DF = 1
    * "DATA" STATEMENT BEGUN; WILL AFFECT INDENTING LATER

47 If B = 43 Then
    Gosub 2
    Goto 34
    * IGNORE A "LET" (IT IS A NULL STRING IN DATA STATEMENT
    LINE #88)

48 If B = 3 Then
    FS = FS-1
    If FS<0 Then
        FS = 0
    * "NEXT" TOKEN REMOVES A "FOR" LOOP INDENT

49 G$ = G$+T$(B)
    Gosub 2
    Goto 34

50 * ADD EXTRA INDENT FOR EACH SPLIT LINE, LIMITING FOR
    LINE-UP OF "REM" AND "DATA" PRINT-OUTS

51 SF = 0
    RS = RS+1
    If RS>2 Then
        RS = 2

52 If DF And RS>1 Then
    RS = 1

53 * GET TOTAL INDENT SPACES FOR PRINT LINE PLUS LOW-LIMIT FOR
    SPLIT-POINT ("E")

54 K = IM*(FS+CS+RS)
    E = K+13
    If K>0 Then
        G$ = Left$(BB$,K)+G$

```

(continued)

Print Line Formatting

The next part of the program sets up indent spaces and splits over-long print lines on a selected character. Splitting is done on ASCII characters since gather string G\$ contains only ASCII values on entrance at line 54.

FOR, REM, and IF spacing counters are added at line 54, multiplied by IM (default value of four), and inserted ahead of G\$. Temporary variable D is an indicator to insert the line number on the first statement. G\$ is set into C\$, then tested for length at line 59. If the C\$ string length is too long, it is split with the right side remainder replacing the former contents of G\$.

Splitting has two priorities. The first priority split occurs at the right-most available space, if it is not a DATA statement. The second priority is an arithmetic operator character (ASCII, not token) or comma; DATA statements split only on commas. While the second priority choice seems arbitrary, it is convenient in terms of ASCII values.

Splitting character search is right to left, beginning with the last available print line column determined by LL. Left limit is determined by temporary variable E (line 54). The original program had an undesired zero left limit; a few print lines were endless blanks!

Another undesired condition occurred with spaces in long strings or PRINTs going beyond the right limit. There was no way to determine if a space existed in the printout. This is solved by lines 74 and 75 adding an underline at the right-most space of the first line, or left-most space of the next line.

Final Print and Cleanup

Every new line calls the TEST PAGE subroutine. This determines if a new page is called for and, if so, prints the "continued" reminder at the bottom, form-feeds, then prints a header on the next page.

Deciding on a one-statement-per-line format gave us the possibility of one or more unnumbered statements on the next page. Holding the readability rule, we decided on placing the next-page line number in brackets (seldom used in Applesoft) while holding the number print justification. Lines 13 and 14 take care of this. An early version used two colons between the

number and statement but conflicted with Backus-Naur notation.

Uncompleted split lines jump to line 51 for extra indents. A remark allows one extra indent count to line up the remark second line with first line text. The REM symbol used here takes four columns or one default indent space. The DATA declaration single indent (line 52) seemed to be most readable.

Separate flag and counter variables on FOR and IF statements allow for concatenation in one line number of source code and the global or local indenting in printout. Local indenting of conditionals is reset on a new line number but global indenting of FOR loops is decremented only on a NEXT token at line 45.

A new source code line number is begun only when the program byte contains the end-of-line null.

Ending it All

Applesoft indicates the end of a listing by three successive nulls. This would appear as a zero line number — a second zero line number, since LISZTER begins with line number zero. This second zero line number falls through the IF in line 16 to begin optional statistical printouts at lines 18, 20, and 21.

Line 23 disables all Apple peripherals by "PR#0", resets screen width to normal by "POKE 33,40", and indicates a finish on the screen. The print command at line 106 allowed the screen to be active at all times even though lower case characters appear as nonsense on a standard Apple.

The POKes in line 25 reset the start and end pointers to their original values prior to the EXEC file command. Variable and array space pointers are also reset permitting the user to RUN the program after LISZTing.

Optional Starting Prompts

The "RUN 23" notice in line 94 should remain until the user is very familiar with LISZT. It is the only way to restore start and end pointers after a RESET. Address locations in line 95 are optional, useful only with very long programs.

Page length, left margin, and indent spacing are useful only if different paper is used. If available, different vertical printer spacing could be added to

Listing 2 (Continued)

```

55 * ADD LINE NUMBER OR EQUIVALENT-SPACE BLANK
56 If Not D Then
   C# = N#+B#
57 If D Then
   C# = LB#+B#
58 * TEST FOR LONG LINE, SPLIT IF NECESSARY
59 K = Len(C#)-LL
   If K<1 Goto 73
   * NOT A SPLIT LINE
60 B# = Right$(C#,K)
   C# = Left$(C#,LL)
   BF = 1
   If DF Goto 65
61 * START SPLIT WITH A SPACE FIRST IF NOT "DATA"
62 D = LL
63 If Mid$(C#,D,1) = S# Goto 71
64 D = D-1
   If D>E Goto 63
65 D = LL
   * SPLIT NEXT AT ARITHMETIC OPERATOR OR COMMA
66 K = Asc(Mid$(C#,D,1))
   If K<42 Or K>47 Goto 69
67 If DF And K = 44 Goto 71
   * "DATA" STATEMENTS SPLIT ONLY ON COMMAS
68 If Not DF And K<>46 Goto 71
   * OTHER STATEMENTS SPLIT BY ALL BUT PERIOD
69 D = D-1
   If D>E Goto 66
70 Goto 73
   * FALL-THROUGH INDICATES END-OF-PRINT-LINE SPLIT
71 K = LL-D
   If K>0 Then
   B# = Right$(C#,K)+B#
   C# = Left$(C#,D)
72 * TEST PAGE LINE-COUNT, INSERT SPACES AS ALLOWED, THEN
   PRINT AT LINE #76. NOTE: SINGLE CHARACTER SET PRINTERS
   SHOULD USE ONLY "PRINT M#;C#" BEFORE "K = FRE(0)" IN LINE
   #76.
73 Gosub 6
   K = Len(C#)
   If BF = 0 Or K<2 Or RF Then
   76
74 If Mid$(C#,K,1) = B# Then
   C# = Left$(C#,(K-1))+Chr$(95)
   * PUT A TRAILING UNDERLINE IN PLACE OF THE LAST SPACE
   AS A MARKER FOR THE LEFT-HAND STRING
75 If Len(B#)>2 And Left$(B#,1) = S# Then
   B# = Chr$(95)+Right$(B#,(Len(B#)-1))
   * PUT A LEADING UNDERLINE IN PLACE OF THE FIRST SPACE
   OF RIGHT-HAND STRING AS A MARKER
76 Gosub 111
   K = Len(C#)
   Print M#;Left$(C#,B#)
   Gosub 112
   Print Right$(C#,(K-B#))
   K = FRE(0)
   If BF Then
   D = 1
   Goto 51
   * PRINT REST OF A SPLIT LINE
77 DF = -1
   RS = 0
   DF = 0
   If FF Then
   FS = FS+1
   FF = 0
78 D = 0
   If CF Then
   CS = CS+1
   CF = 0

```

(continued)

the page length prompt at line 98. A variable left margin requires the BB\$ string to be slightly longer than one-half print line width.

We recommend that you retain the inverse video reminder at line 102. Concentration on program development makes us forget the right buttons to push at crucial moments!

Final Thoughts

A "REM-less" version of LISZTER is about 3.9K long and will run in 5.5K of free memory. Disk operations are not required after the initial EXEC LISZT command.

Hesitation in execution occurs only in parsing long character lines. LISZTER's line 76 takes about 20 seconds to gather, split, and begin printing. The 256-byte string maximum has not yet been reached, including one LISZTING over 30 print pages.

Lack of concatenation character does not seem to hamper reading. Those familiar with the interpreter syntax will know it is always there. Statement separation is easier to understand and is improved further with indenting.

Thanks are due to Cliff Bruhn, Dennis Kaloi, Sterling Tate, Wes Ten, and Bob Keene of Candid Computers for their trial runs, comments, and suggestions.

References

1. "FP LISTER," Mark Capella, *CALL-A.P.P.L.E.*, May, 1980.
2. "Apple BASIC Prettyprinter," Michael Keith, *Microcomputing* August, 1981, pp.110, 113.
3. "Program Listing Formatter," Robert C. Clardy, M. Capella, J. Morrisset, C. Anson, V. Golding, *CALL-A.P.P.L.E. In Depth*, Number One (September 1981), pp. 102, 105.
4. *Applesoft Reference Manual*, Apple Computer Incorporated.
5. *Apple II Disk Operating System Reference Manual*, Apple Computer Incorporated.
6. "The Inspector," Omega Micro-ware, Inc., (one of several "zappers" available).

Listing 2 (Continued)

```

79 SF = 0
   G$ = ""
   If B = 0 Then
     CS = 0
     Goto 16
     * GET ANOTHER PRINT LINE IF NOT E-O-L NULL, ELSE FALL
     THROUGH AND GET ANOTHER LINE NUMBER
80 Gosub 2
   D = 1
   Goto 34
81 * INITIALIZATION OF VARIABLES
82 Dim T$(107),H$(4)
83 * INITIAL VARIABLE SETTING HAS AN 80-CHARACTER WIDE PRINT
   LINE AND 60-LINE PAGE LENGTH (INCLUDING HEADER, EXCLUDING
   'CONTINUED' INDICATOR); CHANGE LL AND LP AS DESIRED FOR
   OTHER FORMAT SIZE.
84 * THE "P=2048" IN LINE #85 ASSUMES A NORMAL APPLESOFT ROM
   START AT DECIMAL ADDRESS 2049. CHANGE FOR APPLESOFT IN
   RAM.
85 P = 2048
   B = 0
   RS = 0
   CS = 0
   FS = 0
   RF = 0
   CF = 0
   FF = 0
   DF = 0
   SF = 0
   QF = -1
   LL = 80
   LP = 60
   IM = 4
   E = 0
   TN = 0
   TS = 0
   TR = 0
   S$ = " "
86 C$ = ""
   N$ = ""
   G$ = ""
   M$ = ""
   H$(0) = ""
   LB$ = " "
   BB$ = " "
87 Data "End","For ","Next ","Data ","Input ","Del ","Dim ","Read ",
   "Gr","Text","Fr$","In#","Call ","Plot ","HLin ","VLin ",
   "HGr2","HGr","HColor = ","HPlot ","Draw ","XDraw ","HTab ",
   "Home","Rot = "
88 Data "Scale = ","ShLoad","Trace","NoTrace","Normal","Inverse",
   "Flash","Color = ","Pop","VTab ","Hmem : ","Lomem : ",
   "OnErr ","Resume","Recall ","Store ","Speed = ","","Goto ",
   "Run","If ","Restore","& "," Gosub ","Return"
89 Data "*"
   * ^ CHANGE "REM" TOKEN WORD INDICATOR AS DESIRED
90 Data "Stop","On ","Wait ","Load ","Save ","Def ","Poke ","Print ",
   "Cont","List ","Clear","Get ","New","Tab("," To ","Fn ",
   "Spc("," Then "," At ","Not "," Stop ","+","-","*"
91 Data "/","^"," And "," Or ",">"," = ","<","Sgn","Int","Abs","Usr",
   "Fre","Scrn(","Pd1","Pos","Sqr","Rnd","Log","Exp","Cos","Sin",
   "Tan","ArcTan","Peek","Len","Str$","Val","Asc","Chr$","Left$",
   "Right$","Mid$"
92 For K = 1 To 107
   Read T$(K)
Next
93 * SCREEN PROMPTS AND ALTERNATE LISTING CONSTANTS
94 Home
   VTab 3
   Flash
   Print " RUN 23 ";
   Normal
   Print " RESTORES ORIGINAL AFTER RESET"
   * 'RUN 23' RESTORES POINTERS FOR PROGRAM START AND END TO
   ORIGINAL VALUES AND RESETS SCREEN

```

(continued)

Listing 2 (Continued)

```

95 Print
Print "START OF PROGRAM LISTED: ";Peek(209)*256+Peek(208)
Print " END OF PROGRAM LISTED: ";Peek(211)*256+Peek(210)
Print "      END OF 'LISZTER': ";Peek(176)*256+Peek(175)

96 * ABOVE IS OPTIONAL, CHECKS TO SEE IF THE 'LISZT' EXEC-FILE
    IS OPERATING PROPERLY

97 Print
Input "PROGRAM NAME: ";H$(1)
Input " PROGRAMMER: ";H$(2)
Input "      DATE: ";H$(3)
Print

98 Print
Print "PAGE LENGTH IS 60 LINES, WANT OTHER?"
Get H$(0)
If H$(0) = "Y" Then
    Input " PAGE LENGTH: ";LP
    If LP>62 Goto 9B
        * LIMIT TO 11" LENGTH AND HEADER START-POSITION; CAN
          CHANGE WITH SMALLER-SPACING PRINTERS.

99 Print
Print "NO LEFT MARGIN, WANT ONE ?"
Get H$(0)
If H$(0) = "Y" Then
    Input " MARGIN SPACES: ";K
    If K>0 And K<49 Then
        M$ = Left$(BB$,K)
        LL = LL-K
        * MARGIN & LINE-LENGTH UNTOUCHED ON WRONG INPUT,
          REMAINS AT DEFAULT VALUE

100 Print
Print "INDENT SPACING = 4, WANT OTHER ?"
Get H$(0)
If H$(0) = "Y" Then
    Input " SPACING: ";IM
    If IM<0 Or IM>12 Goto 100

101 * REMINDER FOR PRINTER SET-UP
102 Home
    Inverse
    Print " BET PAPER TO TOP OF FORM "
    Print "      THEN "
    Print "      TURN ON PRINTER "
    Normal
    Print
    Get H$(0)

103 * SET SCREEN WIDTH, TURN ON PROPER PORT

104 Home
    Poke 33,30
    Pr# 1

105 * SET-UP FOR EPSON MX-80 PRINTER WITH ORANGE MICRO
    'GRAPPLER' OR CENTRONICS-COMPATIBLE PARALLEL INTERFACE
    CARD. CCS CARD MUST ADD 'CHR$(9)"K"' TO REMOVE EXTRA
    LINE FEED.

106 Print Chr$(9)"82N"Chr$(9)"I"

107 * RESERVED LINE FOR OPTIONAL PRINTER CONTROL
108 * CHR$(9) = "CONTROL-I"

109 LC = 6
    PC = 1
    Gosub 11
    Goto 16

110 * MX-80 ITALICS/STANDARD CHARACTER SET SWITCHING
    SUBROUTINES (APPLIES ONLY TO "GRAFRAX"-AUGMENTED
    PRINTERS)

111 Print Chr$(27)"5";
    Return
    * ESC-5 IS STANDARD SET

112 Gosub 111
    If RF Then
        Print Chr$(27)"4";
        * ESC-4 IS ITALICS SET USED FOR "REM"S

113 Return

```

(continued)

Problems You May Encounter with LISZT with Strings

1. A colon ending a line causes a stop and 'error at line 76' display. The best solution is to use a line editor program or keyboard to correct the program line to remove the extraneous byte. Usually appears to be a 'forgotten' removal during program editing.
2. A double colon starting a line causes LISZTER to think the first colon is a REM, but the second colon causes reversion to gathering tokens and characters in the usual manner. Using an italics set on the printer will make this line look like a REM splat, but has both upper and lower case contents. Best solution is to edit out the extra colons.
3. A statement ending nested FOR loops such as "NEXT J,K,L" executes in Apple-soft as if they were three separate NEXT statements. Since LISZTER will only recognize one NEXT token, all following lines will retain the FOR-NEXT indent(s) for the remainder of printout.

We don't have a simple solution for this — yet. Changing the program to "NEXT J:NEXT K:NEXT L" will add only two bytes and bring the left margin back to normal. The two added bytes are the NEXT tokens; concatenation colons take the place of the commas.

4. On any mid-printout deliberate stop, such as RESET, you must key in RUN 23 to restore the program start and end pointers. Failure to do so may attach LISZTER to the program being listed.

Richard F. Searle is manager of software and control systems at an aerospace corporation and a consultant on the application of solar energy. You can write to Mr. Searle at 4511 Adam Road, Simi Valley, California 93063.

Donald Cohen is an electronics engineer specializing in microprocessor control of aircraft emergency power systems. He is also a programming consultant and partner of Donald Cohen Associates. Contact Mr. Cohen at Donald Cohen Associates, 4613 Wolfe Way, Woodland Hills, California 91364.

Leonard H. Anderson is a hardware and systems engineer, who uses his Apple II for number crunching (when not playing games). He is also a contract writer and currently an Associate Editor with *Ham Radio Magazine*. Address correspondence to 10048 Lanark Street, Sun Valley, California 91352.

Listing 2 (Continued)

```

114 * "LISZTER"
115 * Working Program to
116 * re-format
117 * APPLESOFT Programs for
118 * Printing
119 *
120 * BY
121 * LEONARD H. ANDERSON
122 *
123 * Version 4.1.3, 7/24/81
124 * (lower-case version)
125 * MA-80 & "GRAFRAX"
126 * (ITALICS ON REMS)
127 *
128 * DESCRIPTION OF VARIABLES:
129 *
130 * B PROGRAM BYTE DECIMAL VALUE
131 * BB# 'BIG BLANK' STRING OF 48 SPACES
132 * CF "IF" FLAG; 1 = "IF" STARTED; 0 = NO "IF"
133 * CS "IF" (CONDITIONAL) INDENT SPACE COUNTER
134 * C# CHARACTER AND TOKEN STRING TO BE PRINTED
135 * D 'DIRECTION', A TEMPORARY
136 * DF "DATA" FLAG (ALLOWS SPLIT ON COMMA ONLY)
137 * 1 = "DATA" EXISTS ON LINE; 0 = NO "DATA"
138 * E TEMPORARY, PARTLY FOR SPLIT-LINE LIMITS
139 * FF "FOR" FLAG; 1 = "FOR" STARTED; 0 = NO "FOR"
140 * FS "FOR" INDENT SPACING COUNTER
141 * G# 'GATHER' STRING TO BUILD STATEMENT LINE
142 * H# HEADER ARRAY FOR PAGE TITLE
143 * IN INDENT SPACE MULTIPLIER
144 * K TEMPORARY
145 * LB# 'LITTLE BLANK' STRING OF 8 SPACES
146 * LC LINE COUNTER FOR PAGINATION TEST
147 * LL LINE-LENGTH (WIDTH) CONSTANT
148 * LP LINES-PER-PAGE CONSTANT
149 * M# LEFT MARGIN SPACING STRING
150 * N# LINE NUMBER STRING
151 * P POINTER TO PROGRAM BYTE (DECIMAL VALUE)
152 * PC PAGE COUNTER FOR HEADER ON EACH PAGE
153 * QF QUOTE FLAG TO ALLOW/DISALLOW COLON PRINTING
154 * -1 = NO QUOTE OR SECOND QUOTE OF PAIR EXISTS
155 * +1 = FIRST QUOTE OF PAIR EXISTS, ALLOW COLONS
156 * RF "REM" FLAG; 1 = "REM" STARTED; 0 = NO "REM"
157 * RS "REM" INDENT SPACING COUNTER
158 * SF SPLIT-LINE FLAG; SET IF PRINT LINE MUST BE SPLIT
159 * S# SINGLE SPACE STRING
160 * TN TOTAL LINE NUMBER COUNTER
161 * TR TOTAL REMARK-STATEMENT COUNTER
162 * TS TOTAL STATEMENT COUNTER
163 *
164 * AN EXAMPLE OF INDENTS ON NESTED "FOR" LOOPS:
165 For J = 1 To 25
166 For K = J To 26
167 If MT(J,K) = 0 Goto 170
168 For L = J To K
169 If MT(J,K) <> 0 Then
170 MT(K,L) = MT(K,L) - (MT(J,K) * MT(J,L))
171 * BEGINS WITH "LET MT(...)"
172 Next L
173 Next K
174 Next J
175
176 * // // //
17700 * THE PRECEDING LINE CONTAINED TWO CONTROL-1 CHARACTERS
178000 * SEPARATED BY A CONTROL-6 (BELL).

```

End of Listing

Program Length = 10061 Bytes, Total of 175 Line Numbers

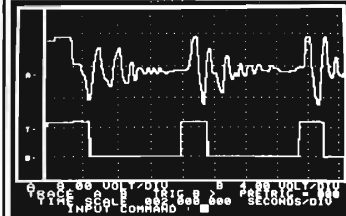
271 Total Non-Rem Statements, 119 Total Remarks

END

AKRO

APPLESCOPE

DIGITAL STORAGE OSCILLOSCOPE
Interface for the Apple II Computer



The APPLESCOPE system combines two high speed analog to digital converters and a digital control board with the high resolution graphics capabilities of the Apple II computer to create a digital storage oscilloscope. Signal trace parameters are entered through the keyboard to operational software provided in PROM on the DI control board.

- DC to 3.5 Mhz sample rate with 1024 byte buffer memory
- Pretigger Viewing
- Programmable Scale Select
- Continuous and Single Sweep Modes
- Single or Dual Channel Trace

Price for the two board Applescope system is \$595
EXTERNAL TRIGGER ADDAPTER \$29

APPLESCOPE ACCESSORIES

APPLESCOPE-HR12 High resolution 12 bit analog to digital converter with sample rates to 100 Khz. Requires 48K Apple II with disk drive. Software provided on floppy disk includes basic SCOPE DRIVER package.

Price per channel \$695

APPLESCOPE-HRHS High Resolution AND High Speed. Circuit combines two 6 bit flash analog to digital converters to give a 10 bit dynamic range. The 10 bit converter resolutions maintained at sampling rates up to the 7 Mhz, maximum for signal slew rates less than 5 volts per microsecond. Larger inputs slew rates will reduce the converter resolution to 6 bits until the signal stabilizes within the .5 Volt per microsecond limit. Requires 48K Apple II with disk drive. Software provided on disk includes the basic SCOPE DRIVER package. Price per channel \$695

APPLESCOPE-EXT External trigger adapter has a switch selectable external trigger input to a BNC connector mounted in a rear slot of the Apple II computer. Price \$29.00

APPLESCOPE-BNC BNC adapter connects the Berg stick connectors on the A1 circuit card to male BNC plugs mounted in a rear slot of the Apple II computer. Price \$14.95

BUS EXTENDERS Allow easy access to Apple II peripheral circuit cards. Price \$19.95

SCOPE PROBES Oscilloscope probes for use with the APPLESCOPE - BNC adapter. Price \$29.95

SCOPE DRIVER Advanced software for the Applescope system provided on 5 1/4" floppy disk. Available options include:

- Signal Averaging - Acquires 1 to 256 signal sweeps and displays the averaged result.
- Digital Volt Meter - Allows use as a real time DVM or use to measure points on an acquired sweep.
- Hard Copy - Uses graphics printer to produce hard-copy output of displayed traces.
- Disk Storage - Allows automatic storage and recover of acquired data on floppy disks.
- Spectrum Analyzer - Calculates and displays frequency spectrum of acquired data.

BUS RIDER

LOGIC ANALYZER for the APPLE II

The BUS RIDER circuit card silently rides the Apple II peripheral bus and allows real time tracking of program flow. Software provided on EPROM allows set up of trace parameters from the keyboard and read back of disassembled code after a program has been tracked.

- 32 bit by 1024 sample memory buffer
- Monitors Data and Address bus plus 8 external inputs
- Trigger on any 32 bit word or external trigger
- Pretigger viewing

The BUS RIDER is an invaluable development tool for anyone working with Apple II or Apple II+ computers. Price \$295

RC ELECTRONICS INC.

7265 Tuolumne St., Goleta, CA 93117

Visa

(805) 968-6614

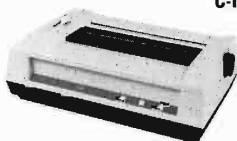
M/C



FOR YOUR APPLE II

Industry standard products at super saver discount prices

PARALLEL PRINTERS



C-ITOH 8510 PRINTER Virtually identical to the NEC 8023. Same electro-mechanical internals. The only difference is the name and the price. Specifications: • 100 CPS dot matrix printer • 80 column print—136 characters per line • Tractor/friction feed • 7 different print fonts included • 2K printer buffer • Proportional spacing • Bit image graphics and graphic symbols.

NEC & C-ITOH List: \$800

Special Price with any other purchase \$595

EPSON 100 PRINTER \$749

Z-80 CARDS FOR YOUR APPLE

MICROSOFT SOFTCARD. With CP/M® & MBASIC.

(List: \$399) **\$289**

NEW! ADVANCED LOGIC SYSTEM Z-CARD. With C/PM®.

Has everything the Softcard has except MBASIC. Works with Microsoft's disks too.

(List: \$269) **\$209**

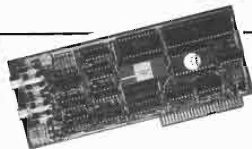
NEW! U-Z-80 PROCESSOR BOARD (from Europe)

Software compatible with Softcard and ALS Software..... (List: \$199) **\$149**

With CP/M®, add: (List: \$75) **\$49**

NEW! ALS SYNERGIZER. CP/M™ operating package with an 80 column video board, CP/M® interface, and 16K memory expansion for Apple II. Permits use of the full range of CP/M® software on Apple II and Apple III.

(List: \$749) **\$595**



80 COLUMN VIDEO DISPLAYS FOR APPLE II

SMARTERM.
(Not to be confused with SUPRTERM)

Software switching from 80 to 40 and 40 to 80 characters. 9 new characters not found on the Apple keyboard. Fully compatible with CP/M® and Apple PASCAL®. With lowest power consumption of only 2.5 watts.

(List: \$360) **\$279**

SMARTERM EXPANDED CHARACTER SET. Add to above: **\$40**
7" x 11" matrix with true decoders.

VIDEX-80 \$279

WIZARD \$269

MONITORS FOR YOUR APPLE

AMDEX 300G \$229

SUPER SPECIALS!

12" GREEN MONITOR \$139

12" BLACK & WHITE MONITOR \$125

All equipment shipped factory fresh. Manufacturers' warranties included. Please add \$3.00 per product for shipping and handling. California: add 6% tax; BART Counties: 6½%.

EXPAND YOUR APPLE II MEMORY

16K MEMORY EXPANSION MODULE. The preferred 16K RAM Expansion Module from PROMETHEUS. Fully compatible with CP/M® and Apple Pascal®. With full 1-year parts and labor warranty.

(List: \$169) **\$99**

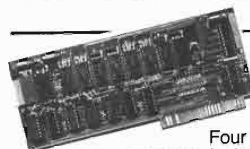
MEMORY MANAGEMENT SYSTEM. Provides more free memory in BASIC by moving DOS into 16K card.

(List: \$49) **\$35**

16K RAM ADD-ON KITS. Eight 16K full specification RAMS.

Reduced to **\$19**

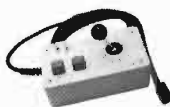
SATURN SYSTEMS 32K RAM CARD \$199



VERSAcard from PROMETHEUS

Four cards on one! With true simultaneous operation. Includes: (1) Serial Input/Output Interface, (2) Parallel Output Interface, (3) Precision Clock/Calendar, and (4) BSR Control. All on one card. Fully compatible with CP/M® and Apple Pascal®.

(List: \$249) **\$195**



JOYSTICK

Takes the place of two Apple Paddle Controllers. From BMP Enterprises. Heavy duty industrial construction and cable. Non-self centering. With polarity switches for consistent motion control.

(List: \$59) **\$39**

MODEMS FOR YOUR APPLE II

HAYS Smartmodem \$229

NOVATION Apple Cat \$349

CENTRONICS COMPATIBLE PARALLEL INTERFACE

NEW! Look out Grappler! From PROMETHEUS. For use with Epson, NEC, C-ITOH, and other printers. Fully compatible with CP/M® and Apple Pascal®.

Only **\$69**

With Screen Dump Graphics for EPSON printer, add **\$25**



Phone for Quick Shipment!

(415) 490-3420

SGC

"The mail order specialists"

342 Quartz Circle, Livermore, CA 94550

GET FREE SOFTWARE FOR YOUR COMPUTER!

HOW? JUST ORDER ANY OF THE ITEMS BELOW, AND SELECT YOUR FREE SOFTWARE FROM THE BONUS SOFTWARE SECTION, USING THE FOLLOWING RULE: FOR THE FIRST \$100.00 WORTH OF MERCHANDISE ORDERED TAKE 1 ITEM; FOR THE NEXT \$200.00 WORTH OF MERCHANDISE ORDERED TAKE ANOTHER ITEM; FOR THE NEXT \$300.00 TAKE A THIRD ITEM, ETC. ALL AT NO COST.

HARDWARE by APPLE COMPUTER APPLE II + .48K 1199 PASCAL LANG. SYS. 397 FLOPPY DR. + CNTRLR 535 FLOPPY DRIVE 455 We carry the rest of the APPLE line at low, low prices! CALL!	PRINTERS EPSON: MX80 495 MX80 F/T 575 MX100 w/Graftrax 775 MX70w/Graftrax 375 APPLE Intfc/Cbl 85 GRAFTRAX 90 GRAPPLER Intfc 149 2K Buffer Ser. Card 135 MX80 Ribbon 15 MX100 Ribbon 24 C.ITOH: 25 CPS Daisy Par. 1440 25 CPS Daisy Ser. 1495 45 CPS Daisy Par. 1795 40 CPS Daisy Ser. 1895 Daisy Tractor Option 225 PRO-WRITER Par. 515	COMPUTER SYSTEMS HEATH/ZENITH: Z89 FA Computer (48K) 2150 Z90 Computer 2490 Z19 Terminal 725 Z47 Dual 8 Drive 2775 WORDSWORTH by LEADING EDGE: (A Zenith Z89 & C.ITOH 25 cps Starwriter & CP/M & Full Word-Processing Software & Super-Calc) 4500 HEWLETT-PACKARD: HP125 2599 HP85 2099 82903A 16K Mem. 250 Z90/M Dual 5" Drive 2125 NEC: PC8001A 32K Computer 1099 PC8012A1/O Unit 699 PC8031A Dual Drives 1099 CP/M 2.2 for NEC 125 Wordstar for NEC 299 Super Calc for NEC 279 COMMODORE: VIC 20 Computer 259 VIC 1515 Printer 315 8032 Micro Mainframe 1100 8050 1MB Dual Disk 1375 8250 2 MB Dual Drive 1700 4040 Dual Drive 999 4022 Printer 630 8024-7 Hi-Speed Printer 1345 ATARI: ATARI 800 (16K) 599 ATARI 400 (16K) 339 810 Disk Drive 449 825 Printer 619 16K Ram Memory 89 850 Interface 159 MICROSOFT BASIC 69 VISICALC 159 815 Disk 1199 Letter Perfect 125 INTEC 32K Ram 129 Axlon 128K Ram Disk 550
OTHER HARDWARE for APPLE D.C. HAYES: Micromodem II 300 Smartmodem 239 MICROSOFT: Z80 Softcard 269 16K Ramcard 159 MOUNTAIN COMPUTER: CPS Multifunction 199 A/D + D/A Card 105 Expans. Chassis 595 Romplus 135 CALL FOR MORE PRICES. WE CARRY FULL LINE! CALIF. COMPUTER SYSTEMS: Centronics Par. Int. 115 A/D Conv. 105 Async. Serial Int. 135 Arith. Proc. /Disk 339 CALL FOR MORE PRICES. WE CARRY FULL LINE! VIDEO: 80 Col. Board 275 Switchplate 15 Enhancer I 105 Enhancer II 125 MORE OTHER HARDWARE for APPLE: SSM AIO 159 ABT Keypad 119 SSM Serial ASI0 115 Keyboard Co. Keypad 129 SSM Par. API0 99 Sunshine Joystick 40 SSM A488 IECE 425 Cisco Paddies 29 Novation APL. CAT 339 Prometheus Z Card 229 VERSAWRITER 249 SUP R MOD 29	NEC SPINWRITERS & MATRIX PRINTERS: 3530 Par./RO 1895 3510 Ser./RO 1895 7730 Par./RO 2445 7710 Ser./RO 2445 77D Sellum 2795 NEC PC8023A 675 QUME: 9/45 2149 630 R/O 2350 IDS: 445G 699 460G 875 560G 1075 Prism Color Printer 1699 TI: 810 Basic 1349 810 Serial 1429 MPI: 88G 579 99G 669 PRINTRONIX: P300 4500 P600 6150	XEROX: 820-1 System w/5" Dr. 2495 820-2 System w/8" Dr. 3145 WORDSTAR 419 CP/M OP. SYS. 175 SUPERBRAIN by INTERTEC: 64K/00 2695 64K/00 2995 DSS 10 Meg Emulator 725
PERSONAL SOFTWARE: Visicalc 3.3 159 VisiSite 199 Visiplot 149 Visiplot/Visitrend 199 Visiterm 125 VisiFax 169 Desktop Plan II 169 Visipak 559 MICROSOFT: APPLE Fortran (Z80) 149 APPLE Cobol (Z80) 499 TASC Basic Compiler 139 Olympic Decathlon 25 MICRO-PRO: Wordstar 245 Mail-Merge 99 Spell-Star 149 Data-Star 199 PCACHTREE: BIZ Packages, all 219 CPABIZ Packages, all 199	ADD-ON MEMORY CARDS & DISK DRIVES FOR APPLE MEMORY: Microsoft 16K Ram card 159 Saturn 32K Ramcard 199 Legend 128K Ram card 649 Prometheus 128K Ram. 550 Legend 64K Ram card 299 4116 Add-In Mem/16K 29 SVA 256K APL-Cache 1045 APPLE-COMPATIBLE FLOPPIES by MICRO-SCI: With Controller: No Controller: A35 Dir. Replacement 450 A35 415 A40 40-Track 489 A40 399 A70 70-Track 599 A70 499 APPLE-COMPATIBLE FLOPPIES by VISTA 40 Track 163KB 389 80 Track 326KB 559 160 Track 652KB 799 8" 800D Controller 549 V1000 Dual 8" V1000 Dual 8" 1600 Subsystem 1300 Dbl. Side SORRENTO VALLEY ASSOC: AMS 8000 Dual 8" 1945 AMS 5000 5 Mb. Drive 2995 AMS 8000 Dual 8" AMS 5010 10 Mb. Drive 3495 Dbl. Sd. 2595 Disk 2 + 2 Controller 359 ZVX4 Quad Controller 495	CRT TERMINALS Heath/Zenith Z19 725 Televideo 912C 685 Televideo 920C 775 Televideo 950C 985 Televideo 910 595 ADDS Viewpoint 595 ADDS R25 875 ADDS R40 1175 AMPCX Dialogue 80, 2-page, Detachable 995 IBM 3101-10 1199
WORD-PROCESSORS for APPLE: Wordstar CP/M 245 EZWriter Prof. Sys. 215 Magic Wand CP/M 229 MUSE Super Text 125 Executive Secretary 199 Wordpower 50 Letter Perfect 125 Hebrew II 55 Magic Window 85 SuperScribe II 110 Spellguard 235 Spellstar CP/M 149 Word Handler (Gives 80-Col. & lower case with no board!) 199	MONITORS & PLOTTERS ZENITH: 12" Green 125 13" Color 375 AMOEK: 12" Green 135 13" Color 375 WATANABE FLATBED PLOTTER: 4 Pen 1150 6 Pen 1400 STROBE CYLINDRICAL PLOTTER: Plotter 699 APPLE Intfc 75 HOUSTON INSTR FLATBED PLOTTER: DMP3 Smart Plotter 1299 6-Pen Adaptor 349	MODEMS CAT 159 DCAT 169 VENTEL 212-3, 1200BD 849 AUTOCAT 229 LEX-11 149 UNIVERSAL AUTO. ANS 220 UNIVERSAL DRIG/ANS 180 UNIVERSAL 1200BD 259
MORE OTHER SOFTWARE for APPLE: DB Master 179 DB Master for CORVUS 399 Data Factory 125 PFS 85 ASCII Express 3.3 55 PFS Report 85 Sorcim Super Calc 250 Spellguard CP/M 235 Howard Tax Preparer 82 125 Broderbund G. Ledger 335 Real Estate Analyzer 129 Broderbund Payroll 335 Ashton d Base II 495 dBase II Manual 50 Graph Trix 49	S100 CARDWARE SSM MB64 64K Mem. 775 HAYES S100 MODEM 349 CCS 64K Dynamic 429 DT Clock/Cal 135	FLOPPY DISCS 3M 5" Soft SSSD/bx 29 3M 8" Soft SSSD/bx 33 Elephant 5.25" Elephant 8" Soft SSSD/bx 25 Soft SSSD/bx 29

BONUS SOFTWARE SECTION!

Let us acquaint you with MESSAGE-MAKING SOFTWARE. Just place the disk in the APPLE, enter the text, and colorful, dynamic messages appear on the screens of TV sets connected to the computer. Use the software to broadcast messages on TV screens in schools, hospitals, factories, store window, exhibit booths, etc. The following program is our latest release:

SUPER MESSAGE: Creates messages in full-page "chunks". Each message allows statements of mixed typesizes, typesizes and colors, in mixed upper and lower case. Styles range from regular APPLE characters, up to double-size, double-width characters with a heavy, bold font. Six colors may be used for each different typesize. Vertical and horizontal centering are available, and word-wrap is automatic. Users can chain pages together to make multi-page messages. Pages can be advanced manually or automatically. Multi-page messages can be stored to disc or recalled instantly.
 REQUIRES 48K & ROM APPLESOFT..... \$50

APPLE PLOTS YOUR DATA & KEEPS YOUR RECORDS TOO! APPLE DATA GRAPH 2.1: Plots up to 3 superimposed curves on the Hi-res Screen both X & Y axes dimensioned. Each curve consists of up to 120 pieces of data. Graphs can be stored to disc and recalled immediately for updating. Up to 100 graphs can be stored on the same disc. Great for Stockmarket Charting, Business Management, and Classroom Instruction!
 REQUIRES 48K & ROM APPLESOFT..... \$35

APPLE RECORD MANAGER: Allows complex files to be brought into memory so that record searches and manipulations are instantaneous. Records within any file can contain up to 20 fields, with user-defined headings. Information can be string or numeric. Users can browse thru files using page-forward, page-backward or random-search commands. Records can easily be searched, altered or sorted at will. Files can be stored on the same drive as the master program, or on another, if a second drive is available.

Records of files can be printed, if desired. Additional modules coming are a STATISTICS INTERFACE, CHECKBOOK, MAILING LIST & DATA-ENTRY.
 REQUIRES 48K & ROM APPLESOFT..... \$40.

APPLE LITERATURE DATABASE: Allows rapid retrieval (via keywords) of references from total APPLE literature thru 1980, on 5.25" disk. Each entry in the data base consists of the article, author-name, periodical-name, date of issue, & page nos. The database is intended to support large magazine files which would require lengthy manual searching to recover information. Annual updates will be available.
 REQUIRES 48K & ROM APPLESOFT..... \$60

WORDPOWER: is a simple, powerful, low cost, line-oriented word-processor program. It offers a fast machine language FIND & REPLACE. Text can be listed to screen or printer, with or without line-numbers. Lower-case adaptors are supported. You can merge files, move groups of lines, and easily add, change, or delete lines. WORDPOWER can be used to create and maintain EXEC files. It can also be used as a rapid, unstructured, information-storage and retrieval system via its rapid search capabilities.
 REQUIRES 48K & ROM APPLESOFT..... \$50

LABEL MAKER: Allows users to quickly create address labels. A given label may be generated in any quantity from 1 to 32767. Space is allowed on labels for a personal and company name, but the space is automatically closed up if only a personal name is entered. Space is also allowed for foreign countries. The program can also generate labels for price-tags, part numbers and mail-messages such as "RUSH", "FRAGILE", etc. A self-incrementing feature allows theatre-tickets to be produced, with a date, and numbers running from a000 to z999. An editor is provided for editing labels prior to printing. All labels may be saved to disk for instant recall.
 REQUIRES 48K & ROM APPLESOFT..... \$35

Above software for APPLE DOS 3.2/3.3 only. Call for BONUSSES for other systems.
TO ORDER: Use phone or mail. We accept VISA, MASTERCARD, COD's, personal checks & money orders. Add 4% for credit card. Customer pays handling on COD orders. Foreign orders must be in American Dollars & include 10% for handling. Connecticut residents add 7.5% sales tax. Prices subject to change without notice.

CONN. INFO. SYSTEMS CO.
(203) 579-0472
218 Huntington Road, Bridgeport, CT 06608

Apple Graphics for Okidata Microline 80

by Charles F. Taylor, Jr.

Programs are given, in Apple/UCSD Pascal and 6502 assembly language, to dump the Apple high-resolution graphics screen(s) to an Okidata Microline 80 printer. This article should also be of interest to owners of the Epson MX-80 printer because graphics on the two printers are implemented similarly.

These Apple Graphics routines require:

Apple
Okidata Microline 80, 82A,
or 83A
Epson MX-80

Pascal version requires:

Apple Pascal system

When I purchased my Okidata Microline 80 dot matrix impact printer, I was attracted to its relatively low price, the 200,000,000-character print head warranty, and the flexible form handling (friction and pin feed). I really didn't consider its graphics, which were advertised as "TRS-80-compatible."

After I'd had the printer for a while, I decided to take another look at its graphics capabilities. The basic graphical unit for the Microline 80 is the graphics character. Each graphics character may be thought of as a 3-row by 2-column matrix, as depicted in figure 1.

The individual elements of the graphics character are numbered 1-6 as in figure 1. Each element of the character may be "on" (black) or "off" (white), which means that there are 2 to the 6th power, or 64, possible distinct graphics characters. An element that is "on" is represented by what appears (under magnification) to be a 3 by 3 matrix of dots. The total graphics character, then, is a 9-row by

6-column matrix. This is achieved with a 7-pin print head by making two passes for every line which contains a graphics character, advancing the paper slightly between passes.

With the printer set for 16.5 characters per inch on an 8-inch line, the horizontal resolution is 0.030 inches [0.77 millimeters]. At eight lines per inch, the vertical resolution is 0.042 inches [1.06 millimeters]. In other words, the smallest "dot" that can be printed is an element of a graphics character which is a rectangle 0.030 inches wide by 0.042 inches high.

Each graphics character is sent to the printer as a single byte with the high-order bit [bit 8] set [1]. Bit 7 may be either 0 or 1. Bits 1 through 6 are set [1] or clear [0] as the correspondingly numbered element of the graphics character is "on" or "off." (See figure 1 again.)

Software could be written to utilize these graphics characters directly. This would include, as a minimum, routines to set and clear individual elements of graphics characters and to draw straight lines between any two points. Because Applesoft BASIC and Apple/UCSD Pascal each provide these graphics primitives for use with the Apple high-resolution screen, a better approach is to develop a utility program to dump, point by point, the contents of the hi-res screen to the printer. This was the approach I took, first in Pascal, then in 6502 assembly language. [The latter version can be called from BASIC programs.]

The basic unit of Apple hi-res graphics is of course the "pixel" or dot. The hi-res screen is organized as a 192-row by 280-column matrix of individually addressable pixels. The display is bit-mapped; that is, there exists a mapping between each pixel on the screen and a bit somewhere in memory.

There were three principal problems to be resolved in designing the

program: the first problem was how to address the bit representation of each pixel in order to determine whether it is on or off. The second problem was to decide whether to print the screen image horizontally or vertically on the printer. Finally, a means had to be found to map six pixels to each graphics character.

Pascal Solution

The solution in Pascal is presented first because it is simpler. This is because of the existence of the SCREENBIT function, which is provided as part of the TURTLEGRAPHICS unit. SCREENBIT(x,y) is a Boolean function which returns the value TRUE if pixel

Figure 1: Microline 80 Graphics Character

1	2
3	4
5	6

Figure 2: Model of High Resolution Screen (6 x 6)

	0	1	2	3	4	5	
5	*						0
4	*	*					1
3	*		*				2
2	*			*			3
1	*				*		4
0	*					*	5
	0	1	2	3	4	5	

(x,y) is on (not black), and FALSE if it is off (black). This makes the first problem cited above almost trivial.

Since only a maximum of 132 characters can be printed on a line (16.5 characters per inch times 8 inches), and each character is two elements wide, the maximum number of pixels which can be presented on a printed line is 264. Because the Apple hi-res screen is 280 pixels wide, two choices are possible: (1) print the screen image vertically on the printer, 192 elements across and 280 down; or, (2) print the screen image horizontally, but print only 264 of the 280 columns. The former choice was made for the Pascal version and the latter for the 6502 assembly language version.

The Pascal program is shown in listing 1. The main program queries the user as to whether to print all or a specified portion of the screen. Procedure SETUP handles the details of turning on the printer and selecting the

proper print size and vertical spacing. Procedure TURNOFF turns the printer off again. The real work is done in the procedure called SCREENDUMP.

How the algorithm works can best be illustrated by example. Assume that the Apple hi-res screen consists of a 6-row by 6-column grid as shown in figure 2, and that an arbitrary pattern has been plotted on it. An "*" is used to indicate which grid elements are "on." In Pascal (as opposed to BASIC) the origin is at the lower left corner of the grid, so the numbers along the left side refer to the row of y-coordinates. The numbers along the right side will not be needed until later.

The Pascal program will reduce this grid to two lines of three graphics characters each. The first line will represent columns (x-coordinates) 0, 1, and 2 of the grid and the second line columns 3, 4, and 5. Remember that the image on the printer will be rotated

Graphics Characters	Screen Grids
Bit	Position
1	(0,0)
2	(0,1)
3	(1,0)
4	(1,1)
5	(2,0)
6	(2,1)

90 degrees, so that columns on the screen correspond to rows on the printer and *vice versa*.

The first graphics character of the first line will represent rows (y-coordinates) 0 and 1 of the columns 0, 1, and 2. We may imagine the grid of figure 1 superimposed on the grid of

```

Listing 1
PROGRAM PRINTSCREEN;

  (**S**)
  (* DUMPS ENTIRE PASCAL SCREEN *)
  (* TO OKIDATA MICROLINE 80 *)
  USES TURTLEGRAPHICS;

  VAR XMIN, XMAX, YMIN, YMAX : INTEGER;

  PROCEDURE SCREENDUMP(XMIN, XMAX, YMIN, YMAX : INTEGER);

    (* DUMPS PASCAL GRAPHICS TO *)
    (* OKIDATA MICROLINE 80 *)

    VAR H, I, J, K : INTEGER;
    LINE : ARRAY [1..96] OF CHAR;
    BIT : ARRAY [1..6] OF 0..1;
    OKI : TEXT;

    PROCEDURE SETUP;
    BEGIN
      (* OPEN PRINTER FILE *)
      REWRITE(OKI, 'PRINTER:');
      (* SET CENTRONICS CARD FOR 132 COLS *)
      WRITELN(OKI, CHR(9), '132N');
      (* SET PRINTER FOR 16.5 CPI & 8 LPI *)
      WRITELN(OKI, CHR(29), CHR(27), '8', CHR(27), '8');
      END; (* SETUP *)

    PROCEDURE TURNOFF;
    BEGIN
      (* RESETS OKIDATA *)

      WRITELN(OKI, CHR(30), CHR(27), 'B', CHR(27), 'A');
      END; (* TURNOFF *)

    FUNCTION GCHAR : CHAR;
    BEGIN
      (* RETURNS GRAPHICS CHARACTER *)
      (* DEFINED BY BIT ARRAY *)

      VAR NR : 0..255;
      POWER, I : INTEGER;

      BEGIN
        POWER := 1;
        NR := 128;
        FOR I := 1 TO 6 DO
          BEGIN
            NR := NR + POWER*BIT[I];
            POWER := 2*POWER;
          END;
      END;
    END;

  END;

```

```

Listing 1 (Continued)
  GCHAR := CHR(NR)
  END; (* GCHAR *)

  BEGIN (* PROCEDURE SCREENDUMP *)
  SETUP;
  FOR K := 1 TO 96 DO
    LINE[K] := CHR(128);
    I := XMIN;
    REPEAT
      J := YMIN;
      K := 1 + YMIN DIV 2;
      REPEAT
        FOR H := 1 TO 6 DO BIT[H] := 0;
        IF SCREENBIT(I, J) THEN BIT[1] := 1;
        IF SCREENBIT(I, J+1) THEN BIT[2] := 1;
        IF SCREENBIT(I+1, J) THEN BIT[3] := 1;
        IF SCREENBIT(I+1, J+1) THEN BIT[4] := 1;
        IF SCREENBIT(I+2, J) THEN BIT[5] := 1;
        IF SCREENBIT(I+2, J+1) THEN BIT[6] := 1;
        LINE[K] := GCHAR;
        K := K + 1;
        J := J + 2;
      UNTIL J+1 > YMAX;
      FOR K := 1 TO 96 DO
        BEGIN
          WRITE(OKI, LINE[K]);
          LINE[K] := CHR(128);
        END;
        WRITELN(OKI);
        I := I + 3;
      UNTIL I+2 > XMAX;
      TURNOFF;
    END; (* SCREENDUMP *)

  BEGIN (* MAIN PROGRAM *)
  REPEAT
    WRITE('FIRST COLUMN TO PRINT (0..279):');
    READLN(XMIN);
    UNTIL (XMIN) = 0 AND (XMIN <= 279);
  REPEAT
    WRITE('LAST COLUMN TO PRINT (XMIN..279):');
    READLN(XMAX);
    UNTIL (XMAX) = XMIN AND (XMAX <= 279);
  REPEAT
    WRITE('FIRST ROW TO PRINT (0..191):');
    READLN(YMIN);
    UNTIL (YMIN) = 0 AND (YMIN <= 191);
  REPEAT
    WRITE('LAST ROW TO PRINT (YMIN..191):');
    READLN(YMAX);
    UNTIL (YMAX) = YMIN AND (YMAX <= 191);
    SCREENDUMP(XMIN, XMAX, YMIN, YMAX);
  END.

```

figure 2, with figure 1 rotated 90 degrees (counter-clockwise). Thus bits of the graphics characters correspond to screen grids as shown in table 1. In this case bits 3, 4, 5, and 6 will be 0 and bits 1 and 2 will be 1. Bit 7 will (arbitrarily) be 0, and bit 8 will be 1, as discussed earlier. The resulting graphics character is, therefore, binary 10000011 [\$83 or decimal 131].

The next graphics character of the line will be constructed from grid elements {0,2}, {1,2}, {2,2}, {0,3}, {1,3}, and {2,3}, which correspond to, respectively, bits 1, 3, 5, 2, 4, and 6. In this case bits 1, 2, and 6 will be 1 and bits 3, 4, and 5 will be 0. The resulting graphics character is binary 10100011 [\$A3, or decimal 163]. The third and final graphics character of the line will be binary 10000111, [\$87, or decimal 135].

At this point the program has constructed the first full line of graphics characters. Using the CHR function, the computed decimal values have been converted to their character equivalents and are stored in the array LINE. Now that LINE is full, it is sent to the printer, one character at a time, and is followed by the usual carriage return and line feed (WRITELN).

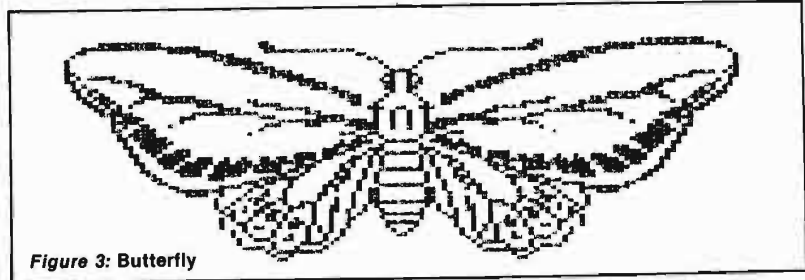


Figure 3: Butterfly

The program then constructs the next line of graphics characters from columns 3, 4, and 5 of the screen grid. These characters will be, in decimal notation, 152, 129, and 128, respectively.

The procedure described above is carried out by the procedure SCREEN-DUMP. The function GCHAR uses simple arithmetic to convert the binary representation of the character to decimal. Then it uses CHR to convert the decimal value to a character. Experienced Pascal programmers may notice that I could have accomplished the binary-to-character conversion directly using a free-union variant record. That technique would have been faster and more efficient, but less clear. Readers who wish to pursue this topic should refer to an article by David

Casseres of Apple Computer Inc., which appeared in the October 1981 issue of *BYTE* magazine.

Figure 3 is an example of output produced by this program. The butterfly image was created on the screen by a demonstration program furnished with the Apple Pascal system and then printed on the Microline 80 by this program.

Assembly Language Solution

The most difficult part of the 6502 assembly language solution was to develop an algorithm to step through memory, addressing of each pixel's bit representation in the proper sequence. (Recall that this was done for us in Pascal by SCREENBIT.) The task is complicated by the fact that, for various reasons, Apple chose to represent the hi-res screen in memory in what appears to be a rather peculiar sequence. The mapping used is documented in the *Apple II Reference Manual* and was the subject of a 1978 article in *MICRO* (7:43) by Andrew H. Eliason. Rather than reiterate the details here, I have chosen to present a short Applesoft BASIC program (listing 2) which prints out the beginning address (in decimal and in hex) of each of the 192 rows of hi-res screen 1. (To get the corresponding values for screen 2, change line 100.) The 280 pixels of each row are represented by seven bits of each of the 40 bytes beginning at the location given. The program prints a screen, then prompts the user to press the space bar before running another screen.

As mentioned above, I decided to represent the screen horizontally on the printer in this version (which considerably simplifies the arithmetic). This means that only 264 columns of the hi-res screen could be printed. The first 264 were arbitrarily selected.

The 6502 assembly language program is shown as listing 3. Instructions for its use are contained in the program's introductory comments. The printer interface I used was the Apple

Decision Systems

Decision Systems
P.O. Box 13006
Denton, TX 76203

SOFTWARE FOR THE APPLE II*

ISAM-DS is an integrated set of Applesoft routines that gives indexed file capabilities to your **BASIC** programs. Retrieve by key, partial key or sequentially. Space from deleted records is automatically reused. Capabilities and performance that match products costing twice as much.
\$50 Disk, Applesoft.

PBASIC-DS is a sophisticated preprocessor for structured **BASIC**. Use advanced logic constructs such as **IF...ELSE...**, **CASE**, **SELECT**, and many more. Develop programs for Integer or Applesoft. Enjoy the power of structured logic at a fraction of the cost of **PASCAL**.
\$35. Disk, Applesoft (48K, ROM or Language Card).

DSA-DS is a dis-assembler for 6502 code. Now you can easily dis-assemble any machine language program for the Apple and use the dis-assembled code directly as input to your assembler. Dis-assembles instructions and data. Produces code compatible with the S-C Assembler (version 4.0), Apple's Toolkit assembler and others.
\$25 Disk, Applesoft (32K, ROM or Language Card).

FORM-DS is a complete system for the definition of input and output forms. **FORM-DS** supplies the automatic checking of numeric input for acceptable range of values, automatic formatting of numeric output, and many more features.
\$25 Disk, Applesoft (32K, ROM or Language Card).

UTIL-DS is a set of routines for use with Applesoft to format numeric output, selectively clear variables (Applesoft's **CLEAR** gets everything), improve error handling, and interface machine language with Applesoft programs. Includes a special load routine for placing machine language routines underneath Applesoft programs.
\$25 Disk, Applesoft.

SPEED-DS is a routine to modify the statement linkage in an Applesoft program to speed its execution, improvements of 5-20% are common. As a bonus, **SPEED-DS** includes machine language routines to speed string handling and reduce the need for garbage clean-up. Author: Lee Meador.
\$15 Disk, Applesoft (32K, ROM or Language Card).

(Add \$4.00 for Foreign Mail)

*Apple II is a registered trademark of the Apple Computer Co.

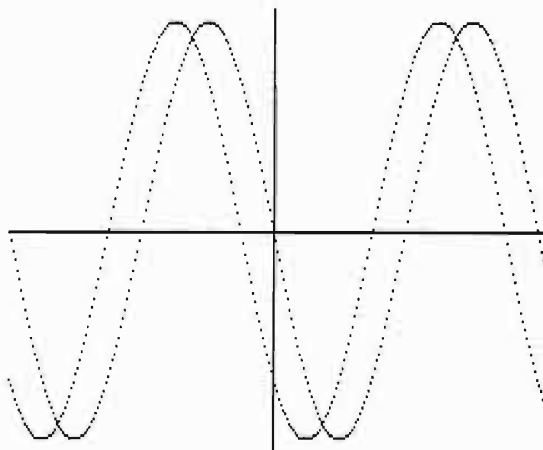
Listing 2

```

10 REM PROGRAM TO DEMONSTRATE
20 REM THE SEQUENCE OF STORAGE
30 REM LOCATIONS USED BY THE
40 REM APPLE HI-RES SCREEN
50 REM
60 REM BY C.F. TAYLOR, JR.
70 REM JULY 24, 1981
80 REM
90 HOME
95 DIGIT$ = "0123456789ABCDEF"
100 B0 = 8192: REM FOR SCREEN 1
110 SLIN = 0: REM SCREEN LINE NR
120 FOR I = 1 TO 3
130 B1 = B0
140 FOR J = 1 TO 8
150 CLIN = B1
160 FOR K = 1 TO 8
170 GOSUB 500: REM CONVERT TO HEX
180 PRINT SLIN, CLIN, HL$
190 SLIN = SLIN + 1
200 CLIN = CLIN + 1024
210 IF SLIN = INT (SLIN / 23) * 23 ( ) 0 THEN GOTO 240
220 PRINT "PRESS (SPACE) TO CONTINUE";
230 GET A$: PRINT
240 NEXT K
250 B1 = B1 + 128
260 NEXT J
270 B0 = B0 + 40
280 NEXT I
290 END
500 X = CLIN: REM SUBROUTINE CONVERT TO HEX
510 HL$ = ""
520 HL$ = MID$ (DIGIT$, X - INT (X / 16) * 16 + 1, 1) + HL$
530 X = INT (X / 16)
540 IF X ( ) 0 THEN GOTO 520
550 HL$ = "$" + HL$
560 RETURN

```

Figure 4: Sine Curves



Centronics Parallel Interface; some modifications will likely be required for use with other interface cards. What is critical is that bit 8 must be controllable (high for graphics, low for text). Some interfaces may not use bit 8 at all, or may force it low. The Epson interface board has bit 8 wired to ground, but a jumper is provided for changing this. If you do modify the setting of this jumper, however, you will have to make some other provision for forcing bit 8 low for text. My recommendation is to replace the jumper

with a single-pole, double-throw switch. This is, in effect, what I have done to my Apple Centronics Interface card.

How the assembly language program works can also be illustrated by example. Refer again to figure 2. This time we will use the row numbers (y-coordinates) along the right edge, recalling that BASIC refers to the upper left corner as [0,0]. This time rows 0, 1, and 2 will be used to construct the first line of graphics characters and rows 3, 4, and 5 the second line.

The first graphics character will therefore represent columns 0 and 1 of rows 0, 1, and 2. We may imagine the grid of figure 1 superimposed on the grid of figure 2, but this time without rotation. Thus bits 1-6 of the graphics character will represent, respectively, the screen grid positions (0,0), (1,0), (0,1), (1,1), (0,2), and (1,2). Bits 1, 3, 4, and 5 will be 1 and bits 2 and 6 will be 0. As before, bit 7 will be 0 and bit 8 will be 1. Therefore, the first graphics character is binary 10011101 (\$9D, or decimal 157).

For the second graphics character, bits 1-6 correspond, respectively, to coordinates (2,0), (3,0), (2,1), (3,1), (2,2), and (3,2). Bit 5 is 1 and bits 1, 2, 3, 4, and 6 are 0. This translates to binary 10010000 (\$90, or decimal 144). The third and last graphics character of the first line is binary 10000000 (\$80, or decimal 128). The decimal values of the three graphics characters of the second line are 149, 130, and 164.

The algorithm illustrated in the BASIC program of listing 2 is used to find the beginning of each of three consecutive rows of the screen in memory. The bytes representing the pixels of these lines are then transferred to working buffers. (Only 38 bytes out of 40 are used because only 264 out of 280 columns are plotted.) The subroutines DUMP and DUMPY then extract the appropriate bits from the buffers and rotate them into a page zero location called CHAR. From there each is sent to the printer.

Figure 4 shows a typical plot of two out-of-phase sine curves. More sophisticated plots (3-D, etc.) are of course possible; anything you can put on the screen, you can print! The one limitation is color since the printer only prints black and white!

Execution time for the assembly language version is typically about six minutes. The Pascal version takes about 2.5 times as long to print a full screen.

Although the programs presented here were reasonably involved to write, they are simple to use. Best of all, they transform a fairly unsophisticated graphics capability on an inexpensive printer into a powerful graphics tool, rivaling printers costing several times as much.

The author may be contacted at 587F Sampson Lane, Monterey, CA 93940.

Listing 3

```

0010      ;HI-RES SCREEN DUMP
0020      ;APPLE II PLUS 70
0030      ;OKIDATA MICROLINE 80 VIA
0040      ;APPLE CENTRONICS PARALLEL INTERFACE
0050      ;
0060      ;WRITTEN BY C. F. TAYLOR, JR.
0070      ;24 JULY 1981
0080      ;
0090      ;INSTRUCTIONS:
0100      ;
0110      ; SET HIMEM: 37760 BEFORE LOADING
0120      ; CALL 37888 FOR HI-RES PAGE 1
0130      ; CALL 37904 FOR HI-RES PAGE 2
0140      .BA #9400
0150      .DS
0160      .DE #638      ;LOCATIONS USED
0170      .DE #688      ;PRINTER INTERFACE
0180      .DE #488      ;ROM
0190      .DE #588
0200      .DE #03EA      ;DOS RE-ENTRY POINT
0210      .DE #06      ;PAGE 0 LOCATIONS
0220      .DE #08
0230      .DE #E3
0240      .DE #C102      ;PRINTER DRIVER
0250      ;
0260      ;ENTRY POINT FOR HI-RES PAGE 1
0270      ;
9400- A9 20 0280 PAGE1 LDA #*20      ;INITIALIZE POINTERS
9402- 8D 97 95 0290 STA BASE+1
9405- 8D 99 95 0300 STA SAVBAS+1
9408- A9 40 0310 LDA #*40
940A- 8D 9D 95 0320 STA CURLIM+1
940D- 4C 1D 94 0330 JMP START
0340      ;
0350      ;ENTRY POINT FOR HI-RES PAGE 2
0360      ;
9410- A9 40 0370 PAGE2 LDA #*40      ;INITIALIZE POINTERS
9412- 8D 97 95 0380 STA BASE+1
9415- 8D 99 95 0390 STA SAVBAS+1
9418- A9 60 0400 LDA #*60
941A- 8D 9D 95 0410 STA CURLIM+1
0420      ;
941D- A9 00 0430 START LDA #*00      ;COMMON POINTER VALUES
941F- 8D 96 95 0440 STA BASE
9422- 8D 98 95 0450 STA SAVBAS
9425- 8D 9B 95 0460 STA SLIN
9428- 8D 9C 95 0470 STA CURLIM
942B- 20 9D 94 0480 JSR SETUP      ;INITIALIZE PRINTER
942E- A9 02 0490 LDA #2      ;INITIALIZE BUFFER LINE NR
9430- 8D 9A 95 0500 STA LINE
0510      ;
9433- AD 96 95 0520 LOOPA LDA BASE      ;CLIN := BASE
9436- 85 06 0530 STA *CLIN
9438- AD 97 95 0540 LDA BASE+1
943B- 85 07 0550 STA *CLIN+1
0560      ;
943D- 20 C4 94 0570 LOOPB JSR BUFLIN ;INCREMENT BUFFER LINE NR
9440- EE 9B 95 0580 INC SLIN ;AND SCREEN LINE NR
9443- A0 25 0590 LDY #37 ;TRANSFER LINE TO BUFFER
9445- B1 06 0600 B1 LDA (CLIN),Y
9447- 91 08 0610 STA (BUFF),Y
9449- 88 0620 DEY
944A- 10 F9 0630 BPL B1
944C- AD 9A 95 0640 LDA LINE      ;TIME TO DUMP BUFFER?
944F- C9 02 0650 CMP #2
9451- D0 03 0660 BNE CONT
9453- 20 0A 95 0670 JSR DUMP
9456- A5 07 0680 CONT LDA *CLIN+1

```

Listing 3 (Continued)

```

9458- 18 0690 CLC
9459- 69 04 0700 ADC #*4      ;INCREMENT MEMORY LOCATION
945B- 85 07 0710 STA *CLIN+1
945D- CD 9D 95 0720 CMP CURLIM+1 ;TIME TO ADJUST BASE?
9460- 90 DB 0730 BCC LOOPB ;NO
9462- AD 9B 95 0740 LDA SLIN ;SCREEN LINE NR
9465- C9 40 0750 CMP #64 ;READY TO SHIFT?
9467- F0 06 0760 BEQ SC0 ;YES
9469- C9 00 0770 CMP #128
946B- F0 02 0780 BEQ SC0 ;YES
946D- D0 15 0790 BNE SC1 ;NO
946F- AD 96 95 0800 SC0 LDA SAVBAS
9472- 18 0810 CLC
9473- 69 28 0820 ADC #*28 ;SHIFT BASE
9475- 8D 98 95 0830 STA SAVBAS
9478- 8D 96 95 0840 STA BASE
947B- AD 99 95 0850 LDA SAVBAS+1
947E- 8D 97 95 0860 STA BASE+1
9481- 4C 33 94 0870 JMP LOOPA
9484- C9 D0 0880 SC1 CMP #192 ;DONE?
9486- F0 14 0890 BEQ EXIT ;YES
9488- 18 0900 CLC ;ADJUST BASE
9489- AD 96 95 0910 LDA BASE
948C- 69 80 0920 ADC #*80
948E- 8D 96 95 0930 STA BASE
9491- AD 97 95 0940 LDA BASE+1
9494- 69 00 0950 ADC #0 ;ADD IN CARRY
9496- 8D 97 95 0960 STA BASE+1
9499- 4C 33 94 0970 JMP LOOPA
949C- 60 0980 EXIT RTS ;DONE
0990      ;
1000      ;SUBROUTINES FOLLOW
1010      ;
1020      ;FIRST SETUP DRIVER
949D- A2 C1 1030 LDX #*C1 ;FOR SLOT 1
949F- A9 09 1040 LDA #*09 ;INITIALIZE DRIVER
94A1- 9D 86 06 1050 STA FLAGS,X ;VIDEO OFF
94A4- A9 FF 1060 LDA #*FF
94A6- 9D 88 04 1070 STA PWIDTH,X ;PRINT WIDTH
94A9- 9D 38 06 1080 STA ESCHR,X ;ESCAPE CHAR
94AC- A9 00 1090 LDA #0
94AE- 9D 88 05 1100 STA MODE,X ;CLEAR 'AFTER ESC' MODE
94B1- 20 EA 03 1110 JSR DOS ;REPLACE WITH 3 NOP'S
1120      ;FOR CASSETTE SYSTEM
1140      ;NOW SETUP PRINTER
94B4- A9 1D 1150 LDA #*1D ;SET 16.5 CPI ON PRINTER
94B6- 20 02 C1 1160 JSR DRIVER
94B9- A9 1B 1170 LDA #*1B ;SET B LINES
94BB- 20 02 C1 1180 JSR DRIVER ;PER INCH
94BE- A9 38 1190 LDA #*38 ;VERTICAL SPACING
94C0- 20 02 C1 1200 JSR DRIVER ;ON PRINTER
94C3- 60 1210 RTS
1220      ;
94C4- EE 9A 95 1230 BUFLIN INC LINE
94C7- AD 9A 95 1240 LDA LINE
94CA- C9 03 1250 CMP #3
94CC- D0 0E 1260 BNE BL1
94CE- A9 81 1270 LDA #L,LINE0 ;SET BUFFER LINE 0
94D0- 85 08 1280 STA *BUFF
94D2- A9 93 1290 LDA #H,LINE0
94D4- 85 09 1300 STA *BUFF+1
94D6- A9 00 1310 LDA #0
94D8- 8D 9A 95 1320 STA LINE
94DB- 60 1330 RTS
94DC- C9 01 1340 BL1 CMP #1 ;SET BUFFER LINE 1
94DE- D0 09 1350 BNE BL2
94E0- A9 A7 1360 LDA #L,LINE1
94E2- 85 08 1370 STA *BUFF

```

Listing 3 (Continued)

94E4-	A9 93	1380	LDA #H,LINE1	
94E6-	85 09	1390	STA *BUFF+1	
94E8-	60	1400	RTS	
94E9-	A9 CD	1410	LDA #L,LINE2 :SET BUFFER LINE 2	
94EB-	85 08	1420	STA *BUFF	
94ED-	A9 93	1430	LDA #H,LINE2	
94EF-	85 09	1440	STA *BUFF+1	
94F1-	60	1450	RTS	
		1460	:	
94F2-	AD 9B 95	1470	LDA SLIN :CHECK FOR SHIFT OF BASE	
94F5-	C9 40	1480	CMP #64	
94F7-	D0 10	1490	BNE NX1	
94F9-	C9 80	1500	CMP #128	
94FB-	D0 0C	1510	BNE NX1	
94FD-	AD 9B 95	1520	LDA SAVBAS	
9500-	18	1530	CLC	
9501-	69 28	1540	ADC #*28	
9503-	8D 98 95	1550	STA SAVBAS	
9506-	8D 98 95	1560	STA BASE	
9509-	60	1570	RTS	
		1580	:	
950A-	A2 00	1590	LDX #0 :DUMP BUFFERS TO PRINTER	
950C-	A9 00	1600	LDA #0	
950E-	85 E3	1610	STA *CHAR	
9510-	A0 02	1620	LDY #2	
9512-	20 37 95	1630	JSR DUMPY :Y+1 CHARS TO PRINTER	
9515-	20 68 95	1640	JSR TRANS :TRANSITION TO NEXT BYTE	
9518-	EB	1650	INX	
9519-	A0 02	1660	LDY #2	
951B-	20 37 95	1670	JSR DUMPY :REST OF BYTE	
951E-	EB	1680	INX	
951F-	E0 24	1690	CPX #36 :DONE?	
9521-	30 E9	1700	BMI DUMP1	
9523-	A0 02	1710	LDY #2	:FINISH COLS 127-132
9525-	20 37 95	1720	JSR DUMPY	
9528-	20 68 95	1730	JSR TRANS	
952B-	EB	1740	INX	
952C-	A0 01	1750	LDY #1	
952E-	20 37 95	1760	JSR DUMPY	
9531-	A9 0D	1770	LDA #*0D :CARRIAGE RETURN	
9533-	20 02 C1	1780	JSR DRIVER	
9536-	60	1790	RTS	
		1800	:	
9537-	7E 81 93	1810	ROR LINE0, X :Y+1 BYTES TO PRINTER	
953A-	66 E3	1820	ROR *CHAR :BIT 1	
953C-	7E 81 93	1830	ROR LINE0, X	
953F-	66 E3	1840	ROR *CHAR :BIT 2	
9541-	7E A7 93	1850	ROR LINE1, X	
9544-	66 E3	1860	ROR *CHAR :BIT 3	
9546-	7E A7 93	1870	ROR LINE1, X	
9549-	66 E3	1880	ROR *CHAR :BIT 4	
954B-	7E CD 93	1890	ROR LINE2, X	
954E-	66 E3	1900	ROR *CHAR :BIT 5	
9550-	7E CD 93	1910	ROR LINE2, X	
9553-	66 E3	1920	ROR *CHAR :BIT 6	
9555-	18	1930	CLC	
9556-	66 E3	1940	ROR *CHAR :BIT 7 = 0	
9558-	38	1950	SEC	
9559-	66 E3	1960	ROR *CHAR :BIT 8 = 1	
955B-	A5 E3	1970	LDA *CHAR	
955D-	20 02 C1	1980	JSR DRIVER :PRINT	
9560-	A9 00	1990	LDA #0	
9562-	85 E3	2000	STA *CHAR	
9564-	88	2010	DEY	
9565-	10 D0	2020	BPL DUMPY	
9567-	60	2030	RTS	
9568-	7E 81 93	2040	ROR LINE0, X :FINISH BYTE	
		2050	:AND START NEXT	

Listing 3 (Continued)

956B-	66 E3	2060	ROR *CHAR	
956D-	7E 82 93	2070	ROR LINE0+1, X	
9570-	66 E3	2080	ROR *CHAR	
9572-	7E A7 93	2090	ROR LINE1, X	
9575-	66 E3	2100	ROR *CHAR	
9577-	7E A8 93	2110	ROR LINE1+1, X	
957A-	66 E3	2120	ROR *CHAR	
957C-	7E CD 93	2130	ROR LINE2, X	
957F-	66 E3	2140	ROR *CHAR	
9581-	7E CE 93	2150	ROR LINE2+1, X	
9584-	66 E3	2160	ROR *CHAR	
9586-	18	2170	CLC	
9587-	66 E3	2180	ROR *CHAR :BIT 7 = 0	
9589-	38	2190	SEC	
958A-	66 E3	2200	ROR *CHAR :BIT 8 = 1	
958C-	A5 E3	2210	LDA *CHAR	
958E-	20 02 C1	2220	JSR DRIVER :SEND TO PRINTER	
9591-	A9 00	2230	LDA #0	
9593-	85 E3	2240	STA *CHAR	
9595-	60	2250	RTS	
9596-		2260	BASE .DS 2	
9598-		2270	SAVBAS .DS 2	
959A-		2280	LINE .DS 1	
959B-		2290	SLIN .DS 1	
959C-		2300	CURLIM .DS 2	
		2310	.BA #9381	
9381-		2320	LINE0 .DS 38	
93A7-		2330	LINE1 .DS 38	
93CD-		2340	LINE2 .DS 38	
		2350	.EN	

ACRO

SOFTWARE FOR THE HARDCORE

THE PROFESSIONAL'S CHOICE FORTH — A Tool for Craftsmen!

It has been said that if Chippendale had made programs he would have used FORTH as his tool. If you want to learn how to program, use a teaching language—PASCAL or BASIC. If you know how to program, use a language designed for craftsmen—FORTH.

FORTH Systems

For all FLEX systems: 6800 & 6809. Specify 5" or 8" diskette and hardware configuration. For standalone versions, write or call.

- ** tFORTH—extended fig-FORTH (1 disk) \$100 (\$15)
- ** tFORTH+ —extended more! (3 5" or 2 8" disks) \$250 (\$25)
- tFORTH+ includes 2nd screen editor, assembler, extended data types and utility vocabularies. GOING FORTH CAI course on FORTH, games, and debugging aids.

TRS-80 COLORFORTH — 10K ROM Pack

Full screen editor. Will work on 4K, 16K, or 32K systems \$110 (\$20). Disk versions available.

Applications Programs

- ** firmFORTH 6809 tFORTH+ only \$350 (\$10)
- For target compilations to rommable code. Deletes unused code and unneeded dictionary heads. Requires tFORTH+.
- ** TINY PASCAL compiler in FORTH. 6800/09 \$75 (\$20)
- ** FORTH PROGRAMMING AIDS: Extensive debugging, compiling, and program analysis tools. \$150 (\$10)

Manuals alone, price in (.). Add \$5/system for shipping. \$12 for foreign air.

Talbot Microsystems

1927 Curtis Ave., Redondo Beach, CA 90278
(213) 376-9941

(TM) tFORTH, COLORFORTH and firmFORTH are trademarks of Talbot Microsystems.
(TM) FLEX is a trademark of Technical Systems Consultants.

**IF YOU CONSIDER THE PRICE, COGNIVOX AT \$249 IS THE BEST BUY IN VOICE I/O.
IF YOU CONSIDER THE PERFORMANCE, YOU WILL BUY IT.
BECAUSE COGNIVOX OUTPERFORMS ALL VOICE INPUT OR OUTPUT PERIPHERALS FOR THE APPLE II.
AT ANY PRICE.**

COGNIVOX VIO-1003 is a state-of-the-art Speech Recognition and voice output peripheral for the APPLE II computer. It enables the APPLE to recognize words or short phrases spoken by the user and it can talk with natural sounding voice.

SPEECH RECOGNITION

COGNIVOX recognizes words (such as "one," "enter," etc.) or short phrases (like "total amount," "net weight," etc.) from a vocabulary of 32 entries. The vocabulary entries are chosen by the user to suit his application. Then COGNIVOX is "trained" to the vocabulary by repeating each entry three times into the microphone under the prompting of the system.

During training, COGNIVOX analyzes the voice of the user and compresses all the important information in each entry into 48 bytes of data called the reference pattern. When training is complete, words spoken in the microphone are similarly analyzed and the resulting 48-bit pattern is compared with all the reference patterns to obtain a best match.

The power of COGNIVOX is derived from proprietary pattern generation and pattern matching algorithms that allow quick and easy training and give a recognition accuracy equal to much more expensive units.

Vocabularies larger than 32 words are possible by swapping reference patterns in memory using a key word, for example, "change vocabulary." Or the swap can be performed under program control.

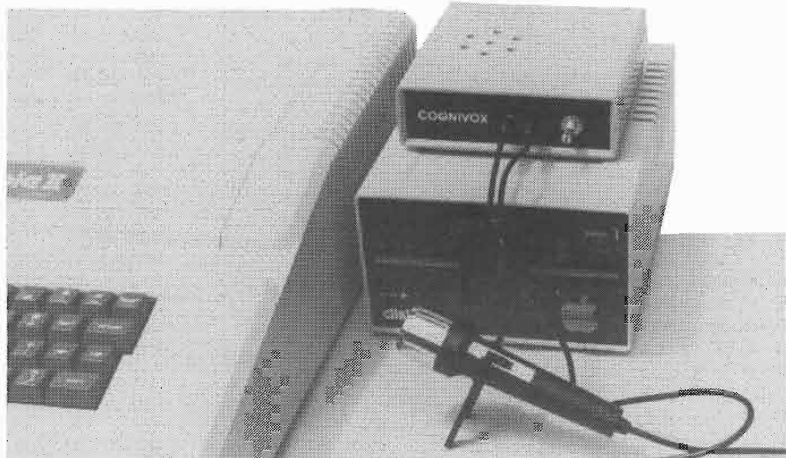
VOICE OUTPUT

COGNIVOX can talk with a vocabulary of 32 words or short phrases. No restrictions are placed on the vocabulary which can be programmed simply by saying the words into the microphone. The speech waveform is then digitized using a data compression method and stored in memory.

When voice output is desired, the selected word or phrase is reconstructed and played back using a built-in speaker/amplifier. A jack is also provided that allows connection to external amplifiers or speaker.

This method of voice output offers two very important advantages: First, the user has full control over the selection of the vocabulary and the type and tone of voice. Second, the voice output is naturally sounding human speech which is pleasant and easy to understand. These features are not available in most other voice output devices in the market.

The voice output and speech recognition vocabularies are independent of each other and can be different. Thus it is possible to establish a dialog with the computer.



USING COGNIVOX

COGNIVOX is designed for extreme ease of use. It is a complete system, fully assembled and tested, including hardware in an instrument case, microphone, power supply, cassette with software and user manual. It plugs into the game I/O port in the APPLE and does not use up the valuable peripheral slots.

Software provided with COGNIVOX include demonstration programs and two voice operated, talking video games. All programs are unprotected so that the user can examine and modify them.

An optional diskette for DOS 3.3 includes all cassette software plus disk facilities to store and retrieve vocabularies on disk.

Adding voice I/O to your own programs is very simple. A statement in BASIC is all that is needed to either recognize or say a word. Complete instructions on how to add voice to your programs are given in the manual.

APPLICATIONS

COGNIVOX adds a whole new dimension to man-computer interaction. It can be used for data and command entry when hands and/or eyes are busy. As an educational tool. As an aid to handicapped. As sound effects generator. As a telephone answering machine. As a talking calculator, or talking clock.

The list is endless. With a BSR home controller interface it can be used to control by voice appliances and lights around the house. With an IEEE 488 interface card it can be used to control by voice instruments, plotters, test systems. And all these devices could talk back, saying their readings, alarm conditions, even their name. Finally, COGNIVOX is a super toy, a fascinating device to play with. Imagine an adventure game that talks to warn you of danger and listens to your commands!

SPECIFICATIONS

- Recognizer type:**
Isolated word, speaker dependent.
- Vocabulary size:**
32 words or short phrases for both recognition and voice response.
- Dialog capability:**
Recognition and response vocabularies can be different.
- Word Duration**
Greater than 150 ms and less than 3 seconds.
- Silence gap between words:**
150 ms minimum.
- Training required:**
Must pronounce vocabulary 3 times to train recognizer. Allows words to be individually retrained.
- Recognition accuracy:**
Up to 98%. Recognition accuracy depends on speaker experience and choice of vocabulary.
- Type of voice output:**
Digital recording of user voice.
- Audio output:**
130 mW
- Frequency response:**
100 to 3200 Hz.
- Power consumption:**
120 mW during recognition, 350 mW maximum during speech output.
- Power supply:**
9V DC, 300 mA, unregulated.
- Dimensions:**
5" x 6" x 1.25"
- Memory requirements:**
Approx. 4K bytes for program and tables. 1.5K bytes per sec. of speech for storage of voice response vocabulary (Approx. 700 bytes per word).

ORDER COGNIVOX TODAY

To order COGNIVOX by phone, call us at (805) 685-1854, 9AM to 4PM PST. Monday through Friday, and charge it to your MASTERCARD or VISA. Or you can order by mail. Send us a check or money order for \$249 plus \$5 shipping and handling (CA residents add 6% tax). Software on diskette (DOS-3.3) order part # DSK-3.3, \$19. Foreign orders welcome, add 10% for air mail shipping and handling. COGNIVOX is backed by a 120 day warranty against manufacturing defects.

VOICETEK

Dept G, P.O. Box 388
Goleta, CA 93116

ALSO AVAILABLE for COMMODORE COMPUTERS and the AIM-65.
CALL or WRITE for MORE INFO.

MICRO™

Microbes and Updates

Jim Sherman of Huntsville, AL, called in with these corrections to "Saucer Launch" by Mike Dougherty (42:53):

On page 59 the listing is out of order. Lines 108E through 1092 belong at the bottom of the page (after line 108C). On page 60, line 10D2 should read: F0 06 BEQ NOXP.

Here are a few corrections to the RUNZMENU article by Frank Shyjka (45:67):

On page 68 in the far right column, the third line of BCDF should read:

85 39 20 51 A8 A9 **8C** 8D 6D
BA8D should read:

D2 D5 CE DA CD C5 CE D5 **BF**

Erken Heinzjosef from West Germany wrote in with this update:

In MICRO 43 you published a Call Routine for the Superboard. Although it is very good, I have found a simpler way. My routine is only nine bytes long but it has a disadvantage: you cannot use hex addresses. But you can use

labels! See listing 1 for the machine-language routine and listing 2 for the equivalent BASIC load. The syntax must be

Z (or any alpha) =
USR (any argument) 65030

or

Z (or any alpha) =
USR (any argument) SC

The label "SCREEN CLEAR" gives syntax error as BASIC thinks it should be SQR.

65030 = hex FE06 = Screen clear in the C 1 S Monitor ROM from Aardvark. If you use labels, don't forget to define the label:

10 SC = 65030
20 WARMSTART = 0
30 X = USR(X) SC
40 X = USR(X) WARMSTART

You have to set the USR Vector at first by POKE 11,64: POKE 12,2. My BASIC load does it, but after a BREAK you have to reset the vector.

Listing 1

```
10 0000 ;CALL ROUTINE FOR SUPERBOARD
20 0000 ;
30 0000 ;H. J. Erken, West Germany
40 0000 ;
50 0000 ;
60 0000 ;Addresses can be decimal values or labels
70 0000 ;
80 0000 ;To use, first set up the USR Vector by
90 0000 ;POKE 11,64:POKE12,2
100 0000 ;
110 0000 ;
120 0240 * = $0240
130 0240 ;
140 0240 20ADAA JSR #A0AD ;Evaluate any expression
150 0243 200BB4 JSR #B40B ;Convert floating to fix
160 0246 6C1100 JMP (#11) ;Hex value of expression is
170 0249 ; stored in #11/#12
```

Listing 2

```
10 REM CALL FOR SUPERBOARD
20 REM H.J. ERKEN, WEST GERMANY
30 REM
40 FOR X = 576 TO 584
50 READ A: POKE X,A: NEXT
60 POKE 11,64: POKE 12,2: REM INIT OF USR FUNCTION
70 NEW
80 DATA 32, 173, 170, 32, 8, 180, 108, 17, 0
```

MICRO

Saturn Navigator now A2-3D1 licensed!



Yes, SubLOGIC's **A2-3DA Saturn Navigator** is now available as a complete package, incorporating all of the graphic routines necessary for a successful flight from earth to Saturn. If you've been looking for a realistic 3D space flight simulation this is the one for you. And of course, our original **A2-3D/A Saturn Navigator** is still available to owners of the A2-3D1 Graphics package. Saturn Navigator, a sophisticated adventure simulation that takes only a short time to learn. It just may take a lifetime to master.

For the Apple II plus or Apple II with Applesoft ROM. 48K. \$34.95.

See your dealer . . .

or for direct orders add \$1.50 and specify UPS or first class mail. Illinois residents add 5% sales tax. Visa and MasterCard accepted.

subLOGIC

Communications Corp.
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex: 206995

"Apple" is the registered trademark of Apple Computer Inc.

The Single Life

By Brad Rinehart

In my previous columns I have explained various HDE Disk BASIC features. This month I will cover some of the most unusual commands, as well as file handling techniques.

HDE Disk BASIC's more powerful commands include INSTR (in string), VARPTR (variable pointer), EXEC, MERGE, GET!, PUT!, PRINT USING, MKI\$, MKS\$, CVI, and CVS.

The INSTR command returns the position of a search string within a target string. The syntax for the command is

```
VAR1 = INSTR ( VAR2, "STRING1",  
"STRING2" )
```

VAR2 is the character position within the target string where you begin searching. STRING1 is the target string and STRING2 is the key. VAR2, or the starting character position, is optional. If you omit it the search starts at the beginning of the target string. After you evaluate the function, VAR1 will contain the character position within the target string where the key was found. For example, if you have the string "THIS IS A TEST", and the key "IS", the statement

```
X = INSTR ("THIS IS A TEST", "IS")
```

will return the variable X with the value 3. You may have expected X to equal 6. However, the word 'THIS' contains an 'IS' in it. The statement

```
X = INSTR (4, "THIS IS A TEST", "IS")
```

will return X with the value 6. This time, we specified the fourth character as the starting point for the search.

The INSTR command also acts as the argument for an IF-THEN-ELSE statement.

```
IF INSTR("THIS IS A TEST", "IS")  
THEN GOSUB 1000 : ELSE PRINT "NO"
```

will cause a GOSUB to line 1000. The statement following the ELSE will not

be executed. However, if you change this statement to

```
IF INSTR(7, "THIS IS A TEST", "IS")  
THEN GOSUB 1000 : ELSE PRINT "NO"
```

control will pass to the ELSE statement and the word 'NO' will be printed to the screen.

The VARPTR command returns the memory address (in decimal) of:

1. The exponent of the variable A,
2. The least significant byte of the two-byte integer A%,
3. The byte defining the length of the string A\$.

To use this information, you must understand how BASIC stores variable data in memory. Numeric variables such as A and A% are stored in five-byte and two-byte locations, respectively. VARPTR will return the address of the beginning of this memory location sequence. In the case of string variables, a three-byte descriptor defines the string. The first byte is the length of the string in memory, and the second and third bytes are the address pointer to the string.

VARPTR may also be used to determine whether or not a variable exists. For example, if you study the statement

```
X = VARPTR (A$(1))
```

you will see that X will be returned with the value zero if the variable does not exist. Frequently, I need to know if an array has already been dimensioned. Without VARPTR, the only recourse is to redefine it and trap the error with an ON ERROR GOTO statement. I avoid ON ERROR GOTO statements; they make it too easy to build hidden 'BUGS' into a program.

The EXEC and MERGE commands each accept input from the disk as though it were entered from the keyboard. Either command accepts input from a SEQUENTIAL DATA file or a LIST# (ASCII) file.

The MERGE command enters program lines from the disk file as opposed to entering them from the keyboard.

This feature is useful when standard subroutines are to be used in several programs. For example, you may have a particular subroutine that is used to address the cursor on your terminal. Rather than manually entering the program lines each time you want to build a new program, the subroutine may first be entered from the keyboard, then LIST#ed out to a file called CURSR. Then whenever you want to use the subroutine within a program you simply enter MERGE "CURSR". This command, entered from the keyboard, will open the CURSR file and insert the lines into the program. You can save quite a bit of development time here!

The EXEC command will EXECute the command lines as they are read from the file. But with an EXEC file, the commands must be legal direct commands, such as PRINT, A=1, OPEN, CLOSE, PUT, and GET. Examples of commands that are not legal direct commands are INPUT and PRINT USING. Therefore, they may be used in files that are to be MERGED, but not in files that are to be EXECed.

The EXEC command is useful for repetitive tasks. For example, when you have several programs you want to list to the printer, you can create an EXEC file that will initialize the output device, load the first file, list it, load the next file, list it, and so on. This can all be done without any human intervention. Remember, any sequence of commands you enter repetitively from the keyboard may be put into an EXEC file and reused.

The EXEC command also accepts input from a string variable. This feature lets you build a command in the variable A\$, and then execute and EXEC A\$ command. Any string variable may be used. However, your commands may be no longer than 250 bytes. Of course, if several commands are to be EXECed, they could be constructed in a string array and executed in a FOR-NEXT loop as in:

```
FOR X = 1 TO 5  
EXEC A$(X)  
NEXT
```

(Continued on next page)

The Single Life *(continued)*

You might use this feature when you execute routines that are to be invisible to the user.

Some Printing Conventions

HDE has implemented a command, 'CALL', for directing output to peripherals such as printers and modems. The syntax for the command is CALL "DEVICE NAME", where the DEVICE NAME is a three character name associated with a binary or machine-language program stored on the system disk.

To use the CALL command you must either write or purchase the device driver program. This device driver is then SAVED to the system disk (drive 0 to 1). The CALL command will load and initialize the driver. With the driver initialized, output may be directed to the screen, the device, or both. To output to the device, commands such as PRINT, LIST, FIND, and LIB are followed by an exclamation point (!), as in PRINT!, LIST!, FIND!, and LIB!. To output to the screen, even while the device is enabled, eliminate the exclamation point. To disable the device, use the command CALLO (call zero). Once the device is disabled, output from statements such as PRINT! will be directed to the screen. To change the output device from a printer to a modem, just execute another CALL with the proper device name as the argument, as in CALL "MOD".

You may want to write a driver that accepts input from a modem or another terminal. Then when you want to pass control to that device, just initialize it with the CALL command.

PRINT USING may be used to manipulate string data. If you consider that A\$="FRED", B\$="SMITH", then the statement

```
PRINT USING "PAY TO !! %    %";
A$";";B$
```

will print PAY TO F. SMITH to the terminal. To dispatch this to an output device, use the statement:

```
PRINT! USING "PAY TO !! %    %";
A$";";B$
```

The exclamation point after the PRINT command directs the output to the external device that was initialized with the CALL command.

PRINT USING allows seven different types of format identifiers for dealing with numbers. The pound sign is used exclusively for defining the field width of a number. The PRINT USING command in conjunction with the pound sign causes number fields to be right-justified. For example, if you wish to print a column of numbers beginning at position 50 on the page, you could use the command:

```
PRINT TAB(50); USING "#####,#"; N
```

The use of the comma in the field specifier will cause a comma to be output every three places in the number. Your printout might look like:

```
123,456
   232
  1,508
```

If decimal positions are to be defined, simply use the command

```
PRINT TAB(50); USING "#####,#.###"; N
```

and the column will be right-justified, rounded to two decimal places, and zero-filled on the right.

```
123,456.25
   232.00
  1,508.07
```

File-Handling Techniques

Along with these unique commands, I want to introduce some of HDE Disk BASIC's file-handling techniques. There are three types of data files: SNAPSHOT, SEQUENTIAL, and RANDOM access. In addition, you have the ability to create an ASCII file of the program listing using the LIST# (list pound sign) command.

The main difference between the different types of data files is the way the data is stored on the disk and the techniques used to access it. First of all, the snapshot data file is, as its name implies, a snapshot of all the data in memory. If you can picture being able to grab the data in memory, compress it into one block, and then write it to the disk, you can understand the operation of this file. It is most useful when saving analytical data. For example, if

you are accumulating data and monitoring the results of laboratory tests, but need something recorded quickly, the command SAVED "TEST1" (meaning "save data") will, in a matter of seconds, write the contents of every variable to the disk file TEST1. To reload the information for later analysis, simply execute the command LOAD "TEST1" and memory will be restored to its previous contents.

With the RANDOM access file you can randomly access records within the file *without* reading or writing any other part of the file. This provides quick access to any record in the file.

The SEQUENTIAL data files are useful for data such as tax tables, rate tables, etc. Sequential files are best used when data fields or records are of varying lengths. Normally this type of data is manipulated in memory then written to the disk file when the user has completed working with it. The disadvantage of sequential file use is that to read the last record in the file, you must read the entire file. The same is true when changing one record in the file: you must read the file, make the change, and rewrite the entire file. But sequential files are usually more compact than RANDOM files.

To use sequential files properly, you must understand the structure of the file. First, records within the file may be terminated by a carriage return character (\$OD), a comma, or, when dealing with numerical fields, a space. The end of the file is signified by an end-of-file, or EOF mark. If you could look into the disk, you might find any of the following structures in a sequential file:

```
THIS IS RECORD 1$OD THIS IS
RECORD 2$ODEOF 22 33 44 $OD 11 66
55 $OD 13 $OD 99 21 $ODEOF 22, 33,
- 44$OD 11, 66, 55$OD 13$OD
99, -21$ODEOF "THIS IS RECORD
1"$OD"THIS IS RECORD 2","THIS IS
RECORD 3"$ODEOF
```

The first two files were created using the PRINT# command, the second two using the WRITE# commands.

Please send all correspondence for Mr. Rinehart to 1508 Stanton St., York, PA 17404.

MICRO

OHIO SCIENTIFIC

THE WIZARD'S CITY — search for gold in the dungeons beneath the Wizard's city or in the surrounding forest. A dynamic adventure allowing progress in strength and experience. All OSI — cassette \$12.95, disk \$15.95.

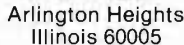
OSI HARDWARE 15% OFF RETAIL PRICES!

GALACTIC EMPIRE — a strategy game of interstellar conquest and negotiation. Compete to discover, conquer, and rule an empire with the computer or 1-2 other players. C4P, C8P cassette \$12.95, disk \$15.95.

AIR TRAFFIC ADVENTURE — a real time air traffic simulation. C4P, C8P disks \$15.95. Plus S-FORTH, PACKMAN, CRAZY BOMBER, ADVENTURE, TOUCH TYPING, INTELLIGENT TERMINAL and more. Send for our free catalog including photos and complete descriptions.

(312) 259-3150

Aurora Software Associates
37 S. Mitchell
Arlington Heights
Illinois 60005



D.S.I. C4P Trs-80 Model I & III

The Room's of Cygnus IV
You are in a room with walls placed randomly throughout. There are three to ten ROBOTS bent on destroying you with laser fire or by touching you. You must destroy all the robots in a room to advance a level. The higher the level the faster the robots. WATCH-OUT the walls are electrified!! There are two skill levels.
D.S.I. C4P - Color Sound Joysticks 8K Cassette - \$9.95
Trs-80 I & III Sound both 16K basic & 32K machine Cassette - \$9.95
Trs-80 Model I Disk 32K basic & 48K machine - \$15.95

Murder Mansion - Adventure #1
You and six other people are exploring a three story house for a treasure. WATCH-OUT!! Someone is killing the others and you could be next!
Trs-80 version has a graphic representation of each room!!
D.S.I. C4P - 8K Cassette - \$12.95
Trs-80 I & III (Graphics) 32K Cassette - \$15.95
Trs-80 I 48K Disk - \$19.95

Color Music
This is a fantastic machine language program. This program has 4 sections that will reproduce patterns on the screen with music. Connect your recorder or stereo to the input jack on the computer and watch the music come alive. The four options are:
1. GRAPHIC PATTERNS 2. COLOR PATTERNS
3. COLOR LASERS 4. COLOR BARS
D.S.I. C4P Color 8K Cassette - \$8.95

Tank Trap
This is a one-player game where you fight the computer. The object is to destroy the ENEMY TANK and HOUSES before he destroys YOU and YOUR HOUSES. But WATCH-OUT for the hidden MINES!!
D.S.I. C4P Color Sound Joysticks 8K Cassette - \$8.95

Star Explorer
You are the only survivor of a sneak attack on the fleet. You must destroy the remaining enemy BATTLESHIPS and their HOME PLANET before time runs out. You must dock at STARBASES to refuel and pick-up pieces of the PLANET DESTROYER weapon. But WATCH-OUT one of the STARBASES has been taken over!!
D.S.I. C4P Color Sound 8K Cassette - \$8.95

Send \$1.00 for catalog.

COMP-U-GAMER SOFTWARE
P.O. BOX 807
NEVADA, MISSOURI 64772

CSE means OSI Custom After Market Software for C1-P and C4-P machines

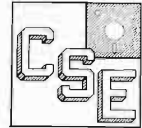
Basic Enhancer:
Renumberer, Auto Sequencer, Screen Control functions, and a tape I/O system that is faster and has file names. For C1-P and C4-P \$21.95

Micro-Processor Word Processing system:

Now do reports, letters, and other documents that need that professional touch. Coded in Machine Code for a minimum hassle (for C1-P only) . . . \$44.95

Write us for information on new disk programs or send \$1 for catalog.

Computer
Science
Engineering



Box 50 • 291 Huntington Ave. Boston 02115

Computer
Case
Company



comp
case
NOW TOLL FREE
800-848-7548

• AP104

Attache style cases for carrying and protecting a complete computer set-up. Constructed of the highest quality luggage material with saddle stitching. Will accommodate equipment in a fully operational configuration along with manuals, working papers and disks. Never a need to remove equipment from case. Simply remove lid, connect power and operate.

• AP101	Apple II with Single Disk Drive	\$109
• AP102	Apple II with Double Disk Drives	119
• AP103	Apple II, 9 inch Monitor & Double Drives	..	129
• AP104	Apple III, two additional Drives & Silentyte		139
• AP105	12 inch monitor plus accessories	99
• P401	Paper Tiger 440/445/460	99
• P402	Centronics 730/737	89
• P403	Epson MX70 or MX80	89
• P404	Epson MX100	99
• P405	IDS 560 or Prism Printer	109
• CC80	Matching Attache Case (5")	85
• CC90	Matching Attache Case (3")	75
• CC91	Matching Accessory Case	95

Computer Case Company

5650 INDIAN MOUND CT. COLUMBUS, OHIO 43213 (614) 868-9464



EXCITING NEWS FOR COLOR COMPUTER USERS

FLEX, OS-9 and the Radio Shack Disk System ALL on the SAME Color Computer

Would you believe that you can run FLEX, OS-9 and Radio Shack disk software on the same Color Computer, and all you have to do is change the disk? That's right, just change the disk. If you have a 32K Color Computer with the Radio Shack disk system, all you need to do is make a trivial modification to access the hidden 32K, as described in the Feb. issue of COLOR COMPUTER NEWS and the April issue of 68 Micro. You can get FLEX from us right now. OS-9 will be ready by summer. Please note that this will only work with the Radio Shack disk system and 32K/64K memory chips that RS calls 32K. Maybe they put 64K's in yours, too. If you don't have a copy of the article, send a legal size SASE (40c stamps) and we'll send it to you.

Using this system to run FLEX and OS-9 has many advantages. First, it gives you 48K from zero right up to FLEX. This means that ALL FLEX compatible software will run with NO MODIFICATIONS and NO PATCHES! There are no memory conflicts because we moved the screen up above FLEX which leaves the lower 48K free for user programs.

What you end up with is 48K for user programs, 8K for FLEX and another 8K above FLEX for the screens and stuff. We have a multi-screen format so you can page backward to see what scrolled by and a Hi-Res screen that will enable us to have 24 lines by 42 character display is on the way. That's better than an Apple!

We also implemented a full function keyboard, with a control key and escape key. All ASCII codes can now be generated from the Color Computer keyboard!

We also added some bells and whistles to Radio Shack's Disk system when you're running FLEX or OS-9. We are supporting single or double sided, single or double density, 35, 40 and 80 track drives. If you use double sided drives, the maximum is three drives because we use the drive 3 select for side select. When you are running the Radio Shack disk, it will work with the double sided drives but it will only use one side and only 35 tracks. Using 80 track drives is okay, but will not be compatible with standard Radio Shack software. You can also set each drive's stepping rate and drive type. (SS or DS -SD or DD).

In case you don't understand how this works, I'll give you a brief explanation. The Color Computer was designed so that the roms in the system could be turned off under software control. In a normal Color Computer this would only make it go away. However, if you put a program in memory to do something first (like boot in FLEX or OS-9), when you turn off the roms, you will have a full 64K RAM System with which to run your program (FLEX or OS-9). When the roms are turned off, it is as if you had removed them from the computer. They are gone!

Now, we need the other half of the 64K ram chips to work, and this seems to be the case most of the time, as the article states. Of course, you could also put 64K chips in.

Some neat utilities are included.

MOVEMOM moves Color Basic from ROM to RAM. Because it's moved to RAM you can not only access it from FLEX, you can run it and even change it! You can load Color Computer cassette software and save it to FLEX disk. Single Drive Copy, Format and Setup commands are also included.

Installing FLEX is simple: insert the disk and type:

RUN "FLEX"

That's all there is to it! You are now up and running in the most popular operating system for the 6809. There are hundreds of software packages now running under the FLEX system. Open your Color Computer to a whole new world of software with FLEX.

FLEX \$99.00

NEW LOW PRICE INCLUDES OVER 25 UTILITIES!

FLEX Editor	\$ 50.00
FLEX Assembler	\$ 50.00
FLEX Standard BASIC	\$ 65.00
FLEX Extended Business BASIC	\$100.00

Other languages available include: FORTH, Pascal, Fortran77, C, plus more.

Application packages include: AIR, GIL, A/P, Inventory, Electronic, Spreadsheets, Accounting, Database programs and more. SEND FOR LIST.

TRS-80 COLOR COMPUTER COMPLETE WITH 64K RAM, 24K ROM, SINGLE DISK DRIVE AND FLEX, SET UP AND READY TO RUN FOR ONLY \$1,375. Includes 60 day extended warranty. If you have a Computer, call about RS disk controllers and drives.

FRANK HOGG LABORATORY, INC.

130 MIDTOWN PLAZA • SYRACUSE NEW YORK 13210 • (315) 474-7866

80 x 25

PET/CBM™

2000/3000/4000 Series

not using a CRT, or display controller chip

\$275.00*

Select either
80 x 25 or 40 x 25

On The
Built-in
Display

From the keyboard or program

Displays the full, original character set

Available from your local dealer or:

EXECOM CORP.

1901 Polaris Ave.
Racine, WI 53404
Ph. 414-632-1004

*Plus installation charge of \$75.00

Available only for Basic 3.0 & Basic 4.0

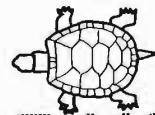
PET & CBM™_a

trademark of Commodore Business Machines

ZOOM HiRes Graphics Printing Utilities

GRAF PAK

for 48K
Apple DOS 3.3



PAK

for
Apple PASCAL

Anadex grafixPLUS, Epson GRAFTRAX, Integral Data 440/445/460/560/Prism, Centronics 739, Apple Silentype, C.Itoh 8510 & F-10, NEC PC-8023 & Spinwriters, Okidata 82a/83a, Qume Sprints, Diablo 6xx/16xx, DEC LA-34(XL), Infoscrite 1000, Datasouth 180, Malibu 200, MPI 88G

Featuring —

- Independent horizontal and vertical scale factors.
- Ability to print horizontal or rotated single pages, or panoramas with page 1 above or beside page 2.
- *Unmatched ease of use* — direct, no guessing zoom window placement — keystroke driven.
- *Complete* instructions for attachment to your program.
- HiRes picture disc-capture utility.
- **GRAFPAK** features *one* keystroke graphic file loading; on screen *text annotation*, bordering, cropping, framing, and window fill/erase.

2281 Cobble Stone Court
Dayton, Ohio 45431
513/426-3579

GRAFPAK
\$34⁹⁵

SmartWare

TURTLEPAK
\$34⁹⁵

OSI Disk Users

Double your disk storage capacity Without adding disk drives

Now you can more than double your usable floppy disk storage capacity—for a fraction of the cost of additional disk drives. Modular Systems' DiskDoubler™ is a double-density adapter that doubles the storage capacity of each disk track. The DiskDoubler plugs directly into an OSI disk interface board. No changes to hardware or software are required.

The DiskDoubler increases total disk space under OS-65U to 550K; under OS-65D to 473K for 8-inch floppies, to 163K for mini-floppies. With the DiskDoubler, each drive does the work of two. You can have more and larger programs, related files, and disk utilities on the same disk—for easier operation without constant disk changes.

Your OSI system is an investment in computing power. Get the full value from the disk hardware and software that you already own. Just write to us, and we'll send you the full story on the DiskDoubler, along with the rest of our growing family of products for OSI disk systems.

™DiskDoubler is a trademark of Modular Systems.

Modular Systems

Post Office Box 16C
Oradell, NJ 07649.0016
Telephone 201 262.0093

NEW PET/CBM SOFTWARE

Let Computer Mat turn your Pet into a
Home Arcade!

ASTEROIDZ — Its your ship vs. a swarm of killer gammaroidz. You are on a collision course and must destroy them before they blast you into the next galaxy. Four levels of play. Has hyperspace keys that move you around. Arcade style entertainment at its finest. Great graphics and sound.

Cass. 8K \$9.95

MUNCHMAN — How many dots can you cover? It's you against the computer munchers ZIP and ZAP. Can you clear the maze first or will they get you? Number keys move you up, down, right and left. GREAT GRAPHICS AND SOUND.

Cass. 8K \$9.95

TARGET COMMAND — Its you against a barrage of enemy lazars that are aimed at your ammo dumps. Sight in on the targets and score as many hits as you dare. As your skill increases so does the the difficulty — (5 levels to select). This is an arcade-style game with great graphics and sound effects. A must for your PET/CBM.

Cass. 8K \$9.95

ALL OUR SOFTWARE RUNS IN 8K
OLD-NEW ROM — 40 CHR. SCREEN
WRITE FOR FREE CATALOG OF VIC/PET SOFTWARE
PLEASE ADD \$1.00 PER ORDER FOR SHIPPING
COMPUTER MAT • BOX 1664R • LAKE HAVASU CITY, AZ. 86403

Commodore and MICRO

by Loren Wright

Commodore has been a dominant force in the microcomputer world since the Personal Electronic Transactor (or PET) was introduced in 1977. The first PETs admittedly had their problems. The small keyboard, a holdover from Commodore's calculator background, gave the PET a toy-like appearance. There was no resident machine language monitor — a tape version had to be loaded into RAM each time. There were also a few problems, of varying annoyance, in the operating system. Documentation had to be obtained from users' groups, since Commodore would not, and perhaps could not, provide very much.

In the U.S., the PET came out in the face of stiff competition from Apple and Radio Shack. The Apple attracted many people with its high-resolution color and full-size keyboard, while Radio Shack, with its nationwide network of stores and well-organized marketing effort, drew even more attention. To compound their problems, Commodore attempted to sell the PET only directly or through its Mr. Calculator stores, and there were delivery delays of many months.

The PET, as many other computer shoppers recognized, offered a system complete with CRT display, cassette mass-storage, and fully implemented BASIC for a price less than comparable systems from either Apple or Radio Shack. In the rest of the world, where Commodore was organizationally better equipped to compete, the PET became (and still is) the number one microcomputer.

The company corrected most of the problems with a new operating system. Unfortunately, this was done without much consideration for those who had already invested a lot of time and money developing commercial software for the old operating system. Many people abandoned ship at this point, but most adjusted and are still loyal PET owners. Since then, there has

been yet another operating system introduced, but this time the changes were far less radical, and Commodore cooperated considerably more in the transition.

To understate the situation, Commodore has been unpredictable in its approach to the market. When the 80-column business machine and the VIC were announced, there were widespread fears that the company would abandon PET owners in favor of the more lucrative entertainment and business markets. So far, those fears have not been justified. It is clear now, particularly with the announcement of several new computers, that Commodore wants to compete in all microcomputer markets. The new line-up will apparently include the Ultimax, the VIC, the SuperVIC, the PET, the 8016/8032, the color 8032, the Commodore-64, the SuperPET, and the 8096. Each of these is aimed at a particular segment of the market.

If its new advertising campaign is any indication, Commodore intends to provide the best value for any microcomputing need. The company plans to accomplish this not by inventing radically new computers, but rather by producing variations on its PET and VIC themes to compete over the full range of the market. To quote Jack Tramiel, the man behind Commodore, "We will become the Japanese!" — meaning that they will offer a lower-priced alternative. Whether Commodore can actually accomplish its goals is still uncertain.

MICRO has been covering the PET since its inception. Much of our job in the early days was to provide the information not provided by Commodore, and to help PET owners get around the bugs in their systems. Things have progressed much further than that now. The PET system is virtually bug-free and good documentation is available not only from Commodore, but from a

number of other sources. We will continue to publish articles of special interest to PET users, but you will find many of our other articles valuable as well. More articles written for other computers will be accessible to PET users, and we will continue to expand your horizons with material on new programming techniques, languages, and applications.

This issue's feature article "Growing Knowledge Trees," by David Heise, introduces artificial intelligence to MICRO readers. While it is written especially for the PET, I recommend that all MICRO readers try to see this program in operation. It should provide some ideas for your own artificial intelligence programs.

"Menu and Tape Timer," by Dale DePriest is a sequel to last month's "A Real Tape Operating System." In that article he discussed the good and bad features of the PET's tape system and presented some techniques to get the most from that system. This month's programs will help you turn your cassette collection into a well-organized file retrieval system. Although a disk drive is faster and more convenient, the PET's cassette system, with a few refinements, can offer a considerably less expensive alternative, which is still very satisfactory.

Louis Sander's "PET Memory Protector" is a simple circuit that is inserted between one of your PET's static RAM chips and its socket. Depending on where it is installed, PMP protects 1K or more of RAM from BASIC, LOADs and resets. The reset button, which is part of the circuit, can be used for either a cold or a warm reset.

MICRO



Skyles Electric Works

BASIC Programmer's Command-O™

For PET/CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any PET or CMB system**. 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

For PET 4016, 4032, CBM 8032, SuperPET or BASIC 4.0 upgrade

BASIC Programmers Command-O™

AUTO DUMP DELETE FIND(improved) HELP KILL OFF TRACE(improved) RENUMBER(improved) INITIALIZE MERGE MOVE EXECUTE SCROLL OUT SET SEND PRINT USING BEEP

```
100 GOSUB 180
105 PRINT USING C$, A, B$
130 INPUT "TIME", D$
131 INPUT "DAY", E$
160 IF B$ - C THEN 105
180 FOR X = IT09
183 PRINT Y(X):NEXT
184 RETURN
200 I = X/19
READY
RENUMBER 110, 10, 105-184
READY
LIST
100 GOSUB 150
110 PRINT USING C$, A, B$
120 INPUT "TIME", D$
130 INPUT "DAY", E$
140 IF B$ - C THEN 110
150 FOR X = IT09
160 PRINT Y(X):NEXT
170 RETURN
200 I = X/19
READY
```

```
MERGE D1 "BUY NOW"
SEARCHING FOR BUY NOW*
LOADING
READY
RENUMBER 100, 10
READY
FIND B$
110 PRINT USING A$, B$, C$ - D$
280 B$ = "NOW IS THE TIME"
READY
```

```
580 BA = BA - 1
590 RA = 123*5X/92 * BA*10
600 IF BA = 143 THEN 580
610 RETURN
620 C$ = "PROFIT $#, ###.## DAILY"
630 PRINT USING C$, P1
640 D$ = "LOSS $#, ###.## DAILY"
650 PRINT USING D$, L1
RUN
PROFIT $1, 238.61 DAILY
LOSS $ 0.00 DAILY
READY
```

◆ NOTICE ◆
COMMAND-O IMMEDIATELY AVAILABLE
FOR
ALL MODELS PET/CBMS
PLEASE SPECIFY !!
◆ NOTICE ◆

INSTALLATION:

Installs into socket UD12 (\$9000) in "fat forty's", 8032's or SuperPET's. Installs into socket UD3 (\$9000) in BASIC 4.0 upgrades.

PRICES:

For PET 4016, 4032, BASIC 4.0 upgrade COMMAND-O CO-40N \$75.00
For CBM 8032, SuperPET (SP9000) COMMAND-O CO-80N \$75.00

*Shipping and handling \$2.50 USA/Canada, \$10.00 Europe/Asia
California residents please add 6% or 6-1/2% sales tax as required
order your Disk-O-Pro, today*

AVAILABILITY: Immediately from your LOCAL DEALER

or

VISA, MASTERCHARGE ORDERS CALL (800) 227-9998 (except California residents)
CALIFORNIA ORDERS PLEASE CALL (415) 965-1735



Skyles Electric Works

231 E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

PET Memory Protector

by Louis F. Sander

This easy-to-build add-on for 8K PETs selectively isolates 1K or more of memory from BASIC and from resets. The protected area is an ideal repository for monitors or other machine-language programs.

PMP

requires:

PET with socketed 6550 or 2114 RAM chips (all small-keyboard PETs, except the most recent release) and a number of electronic parts.

PET users have a shortage of protected memory for machine-language programs. The PET Memory Protector is a simple add-on device that eliminates this shortage. In a typical PET, only the second cassette buffer, with its meager 192 bytes, is out of reach of BASIC, resets, and LOADs. The only way to protect an area in high memory is to do several zero-page POKEs, an annoying task. The PET Memory Protector, or PMP, provides a simpler and more reliable way to reserve 1K blocks of high memory for machine-language programs or any other use.

PMP is activated by momentarily depressing one switch, eliminating the need for memory-reserving POKEs. BASIC cannot write into the PMP-protected area unless specifically directed, and LOADING a tape from either cassette does not affect it. The PMP includes a reset function that allows selection or deselection of memory protection while the reset is performed, all with one simple control.

Installation of the PMP requires no drilling or cutting of the PET, and no soldering to any PET component. You simply insert the PMP connector between one memory chip and its socket,

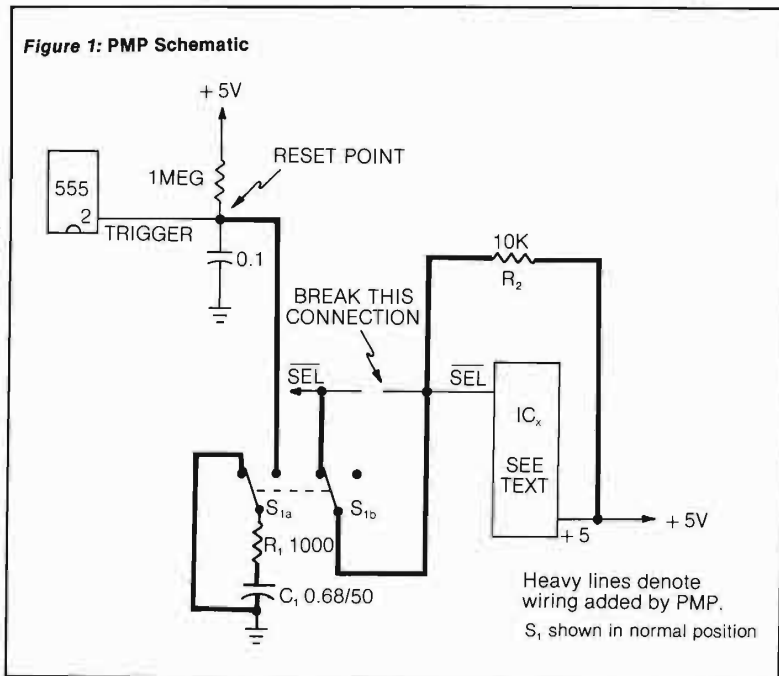
and mount the PMP switch in existing holes in the PET. The typical installation protects 1K of memory, but larger 1K multiples can be protected simply by moving the PMP connector. The present version of the PMP will work only with PETs using the type 6550 or 2114 RAMs in plug-in sockets (basically, all the small-keyboard machines). Work is in progress on a version for the large-keyboard machines. Construction of a PMP is not difficult for an experienced electronic builder; non-builders can purchase a fully assembled and tested version from the authors.

Theory of Operation

Figure 1 is the schematic diagram of the PMP. When S_1 is closed, it connects R_1 and C_1 to the trigger pin of PET's power-up timer, and opens the chip-select line to IC_x . IC_x can be any one of

PET's RAM chips; typically it will be one of the two that constitute the top 1K of user RAM. The charging effect of C_1 momentarily lowers the voltage on the timer trigger pin, which activates the timer and a power-on reset. At the end of the timer's one-second cycle, PET writes a character into the lowest memory location of the user program area, then reads the contents of that location. If the read and write are identical, PET repeats the process at the next higher memory location. The first time the read and write do not match, PET concludes that it has passed the top of available RAM. It then sets its zero-page BASIC pointers accordingly, and puts the appropriate BYTES FREE message onto the CRT.

If S_1 is still actuated when the reset routine tries to write into IC_x , the \overline{SEL} line will still be broken by S_{1b} and the



\overline{SEL} pin will be held high by the voltage from R_2 . So the read/write process into IC_x will fail, and PET will conclude that IC_x and all memory above it does not exist. Because IC_x is paired with another chip that is not disabled by S_1 , the reset will have modified the lowest single byte in IC_x , but will not have affected any higher memory locations. The BYTES FREE message will include only those memory locations below IC_x . S_1 can be released as soon as the BYTES FREE message appears, and at that time IC_x will be fully functional, but BASIC will not know that it is there. In other words, IC_x and above will constitute a fully protected area of memory.

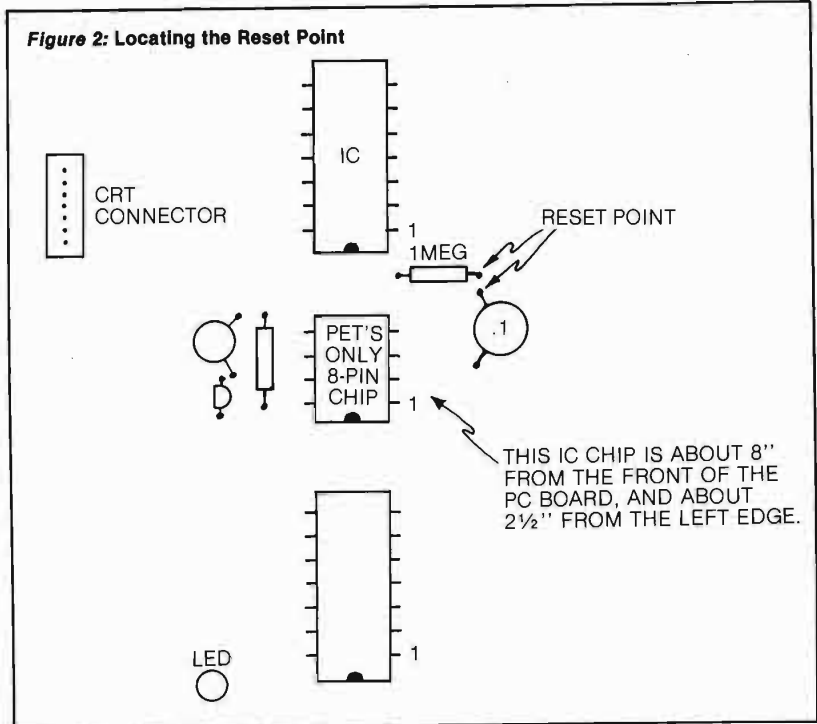
If S_1 is released before the timer finishes its cycle, IC_x will be properly connected when PET attempts to access it. The reset process will proceed normally, writing over any information in IC_x and above, and including those locations in the BYTES FREE message. IC_x and above will not be protected memory. In either case, as soon as S_1 is released, C_1 will discharge through R_1 , to be ready for the next reset.

Construction

If you are not an experienced electronics builder, you shouldn't try this project on your own, since a miswired PMP could mean disaster to your PET. For the builder with any experience, PMP construction is straightforward, except for breaking the \overline{SEL} line to IC_x . For the 6550, \overline{SEL} is pin 18, and +5 is pin 17. For the 2114, \overline{SEL} is pin 8 and +5 is pin 18. Make up a "PMP connector" from two wire-wrap IC sockets. Plug the sockets together, piggyback-style, and cut the pin carrying the \overline{SEL} lead from one socket to the other. If you want to do a more professional job, use one wire-wrap socket and the plastic base from another, cutting one of the pins. Either way, solder your S_{1B} leads to the severed ends, and glue everything together so it can't move. Solder R_2 to the appropriate pins of the upper socket, and you're in business. (You need wire-wrap-type sockets for this work, because the solder-tab-type pins are too short to work with, unless you're used to microsurgery.)

We advise the prospective builder to be persistent in his search for parts, since S_1 and the IC sockets are not common items. They are manufactured by the thousands for industrial use, but your local Radio Shack doesn't carry them.

Figure 2: Locating the Reset Point



Installation

The first step in installation is to unplug your PET. Then find a way to mount S_1 permanently. You can either drill a hole in your PET, or drill a 1/2" x 3" strip of heavy sheet metal to accept S_1 . Then mount it to the cover hold-down bracket on PET's right side, using two additional holes drilled through your piece of sheet metal. If S_1 is properly chosen, it will easily fit in the 1/2" space between PET's cover and base, making a very attractive and unobtrusive installation.

Next, connect the wire from S_{1a} to the reset point. Here you can solder a wire directly to PET's circuit board, or you can use a tiny test clip to connect it to a component lead. The reset point is easy to find by locating the 555 timer chip, which is the only eight-pin IC in the PET. It's on the far left side of the printed circuit board, about eight inches from the front edge. The reset point is accessible either at pin 2 of the 555, or at the resistor or capacitor lead wire shown in figure 2. (By the way, this is the same point used by the reset buttons on old ROM PETs.)

Finally, locate IC_x and put the PMP connector between it and its socket. At the very front of the main printed circuit board are two identical rows of eight IC chips in sockets; this is PET's RAM. Each 1K of memory is made up

of one IC in the front row, plus its partner in the second row. Half a byte is stored in each chip, for 1024 memory locations in each pair.

If your memory chips have 18 pins each, they are 2114's, and the IC's making up the lowest memory locations are to the far left. The highest memory locations are to the right. To protect 1K of memory, the PMP plugs into either one of the rightmost chips. With 2114's the PMP can be plugged into any RAM socket, protecting any number of 1K memory blocks.

If your memory chips have 22 pins, they are 6550's, and things work differently. The low memory locations are to the right in this case, and the high ones are to the left. Your PMP will only work properly in the highest 1K (the leftmost socket), or the highest 4K (number 4 from the left).

To locate IC_x , first determine how much memory you want PMP to protect. If it's 1K, then IC_x is the rightmost or leftmost IC in the front row. If you want to protect 2K of memory, IC_x is the chip just next to that one, and so on, at the rate of 1K per chip. For test purposes, you will need to protect 1K, so initially use the end chip in the front row. Use the left chip for 6550's, or the right chip for 2114's. (In every case, the corresponding chip in the second row could be used, with identical effect.

We've arbitrarily chosen the front row chips because they're easier to get to.)

Before removing IC_x, note the U-shaped depression on its top at one end. That is an orientation mark, and when it faces you, with the IC pins pointed downward, pin 1 is the closest pin to you on the right. See figure 2 for examples. Take careful note of IC_x's orientation, so that you'll be able to insert the PMP connector and IC_x in the proper direction.

When you've done this, gently pry IC_x from its socket, using a small screwdriver inserted from the front. Use standard static protection techniques: keep yourself grounded, and lay the naked IC, pins downward, on a piece of foil or conductive foam. Now insert the PMP connector into the vacant socket, being extremely careful to preserve proper orientation. Using static protection techniques, and once again paying careful attention to orientation, insert IC_x into the PMP connector, and you're ready to test your PMP.

Test and Operation

Visually inspect the installation to make sure there are no broken wires or short circuits. Make sure that the PMP is plugged into the correct socket, and that all its pins are making contact. (Look closely, as bent pins are common, and easy to miss.) Make the same check on the IC chip, where it plugs into the PMP. Finally, double check the orientation of the IC and the PMP; if either one is in backwards, correct it immediately.

When you're certain that everything is as it should be, turn on the power to your PET. You should get the normal BYTES FREE message (7167 bytes on the 8K PET). Now load a machine-language program (MLP) of some sort into the top part of the top 1K of memory. Ideally, it should extend to the very last free byte. Be sure your program doesn't use the very first byte of the top 1K, since that byte will be modified by the reset routine. Run your MLP to make sure that it works.

Now activate S₁, and keep it activated until the BYTES FREE message appears once again. If all has gone well, that message will have appeared about one second after you first activated S₁, and will indicate 1024 fewer bytes than normal. Next, LOAD and RUN a BASIC program that uses several string variables. Run your MLP once again. If both programs work properly, PMP has protected upper memory from being written into by BASIC.

Parts List

C₁ — 0.68 F, 50 wv

R₁ — 1000 ohm, ¼ watt

R₂ — 10K ohm, ¼ watt

S₁ — DPDT momentary toggle or pushbutton switch

Two 18-pin or 22-pin wire-wrap-type IC sockets (pin configuration depends on your RAM type)

Hookup wire

Glue (Devcon clear epoxy or similar)

Optional ½" x 3" piece of heavy sheet metal (for switch mounting bracket)

Total parts cost should be \$10-\$12 for top-quality, name-brand parts.

For the final test, momentarily activate S₁, this time being sure to release it before the BYTES FREE message appears. If you get a normal BYTES FREE message, and if both programs are gone from memory, your PMP is working correctly. Congratulations on a job well done!

Now here's the full story on clearing and protecting memory in your PMP-equipped machine.

1. POWER ON clears all memory, overwriting it with characters dictated by your ROM set.
2. Using either cassette drive to LOAD, SAVE, or VERIFY clears the associated cassette buffer, replacing what was there with data from the tape header. The unused cassette buffer is not affected.
3. Momentarily depressing S₁ and releasing it before the BYTES FREE message appears clears all memory *except* the cassette buffers, and gives a normal BYTES FREE message. The cleared memory is overwritten with characters dictated by your ROM set.
4. Holding S₁ until the reduced BYTES FREE message appears clears all memory *except* the two cassette buffers and the memory above the first protected byte. That first byte will be altered by the reset process, but is protected afterward. Anything previously existing above that byte will be unaffected by the reset, and will

be protected from being written into by BASIC. It can be PEEKed and POKEd, but that is all.

That's the full story on the PET Memory Protector. We've found it to be a very handy device for protecting high memory, and we hope that you will, too. If you'd like to have a fully assembled and tested PMP, we've made some up that we'll send you for \$20 each. Just send your name, address, and RAM type to: Louis F. Sander at 153 Mayer Drive, Pittsburgh, PA 15237.

Louis F. Sander designs and markets electronic systems for hospitals and other health care providers. He is the originator of COMPTUER KINDERGARTEN™, a computer familiarization course for adults. Victor H. Pitre installs and services medical electronic systems. Both have worked in electronics since pre-transistor days, and they have collaborated on several hardware and software innovations for small computers.

MICRO

OSI-C4PMF

GOBBLER MANIA

Attempt the maze eating all you can but don't get caught
Full color & sound!
5¼ in. disk . . . \$12.95

SENTINEL

Is there any enemy around the next hall? Better ready your crossbow for battle.
5¼ in. disk . . . \$19.95

We also have: ALPHA BASE, RADAR TANK, HYPER ATTACK, and many utilities too!
All in 100% machine code.
Send \$1 for complete catalog to:



INTERESTING SOFTWARE

21101 S. Harvard Blv.
Torrance, Cal. 90501

- Z-FORTH IN ROM** by Tom Zimmer
5 to 10 times faster than Basic. Once you use it, you'll never go back to BASIC!
source listing add \$ 75.00
\$ 20.00
- OSI FIG-FORTH** True fig FORTH model for OS65D with fig editor named files, string
package & much more \$ 45.00
- TINY PASCAL Operates** in fig-FORTH, an exceptional value when purchased with forth.
TINY PASCAL & documentation \$ 45.00
FORTH & TINY PASCAL \$ 65.00
- SPACE INVADERS** 100% machine code for all systems with 64 chr. video. Full color & sound
on C2, 4P & 8P systems. The fastest arcade program available. \$ 14.95
- PROGRAMMABLE CHARACTER GENERATOR**
Use OSI's graphics or make a complete set of your own! Easy to use, comes as a kit.
2 Mhz. boards \$ 99.95
\$ 84.95
\$109.95
- PROGRAMMABLE SOUND BOARD**
Complete sound system featuring the AY-3-8910 sound chip. Bare boards available. \$ 74.95
\$29.95
- 32/64 CHARACTER VIDEO MODIFICATION**
Oldest and most popular video mod. True 32 chr. C1P, or 32/64 chr. C4P video display.
Also adds many other options. \$ 39.95
- ROMS!!!**
Augment Video Mod with our Roms. Full screen editing, print at selectable scroll, disk support and many more
features. Basic 4 & Monitor \$ 44.95
Basic 3 \$ 18.95
All 3 for \$ 59.95
- 65D DISASSEMBLY MANUAL.** by Software Consultants. First Class throughout. \$ 25.95
A must for any 65D user.
- NUMEROUS BASIC PROGRAMS, UTILITY PROGRAMS AND GAMES ALONG WITH HARDWARE PROJECTS. ALL
PRICES ARE U.S. FUNDS. Send for our \$1.50 catalogue with free program (hardcopy) Memory Map and Auto Load
Routine.*

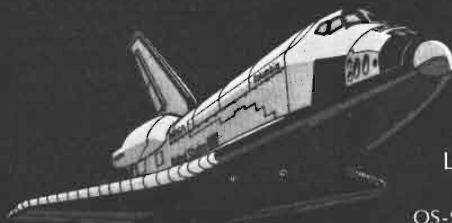


3336 Avondale Court
Windsor, Ontario, Canada N9E 1X6
(519) 969-2500

3486 Countryside Circle
Pontiac Township, Michigan 48057
(313) 373-0468



What's Good for the Space Shuttle is good for your Apple II! . . .

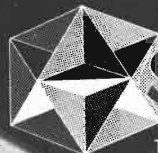


MICROWARE, creator of OS-9/BASIC Ø9 (used by N.A.S.A., and leading Universities, government agencies, and corporations Worldwide) joins with STELLATION TWO to deliver the same Operating system and Programming Language to the APPLE II.

OS-9/BASIC Ø9 are the result of a 3 year research project—designed with the 6809 in mind. This "Operators dream machine" combines with THE MILL microprocessor board to provide Apple II users with software features previously reserved for Mainframes and mini's.

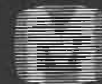
JUST PLUG IN THE MILL AND LET BASIC Ø9 WORK FOR YOU! other Stellation Two products include:

- Spooler: Allows Apple II to print while processing.
- Pascal Speedup: THE MILL with software for a 50% faster Apple with Pascal.
- Floating Point: Extends the MILL's power to floating point numbers.



**STELLATION
TWO**

The Lobero Building P.O. Box 2342
Santa Barbara, Ca, 93120
(805) 966-1140 TELEX 658439



MICROWARE

OS-9 is a trademark of Microware. BASIC Ø9 is a trademark of Microware and Motorola.

Apple II is a trademark of Apple computers.

Growing Knowledge Trees

by David R. Heise

Knowledge can be represented in tree diagrams that are stored and analyzed by computers. This PET program finds out what people know about a topic, analyzes answers, and shows users the organized results.

"Knowledge Trees" requires:

- 16K or 32K PET
- 3.0 Operating System
- Printer and disk drive are supported, but optional. Notes are provided for 1.0 and 4.0 conversion.

A computer needs to be knowledgeable if it is to help you classify plants, diagnose illnesses, or identify beliefs that hold down productivity. But how do you make a computer knowledgeable? How do you teach a computer what an expert knows? How do you represent knowledge?

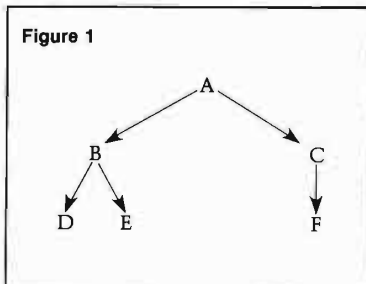
Research on these questions has been performed in computer science specialties like data base management and artificial intelligence, and also in social sciences like anthropology, sociology, and cognitive psychology. Data base management systems and psychological research demonstrate that some kinds of knowledge can be stored as associative networks with pointers linking items to related items. As a sociologist, I've shown that measurements about actors and behaviors can be used mathematically to generate information about social events. Artificial intelligence research and contemporary anthropology often represent knowledge as hierarchical trees of relationships.

This article discusses the tree concept for knowledge representation and presents a program that turns a Commodore microcomputer into a machine

for gathering hierarchical knowledge from people and reducing it to its simplest form. The program illustrates how to pass multiple parameters to an assembly-language subroutine via the USR function in BASIC. It shows you how to create a BASIC subroutine that can call itself recursively, and demonstrates a method for automatically loading and protecting assembled code, along with a BASIC program. Assembled routines are provided for carrying out operations on trees.

Data Structures

Hierarchical knowledge can be represented in a tree diagram, which is a type of directed graph. See figure 1.



Letters A through F represent *nodes* in the graph. These are the entities involved in hierarchical relationships. For example, the nodes could be A = living creature, B = mammal, C = bird, D = dog, E = cat, and F = robin.

The arrows are called the *edges* of the graph. All of the arrows on a graph represent a certain relationship. For example, in figure 1, each arrow could represent the relation "is a kind of." Thus, reading backward along an arrow, we see that B is a kind of A. Different arrows show which entities are related to other entities. In general, the node at the top of an arrow is a *superordinate* of the node at the bottom of the arrow. The node at the bottom of an arrow is a *subordinate* of the node at the top.

Relations in a tree diagram are *transitive*. This means that a relationship between two entities can be inferred when they are connected by a chain of arrows rather than by a direct arrow. For example, in figure 1, D (a dog), is a kind of B (mammal), and B (a mammal) is a kind of A (living creature). Therefore we can infer "a dog is a kind of living creature" even though there is no direct arrow from A to D. Many kinds of relationships are transitive. For example, time ordering is transitive: if event B occurs after A, and C occurs after B, then C occurs after A. "Is part of" also is transitive: if B is part of A, and C is part of B, then C is part of A. Some relations seem transitive but are not, like the relation "has." Jane may have a husband, and her husband may have a bald head. We would not want to conclude that Jane has a bald head.

Tree diagrams can be represented in matrix form. Figure 2 shows the matrix for the diagram in figure 1. Letters along the top of the matrix show nodes as originating sources for arrows. Letters along the side of the matrix show nodes as destination points for arrows. A zero in a cell indicates that no arrow connects the column node to the row node. A one indicates that an arrow goes from the originating node (column label) to the destination node (row label).

Figure 2

	A	B	C	D	E	F
A	<u>0</u>	0	0	0	0	0
B	1	<u>0</u>	0	0	0	0
C	1	0	<u>0</u>	0	0	0
D	0	1	0	<u>0</u>	0	0
E	0	1	0	0	<u>0</u>	0
F	0	0	1	0	0	<u>0</u>

The major diagonal is emphasized with underlining in figure 2. In a matrix that represents a hierarchical

tree, the major diagonal is always filled with zeros, and nodes can be ordered such that all entries above the major diagonal are zero. A *topographical ordering* of nodes has the node at the top of the graph first (A). Nodes that are directly connected to this node come next (e.g., C and B), then nodes directly connected to these follow (e.g., F, E, D), and so on, until all nodes are listed. A topographical ordering orders the nodes as they are encountered when going from the top of the tree diagram to the bottom. If nodes are listed in topographical order for the matrix representation, then all cells above the major diagonal contain zeros.

A tree diagram can be stored inside a computer in various ways. We could store the matrix representation, but this wastes space on zeros above the main diagonal. Instead we will use the *edge list* representation. This approach stores a tree, including verbal labels for the nodes, in two lists — name and edge — as shown in figure 3.

Figure 3

Name List

- (1) a living creature
- (2) a mammal
- (3) a bird
- (4) a dog
- (5) a cat
- (6) a robin

Edge List

- 1,2
- 1,3
- 2,4
- 2,5
- 3,6

The name list is simply a list of node labels with index numbers implied. Each entry in the edge list corresponds to an arrow on the tree diagram, and the entry consists of two numbers. The first is the index number for the arrow's originating node. The second is the index number for the destination node. The edge list has as many entries as there are arrows on the diagram. Their order is not important.

In the program presented here, index numbers are interpreted as ASCII values and converted to characters. Thereby the edge list can be maintained as a character string, taking advantage of dynamic string allocation in CBM BASIC; we do not have to set aside space for the edge list, whose size is unpredictable beforehand. By adding

64 to each index number before converting to a character, we get ASCII values for letters of the alphabet. For example, the edge list in figure 3 could be represented as the following string:

'ABACBDBECF'

This string contains all the information represented in figures 1 and 2. Marking the string off into pairs of letters precisely defines each arrow in the tree diagram.

We now have a neat, concise way of representing hierarchical knowledge in a CBM microcomputer. Artificial intelligence research usually employs a list representation that calls for a LISP language interpreter, but that would not be as convenient as this approach that works in BASIC.

Elicitation

The next step is to input knowledge from humans into computers. Storing knowledge within programs is not an efficient approach because too many people do not know how to program. Rather, the computer should elicit and store people's knowledge:

- accommodating to a user's interest in a certain kind of relation,
- talking with the user about a given topic,
- helping the user recall topical elements (nodes),
- helping the user to define relations among elements (edges).

Ideally the computer would deal with any kind of relation for making trees and would talk in accordance with rules of grammar and discourse. These kinds of flexibilities are costly in terms of program space, so we compromise. The program here offers three kinds of relationships for analyses, and presents only a limited number of queries.

Requirements for eliciting nodes and edges are more critical. To represent a person's knowledge about a topic, we must get as close as possible to an exhaustive listing of concepts (nodes). Furthermore, we must meticulously examine every possible relationship to assure that all real ones are included.

This program uses several tactics to help a person recall entities in the domain being considered. At the begin-

ning of a session, and periodically thereafter, a general elicitation question is presented, in the form: "What is an entity in the domain being considered?" The user is reminded of all the entities that have been entered already. Once some entities have been specified, these are used to stimulate recall of more entities, using questions like: "What other entities are related to entity X in the domain being considered?" Ultimately, every recorded entity is used as a stimulus for obtaining more entities. Additionally, the user occasionally is asked to name an entity that differentiates some entities from others, in a query like: "X, Y, and Z are entities in the domain of interest; what entity might be implied by two of these?" While no methodology guarantees exhaustive recall of all entities, these techniques do promote extensive recall.

Definition of a new entity's relations is complex because a number of conditions have to be fulfilled in building a tree. These conditions are:

1. As we consider a new entity, we have to allow that it could be subordinate to any existing entity, and/or superordinate to any entity except the top one (which defines the domain of interest). That is, a new node might be positioned anywhere in the tree diagram except above the top node. In principle, this means that for every existing entity except the topmost one, the computer has to ask the user whether the new entity is a superordinate and/or subordinate. Fortunately, principles of logic and transitivity permit economies.
2. The complexity of hierarchical knowledge requires allowing any entity to have multiple subordinates or multiple superordinates.
3. Finally, proper tree structure calls for deleting any relations that can be inferred from transitivity.

The program assumes that a new entity is subordinate to the top node — it is in the domain of interest. Then a series of yes-no questions is asked to determine which existing entities are superordinates or subordinates of the new entity. Each query is of the form: "Does X (an existing entity) imply Y (the new entity)?" "Does Y imply X?"

The entities that are superordinate to the new entity are determined first. Querying works from the top of the tree downward, the procedure considers existing entities in topological order.

Let's say an existing entity is not superordinate to the new entity. Then subordinates of that existing entity are not superordinate to the new entity because of the transitivity principle.

Once a new entity's superordinates are known, more questions are asked to find its subordinates. The procedure employs two logical principles that follow from transitivity. First, if a new entity, Y, is not subordinate to an existing entity, X, then the new entity cannot be superordinate to any of X's subordinates. For example, since a sparrow is not a kind of mammal, various kinds of mammals cannot possibly be kinds of sparrows. Thus, no queries need to be presented regarding subordinates of entities which are not superordinate to the new entity. To find all of a new entity's subordinates, we only need to ask about the subordinates of entities for which the new entity is itself a subordinate.

Second, once we discover that the new entity is superordinate to an existing entity, we know that it is superordinate to all of that entity's subordinates. We do not want to represent those relations directly since they are derivable.

"Knowledge Trees" is written for the PET's 'upgrade' or 3.0 operating system. Notes have been provided to modify it to run on 1.0 and 4.0 systems. With the changes indicated, the program should work on 1.0 PETs. However, to run properly on 4.0 PETs, changes to the machine language portion of the program, beyond those indicated for PET ROM routines, will be necessary.

The problem is in the infamous 'garbage collection' routine. To speed up garbage collection (the process of removing old copies of dynamic strings), the 4.0 ROMs store a back pointer after the actual string characters in high memory. The garbage collection routine checks these back pointers to be sure they point to valid string descriptors in low memory. Bad strings are wiped out. The "Knowledge Trees" machine language routines do not accommodate these back pointers, and as a result, the system is likely to crash when a garbage collection takes place. MICRO will publish a fix for this problem in a future issue.

The transitivity principle eliminates queries about many possible relations, thereby substantially reducing the labor in establishing a new entity's position in the tree diagram. However, the potential for multiple subordinates and multiple superordinates requires that we ask many questions that might seem unnecessary.

To illustrate, suppose that "house pet" is added to the tree represented in figure 3 (abbreviated in figure 1). A house pet is a living creature, but

queries establish that a house pet is not necessarily either a mammal or a bird. We then search for subordinates of house pet among the subordinates of living creature. Mammals are not all house pets, so mammal is not subordinate to house pet. Nevertheless, to avoid error, we must continue searching among the subordinates of mammal. It happens that dogs are a kind of house pet. Thus, in the new tree diagram, dog will be subordinate to both mammal and house pet; dog has two superordinates. Similarly, cat also

Amper-Magic™

**MACHINE LANGUAGE SPEED
WHERE IT COUNTS...
IN YOUR PROGRAM!**

For the first time, Amper-Magic makes it easy for people who don't know machine language to use its power! Now you can attach slick, finished machine language routines to your Applesoft programs in seconds! And interface them by name, not by address!

You simply give each routine a name of your choice, perform the append procedure once at about 15 seconds per routine, and the machine language becomes a permanent part of your BASIC program. (Of course, you can remove it if you want to.)

Up to 255 relocatable machine language routines can be attached to a BASIC program and then called by name. We supply some 20 routines on this disk. More can be entered from magazines. And more library disks are in the works.

These routines and more can be attached and accessed easily. For example, to allow the typing of commas and colons in a response (not normally allowed in Applesoft), you just attach the Input Anything routine and put this line in your program:

xxx PRINT "PLEASE ENTER THE DATE."; : & INPUT,DATE\$

&-MAGIC makes it Easy to be Fast & Flexible!

PRICE: \$75

&-Magic and Amper-Magic are trademarks of Anthro-Digital, Inc.
Applesoft is a trademark of Apple Computer, Inc.

Some routines on this disk are:

- Binary file info
- Delete array
- Disassemble memory
- Dump variables
- Find substring
- Get 2-byte values
- Gosub to variable
- Goto to variable
- Hex memory dump
- Input anything
- Move memory
- Multiple poke decimal
- Multiple poke hex
- Print w/o word break
- Restore special data
- Speed up Applesoft
- Speed restore
- Store 2-byte values
- Swap variables

**Anthro - Digital Software
P.O. Box 1385
Pittsfield, MA 01202**

The People - Computers Connection

is a subordinate of house pet. Having found these subordinates of house pet, we still must continue searching for other subordinates among the other subordinates of living creature. Searching through the bird subgraph yields no more subordinates in this example, but it would if we had canary as an entity.

Once all required queries have been answered, the program positions the new node by linking it to all of its immediate superordinates and all of its immediate subordinates. This is done by adding appropriate entries to the edge-list. If a superordinate and a subordinate of the new entity were originally connected by an arrow in the tree diagram, that relationship is deleted since it now can be derived using the transitivity principle.

Illustrative Analyses

One program option permits the analysis of the kinds of people at a scene. This elicitation provides a way to study people's subjective notions of social structure. The results can be related to issues of role identification, role conflict, and definitions of situations.

When the program has been loaded, the user is instructed on how to use the computer's keyboard. The user is given the instructions in exhibit 1 to read, and a sheet like exhibit 2 showing available commands.

The progression of questions and answers during the first part of a session might be as shown in exhibit 3. You might find a different structure if you did it yourself.

Choosing option 2 in answer to the first question would lead to the elicitation of happenings in an event. For example, analyzing what happens when we dine out might give the final results shown in exhibit 4. This elicitation could precede critical path analyses in order to define all the events that have to be considered. It also is an empirical approach to constructing "frames" for artificial intelligence research.

The BASIC Program

Input

The STOP key is disabled while the program is running (lines 340-350). A special input routine (lines 640-780) uses a shaded cursor to keep the program from ending if the user hits RETURN without making an entry. However, you can still break from a run

by typing SPACE, then RETURN. The STOP key's function is turned on and off again during the input routine, so STOP is available if you break out of the program for programming purposes.

In the introductory part of the program (lines 370-510), a menu allows the user to choose one of three kinds of analysis.

1. *Analysis of a social scene.* This option elicits social roles and orders them by the "kind of" relation.

Specifically, relations among roles are defined by asking "Is an X always a Y?" The name of the scene (the top node) and one role are elicited as part of the opening procedure (lines 880-1010).

2. *Analysis of an event.* Particular happenings are elicited, and then ordered in terms of their temporal priority: "Is X over before Y?" The definitional subroutine (lines 1030-1160) also gets the name of the general event (top node) and one specific happening.

Exhibit 1: Instructions

You can look at people in different ways. For example, the President of the United States is the President, a politician, a man, an adult, a husband, a father, and other things, too. Everyone can be seen in several ways.

How you see people depends a lot on where you see them. You might think of someone on the street as a pedestrian. The same person inside a store would be a customer or a browser or an employee. One idea about a person can lead to another. For example, teachers always are adults, so if people are talking about a teacher, you automatically know they are talking about an adult.

In this session with the microcomputer, you will name the scene you want to analyze. Thereafter the computer will ask you what kinds of people are found there, and how one type relates to another.

The machine begins by asking questions like, "Is a child always an adult?" But the computer learns what you tell it, so later questions are more interesting. In fact, you have to think carefully to answer some of them.

After you've named different types of people, the computer will analyze your answers and present the results in a chart. The chart has a row and a column for each type. Dots show which types are special cases of other types. For example, if professor and teacher were on the chart, a dot might appear in the professor row and the teacher column indicating that a professor is a kind of teacher.

Two dots in one column show what people have in common. For example, teacher and secretary might both have dots in an adult column, indicating that teachers and secretaries have adulthood as a bond between them. Two dots on one row suggest how a person might get confused in the situation. For example, professor might have a dot under teacher and another under researcher, indicating that sometimes a professor might feel torn between acting as a teacher and acting as a researcher.

Answer the first question by typing number 1 and pressing the RETURN key to analyze types of people.

The second question asks what kind of scene you want to analyze. Answer by typing a singular noun, like FAMILY, SCHOOL, or PARTY. Then press RETURN.

The next questions ask for types of persons you might find at the scene. Use just one or two words to name each type. Use nouns rather than adjectives — e.g., DUMMY rather than DUMB. Use singular rather than plural — e.g., TEACHER rather than TEACHERS. Type each name and press RETURN. If you misspell a word, you can correct it with the change command the next time you are asked to enter something.

Pretty soon the computer starts asking yes-no questions like, "Is a teacher always an adult?" Answer by typing Y or N. Answer yes-no questions for the ideal or general case. If you make a mistake while answering a yes-no question, continue answering Y until the yes-no questions are over. Then delete the entry with the change command so you can re-enter it correctly.

After the first few questions, the chart may appear automatically just before you name another type. When you are through looking at the chart, press the RETURN key, and the questions will start again. You also can get the chart by entering # instead of a name.

Enter \$ to skip a question, if you cannot think of a good answer. (The \$ option does not apply when answering yes-no questions.)

3. *Analysis of an entity.* Components of an entity and their incorporation relationships are elicited, using the relational question "Does X definitely have Y?" The name of the entity and of one of its parts are acquired while defining this option (lines 1180-1310).

Wordings of the various elicitation questions are adjusted for each option. You can change the wordings by changing the definitions of the string variables W3\$, W4\$, and W5\$ [e.g., in lines 980-1000].

After these preliminaries, the program drops into a loop which continually cycles through all of the existing nodes in the graph (lines 530-620). The loop has no termination: the program ends when you press SPACE RETURN or turn off the machine.

The subroutine eliciting new entries (lines 1430-1840) ordinarily presents one of the existing entries as a stimulus, asking a question like: "What else might X relate to in the entity?" Existing entries that relate are

listed as a reminder: "Aside from Y, Z?" The new entry is accepted, preceded by "a" or "an" (if that is appropriate), added to the name list, and another subroutine is called to place the new node in the tree.

Instead of entering a word in response to the elicitation question, the user can enter any of the program commands to correct, display, analyze, or save the data.

The main program loop calls a second elicitation subroutine (lines 2730-3010) that checks whether a node has more than two immediate subordinates. If so, another elicitation question is presented, such as: "Here are some parts of a W: X, Y, Z. What is a more general term for some of these?" The user may skip the question. If an answer is provided, it is treated in the same way that new entries ordinarily are handled.

The subroutine to establish a new node's position in the tree structure is the longest in the program (lines 1860-2710). First it clears some short-term memory for storing yes-no answers. Next it determines which existing nodes are superordinate to the new node. Then, working among the subordinates of these superordinates, it searches for existing nodes that are subordinate to the new node. The edge list is modified to reflect the new node's position.

An error-correction subroutine (lines 3680-4570) allows two kinds of changes to be made in the data. An entry in the name list can be changed to a different spelling (lines 3800-3850). Or a node can be deleted entirely from the graph (lines 3870-4570). In the latter case, the program checks whether subordinates are also to be deleted and, if so, deletes them first using the subroutine recursively. Recursion is achieved by defining a push-down 'stack' containing the nodes to be deleted, and letting the procedure work until the stack is empty. Deletion involves removing edges from the edge list and changing other edges in order to close up the graph over the deleted node. Deletion also involves removing the node from the name list and adjusting index numbers.

Output

A subroutine to display the matrix representation of the tree (lines 3130-3660) writes to screen or to printer. The lower triangle of the matrix is presented. Entry names, truncated to 13 characters, are listed in topological

Exhibit 2: Commands

\$ Skip to next question without answering this one.
 # Look at the chart.
 #H Print the chart on paper (requires a printer hooked up to the computer).
 !NAME Change an entry that was entered as NAME. The program will instruct you on how to change the name, or delete the entry entirely.
 @ Analyze commonalities of two or more entries. The program will tell you what more general term covers all of the chosen entries, and in what way each chosen entry is a special case of the more general term.
 ". FILE NAME Save all of the data on tape so that they can be read in again later.
 'FILE NAME Save all data on disk.
 T As an answer to program's first question, this causes the program to read data from tape.
 D As an answer to program's first question, this causes the program to read data from disk.

Exhibit 3: Sample Run

This program allows you to analyze the organization of:
 1. people at a scene
 2. actions in an event
 3. the parts of an entity
 Which analysis do you want to do? 1
 What's a word naming the scene you're going to analyze? FAMILY
 What's one kind of actor you might find at a family scene? FATHER
 Who is an actor at a family scene — aside from a father? MOTHER
 Is a mother always a father? N
 Is a father always a mother? N
 Answer \$ to skip:
 What else might a father be at a family scene? HUSBAND

Exhibit 3 (continued)

Is a husband always a father? N
 Is a husband always a mother? N
 Is a father always a husband? Y
 Is a mother always a husband? N
 The final results of this analysis might look as follows when the matrix representation is printed.

AT A FAMILY SCENE

ADULT X<
 CHILD X+<
 MALE X++<
 FEMALE X+++<
 HUSBAND +X+X+<
 WIFE +X+X+<
 SON ++X X+++<
 DAUGHTER ++X X+++<
 FATHER ++++X+++<
 MOTHER +++++X+++<
 BROTHER ++++++X+++<
 SISTER +++++++X+++<

Exhibit 4: Sample Printout

WHILE DINING OUT

LEAVING RESTA X<
 PAYING BILL +X<
 LEAVING TABLE ++X<
 LEAVING TIP +++X<
 WAITING FOR B ++++X<
 EATING FOOD +++++X<
 GETTING FOOD ++++++X<
 SITTING AT TA +++++++X<
 WAITING FOR F ++++++++X<
 GIVING ORDER ++++++++X<
 READING MENU ++++++++X<
 WAITING FOR M ++++++++X<
 FINDING TABLE ++++++++X+X<
 ENTERING REST ++++++++X+X<

WHILE DINING OUT
 LEAVING RESTAURANT
 PAYING BILL
 LEAVING TABLE
 LEAVING TIP
 WAITING FOR BILL
 EATING FOOD
 GETTING FOOD
 SITTING AT TABLE
 WAITING FOR FOOD
 GIVING ORDER
 READING MENU
 WAITING FOR MENU
 FINDING TABLE
 ENTERING RESTAURANT

order along the lefthand border (the full names are listed separately at the bottom when a hard copy is printed). Cells with zeros are left blank; cells with ones are marked graphically. Screen displays are limited to trees with less than 23 nodes; printed output covers the maximum size tree that the program handles — 63 nodes. The screen display is invoked randomly about one-third of the time after new entries are made (line 1830).

Common superordinates of two or more entries are found by another subroutine (lines 5070-5600). This procedure also indicates the immediate subordinates of the common superordinate. These subordinates lead down to each of the originally specified nodes — an analysis not easily done by inspection of the matrix. The algorithm involves concatenating all the up-graphs of the focus nodes, creating a dummy subordinate node of the focus nodes, and subtracting the up-graph of the dummy from the concatenated graphs (repeatedly if there are more than two focus nodes). The topologically lowest node of the remainder represents a commonality of the focus nodes. A search is made to find the immediate subordinates of the common

node that are linked to focus nodes. When multiple common nodes exist, they all are presented in turn. The dummy node is deleted before the subroutine ends.

The routine to save a knowledge base on tape or disk (lines 4590-4810) creates a file name from the label given as part of the save command, plus the name of the top node in the graph, as follows: LABEL.NODE1. The file contains the type of analysis as specified for the program's first question, the number of nodes, the node names, the number of edges, and the list of edges. The file can be read later by another routine (lines 4830-5050), called by typing T or D in answer to the first program question [data are listed as they are read into the program]. Writing and reading are done with tape unit #1 or disk unit #0. A disk must be properly initialized before the program is run.

Utilities

Articles are appended to the front of entries for readability (lines 1330-1410). This routine inserts a nonprinting character with 'a' to make the appendage uniform in length, simplifying removal whenever necessary. Articles are not

added to event entries (option 2 in the program).

A subroutine (lines 800-860) presents the frequent query about relationships (e.g., "Is an X always a Y?"). The routine gets the answer and returns. A separate routine (lines 3030-3110) is used to add a name to the name list.

Loader Program

A separate loading program is shown in listing 2. This should be saved as the first program on a tape or disk. When it is loaded and RUN, it automatically loads the assembly language subroutines, guards the memory allocated to them, and then loads the main BASIC program and starts it running. On tape, the programs must be saved in the following order: loader (named anything); the file of assembly language subroutines, named CODE; main BASIC program, named TREES. These same file names must be implemented in order to use the loader program with a disk.

Entering the BASIC Code

A Glossary of special symbols used in the program listings is given at the
(Continued on page 74)

OSI COMPATIBLE HARDWARE

IO-CA10X SERIAL PORT \$125
ACIA based RS-232 serial printer port. DIP SWITCH selectable baud rates of 300-9600. Handshaking (CTS) input line is provided to signal the computer when the printer buffer is full. Compatible with OS-65U V1.2 and OS-65D.

IO-CA9 PARALLEL PORT \$175
Centronics Standard Parallel printer interface for OSI computers. The card comes complete with 10 ft. of flat ribbon cable. Compatible with OS-65D and OS-65U software.

IO-CA9D DIABLO PARALLEL PORT \$175
DIABLO 12 BIT WORD Parallel port for use with word processor type printers. Complete with 10 ft. cable. Compatible with OS-65U software.

IO-LEVEL 3 MULTI-USER EXPANSION \$450
Provides 3 printer interfaces currently supported by OSI-Serial, Centronics Parallel, Diablo Parallel. 4K of memory at D000 for Multi-user executive. 4 Port serial cluster. The LEVEL 3 card allows expansion of an OSI C3 machine up to 4 users with appropriate additional memory partitions.

24MEM-CM9... \$300 **16MEM-CM9... \$300** **8 MEM-CM9... \$210**
24K memory card is available at 3 different populated levels. All cards are fully socketed for 24K of memory. The card uses 2114-300ns chips. DIP SWITCH addressing is provided in the form of one 16K block and one 8K block. Also supports DIP SWITCH memory partition addressing for use in multi-user systems.

FL470 FLOPPY DISK CONTROLLER \$100
OSI-Type floppy disk controller and real time clock. Will Support 5 1/4" or 8", Single or double-sided drives. Requires drives with separated data and clock outputs.

BIO-1600 BARE IO CARD \$50
Super I/O Card. Supports 8K of 2114 memory in two DIP SWITCH addressable 4K blocks. 2 16 Bit Parallel Ports may be used as printer interfaces, 5 RS-232 Serial Ports with CTS & RTS handshaking. With manual and Molex connectors.

8MEM-CM9 BARE MEMORY CARD \$50
Bare 24K memory card, also supports OSI-type real time clock and floppy disk controller. With manual and Molex connectors.

#96 PROTOTYPE CARD \$35
Prototype board holds 96 14 or 16 pin IC's. Will also accommodate 18, 24, or 40 pin IC's. Row and column zone markings, easy layout. 1/8" epoxy glass P.C. board.

C1P-EXP EXPANSION INTERFACE \$65
Expansion for C1P600 or 610 boards to the OSI 48 Pin Buss. Uses expansion socket and interface circuitry to expand to 48 Pin Backplane. Requires one slot in backplane.

BP-580 BACKPLANE \$47
Assembled 8-slot backplane with male Molex connectors and termination resistors.

DSK-SW DISK SWITCH \$29
A circuit when added to OSI Minifloppy systems extends the life of drives and media. Accomplish this by shutting off Minifloppy Spindle motor when system is not accessing the drive. Complete KIT and manual.

D&N MICRO PRODUCTS, INC.
3684 N. Wells Street Ft. Wayne, Indiana 46808
219/485-6414
TERMS: Check or money order Add \$2 Shipping, Outside U.S. add 10%.

PRINT II

THE ULTIMATE PRINT STATEMENT FOR THE APPLE II COMPUTER.

Standard Apple Output	PRINT II Output
\$1234.56	\$1234.56
\$12	\$ 12.00
\$1.9	\$ 1.90
\$.75	\$.75
\$765	\$ 765.00

Print "\$"; I (without Print II) Print #I (Field Width 8 with 2 decimal places.)

Updates Applesoft™ **PRINT** to accept:

- User-specified print formatting.
- X, Y cursor positioning.




Simplifies tabular displays.

Transparent to user—not affected by re-booting.

IF YOUR TIME IS WORTH MONEY, THEN YOU NEED PRINT II.

Available at your local dealer for \$24.95 or write to:

Computer Systems Design
2139 Jackson Blvd.
Rapid City, SD 57701
(605) 341-3662

"NIBBLE® IS TERRIFIC" (For Your Apple)



NIBBLE IS: *The Reference for Apple computing!*

NIBBLE IS: One of the Fastest Growing Magazines in the Personal Computing Field.

NIBBLE IS: Providing Comprehensive, Useful and Instructive Programs for the Home, Small Business, and Entertainment.

NIBBLE IS: A Reference to Graphics, Games, Systems Programming Tips, Product News and Reviews, Hardware Construction Projects, and a host of other features.

NIBBLE IS: A magazine suitable for both the Beginner and the Advanced Programmer.

Each issue of NIBBLE features significant new Programs of Commercial Quality. Here's what some of our Readers say:

- "Certainly the best magazine on the Apple II"
- "Programs remarkably easy to enter"
- "Stimulating and Informative; So much so that this is the first computer magazine I've subscribed to!"
- "Impressed with the quality and content."
- "NIBBLE IS TERRIFIC!"

In coming issues, look for:



- Stocks and Commodities Charting
- Assembly Language Programming Column
- Pascal Programming Column
- Data Base Programs for Home and Business
- Personal Investment Analysis
- Electronic Secretary for Time Management
- The GIZMO Business Simulation Game

And many many more!

NIBBLE is focused completely on the Apple Computer systems.

Buy NIBBLE through your local Apple Dealer or subscribe now with the coupon below.

Try a NIBBLE!

nibble  

We accept Master Charge & Visa

Box 325, Lincoln, MA. 01773 (617) 259-9710

I'll try nibble!
Enclosed is my \$19.95 (for 8 issues) Price effective Jan. 1, 1982
(Outside U.S., see special note on this page.)

check money order

Your subscription will begin with the next issue published after receipt of your check/money order.

Card # _____ Expires _____

Signature _____

Name _____

Address _____

City _____

State _____ Zip _____

NOTE.

- Domestic U.S. First Class subscription rate is \$36.50
- Canada Air Mail subscription rate is \$42.50
- Outside the U.S. and Canada Air mail subscription rate is \$47.50

All payments must be in U.S. funds drawn on a U.S. bank.

©1980 by MICRO-SPARC, INC. Lincoln, Mass. 01773. All rights reserved.
 *Apple is a registered trademark of Apple Computer Company.

(Continued from page 72)

end of the main program (lines 5620-5700).

REMARKS must be ignored when entering the main BASIC program into a 16K PET because all of the lines in listing 1 take up 18K. REMARKS could be included when using a 32K PET, but then the assembly language subroutine would have to be relocated.

6502 Assembly-Language Routine

A number of analytic functions are coded in 6502 assembly language for the sake of speed. The code is presented in listing 3, which begins with definitions of zero-page cells and the ROM subroutines used. These locations refer to Operating System 3.0 on the PET. Notes are included, where appropriate, for conversion to O.S. 1.0 and 4.0.

The short routine at lines 370-460 (called before the BASIC program is loaded) points the USR function to the code and protects the code from BASIC. The overall logic of the procedure can be seen in the executive program (lines 510-560).

A number of parameters have to be established every time the routine is called (lines 610-1630). The number of nodes is determined by finding the name list and counting the number of entries in it. The routine looks for a specific array, so the name list *must* be NM\$(). Next the routine finds the location and the length of the edge list, which *must* be named L\$(0). The name list and the edge list — and later two graphs named GR\$(0) and BR\$(0) — are found using a separate subroutine (lines 1100-1630). If an array is not found, a pointer is set to the word "array" among the PET's canned messages, and control is transferred to the standard error routine in ROM.

Three parameters are passed from BASIC when the USR function is invoked: the starting node from which to begin tracing a subgraph, a code indicating whether to trace upwards (0) or downwards (2), and a code indicating whether the list of encountered nodes should (1) or should not (0) contain the same element if the trace goes through it repeatedly. The no-repeat option usually is used in this application. The three numbers are combined into a single argument for USR; the starting node is added to the two code values, each multiplied by 256. The USR function transfers this number to FACC — the BASIC accumulator — in five-byte floating point format. After the number

has been converted to two-byte integer format (using the ROM routine, FLPINT), the parameters are recovered. The low byte of the argument equals the specified starting node. The high byte of the argument is ANDed with one to recover the Repeat code. The low byte is ANDed with two to recover the Up-Down code.

Lines 1670-2280 create a topological list of node index numbers, using the second tape cassette buffer as a work space. The list is not returned directly; it is used to order nodes encountered while tracing a subgraph. The procedure follows an algorithm presented by Gotlieb and Gotlieb (1978). If a loop is encountered in the graph, then the routine aborts and prints "complex error."

The routine in lines 2330-3050 makes a list of all the nodes in a graph that are reachable, either in an upward direction or a downward direction, from the specified starting node. The index numbers are converted to characters and stored in a string named GR\$(0), which must be dimensioned in the BASIC program. The length of GR\$(0) is returned as the value of the USR function.

The GR\$(0) list is put into topological order by lines 3090-3350. If the starting node is specified as one, with direction down, then GR\$(0) will return with a topological listing of all nodes in the graph.

The final part of the procedure, lines 3390-3760, removes elements in the GR\$(0) list from another list of elements, BR\$(0). This array must be dimensioned in the BASIC program. If none of the members of GR\$(0) is in BR\$(0), then BR\$(0) remains unchanged. If elements of GR\$(0) are in BR\$(0), then they are removed, making BR\$(0) shorter. BR\$(0) must always be defined with a space following it (e.g., see line 2070 of the BASIC program).

Every USR call performs all of these functions, even if some of them are not used.

Entering the Code

The procedure has been assembled for placement at the top of RAM in a 16K computer, and the same positioning can be used with a 32K machine. The positioning can be changed to the top of 32K RAM by changing every hex memory address in the \$3000 range so that it begins with seven instead of three. Enter the code with the CBM monitor, display the relevant cells with the M command, and overwrite the contents following listing 3. Use the following monitor command to save to disk.

```
.S "0:CODE",08,3DA8,3FF0
```

Use the following command to save on tape.

```
.S "CODE",01,3DA8,3FF0
```

Bibliography

1. C.C. Gotlieb and L.R. Gotlieb, *Data Types and Structures*, [Prentice-Hall, 1978].
2. J.P. Spradley and D.W. McCurdy, *The Cultural Experience*, [Science Research Associates, 1971].
3. P.H. Winston, *Artificial Intelligence*, [Addison-Wesley, 1979].

David Heise is Professor of Sociology at Indiana University. He recently edited "Microcomputers in Social Research" (Sage Publications, 1981). His books include *Causal Analysis and Understanding Events*. He has created commercial microcomputer programs for word processing and statistical analysis, and he has published a number of programs for PET computers. Contact Mr. Heise at the Department of Sociology, Indiana University, Bloomington, IN 47405.

Listing 2: Loader Program

```
100 : REM: LOADER PROGRAM
110 :
120 : REM: ROUTINE TO LOAD THE ASSEMBLED CODE, THEN THE MAIN PROGRAM.
130 DV$=" ,8";CU$="00000"
140 REM DV$="";CU$=CU$+"0" :REM ADD THIS LINE TO LOAD FROM TAPE.
150 : REM: SET GRAPHICS MODE.
160 POKE 59468,12
170 : REM: STORE 4 CARR. RETURNS IN INPUT BUFFER,
      (USE 525 AND 527-530 FOR 1.0)
180 POKE 158,4:POKE 623,13:POKE 624,13:POKE 625,13:POKE 626,13
190 : REM: SET UP SCREEN TO INVOKE THE SUBROUTINE FILE -- "CODE".
200 PRINT "LOAD" CHR$(34) "CODE" CHR$(34):PRINTCU$;SYS 15784
210 : REM: THEN LOAD PROGRAM "MAIN".
220 PRINT "LOAD" CHR$(34) "MAIN" CHR$(34):DV$
230 PRINT CU$ "RUNG";:END
```

OWING SCIENTIFIC

USERS!

READ . . .

PEEK (65)

The Unofficial OSI Users Journal

THE WORLD WIDE PUBLICATION
EXCLUSIVELY DEDICATED TO OSI USERS!

- Hardware Mods.
- Software Exchange
- Peeks and Pokes
- Software Reviews
- Bugs and Fixes

SEND \$15.00 FOR 12 ISSUES TO:
PEEK (65) P.O. BOX 347, OWINGS MILLS, MD 21117 (301) 363-3267

Maryland Subscribers Add 5% Tax

Inquire for Foreign Rates

Gosub International, Inc.

GOSUB'S MX80/70 Friction Feed Kit \$49.95 ppd

- Use Single Sheet Paper
- Use Your Own Letterhead

- Use Inexpensive Roll Paper •
- Does Not Affect Pin-feed Use •

PRINTERS

PRINTERS

PRINTERS

EPSON MX80...\$545 ppd

EPSON MX100...\$825 ppd

THE PRINTER STAND \$29.95

SAVE MONEY AND SPACE—Holds almost any printer that uses 9½"-size paper or forms. Made from ¼" acrylic, allows 3" of paper or forms to be stacked under printer. Great for use with bottom load printers! Fits MX80/70-Microline 80's BASE 2-850 Bytewriter-1 and many more!

AUTHORIZED DEALERS FOR THE GOSUB FRICTION FEED KIT

USA—Alpine Computer Center, 2526 S. Alpine Rd., Rockford, Ill. 61108, (815) 229-0200; **OMEGA Micros**, 3819 West Michigan St., Milwaukee, WI 53208, (414) 933-6088; **EXATRON**: 181 Commercial St., Sunnyvale, CA 94086, (800) 538-8559/CA, (408) 737-7111.

CANADA—Microwest Dist. Ltd., 1940 Larson Rd., North Vancouver, B.C., Canada V7M 2Z8, (604) 988-9998 or 988-6877.

AUSTRALIA—Computer Campus Pty. Ltd., 11 Rundle Street, Kent Town, South Australia, 5067, ph. 08.424826, telex 88156.

STRATEGY SIMULATIONS FOR THE TRS-80

MERCENARY FORCE

Command a mercenary army. Decide the number of men, type of weapons, armor, air support, medical aid and transports. Battle the enemy in jungles, underwater, on moons and in space. (1-4 players)

TRS-80 L2/16K \$16.95

SPACE MERCHANT

Build an Empire in the stars. Choose your cargo, means of shipping, and security. Risk pirates, ion storms, engine failure and other hazards while you try to become a Space Merchant. (1-6 players)

TRS-80 L2/16K \$9.95

80 SPACE RAIDERS

Starship Simulator and Combat Game

This is the most realistic combat space simulation available for any computer system anywhere! Operator is placed in a rapidly moving ship looking out through a window into the depths of space. Thousands of stars appear in the distance and part toward the screen edges as the ship flashes past. Forward and aft views available at any time. Accurate celestial navigation using both XYZ and spherical coordinates! Comment from Aug. '81 *80 Microcomputing*: "May be the ultimate Star Trek-type game and, possibly, the most complex micro-computer game ever attempted"! Fast action graphics never stop!

Model 1/Level 2/16K \$24.95

SOFTWARE

Software for the VIC 20 and Color Computer

SPACE MERCHANT—See Strategy Simulations for the TRS-80

WARRIOR—Head to head combat in a game where you and your opponent are the rulers of warring city-states, each trying to destroy the other. You decide how many of your population will be warriors, farmers, factor workers, and scholars. **\$9.95**

YAHTZEE I

\$10.95

GAME PACK #1 Quarter Horse and Yahtzee **\$14.95**

GAME PACK #3 Warrior and Space Merchant **\$14.95**

Kansas residents add 3% sales tax
All prices subject to change without notice
FREE CATALOG UPON REQUEST
(TRS-80 is a trademark of Tandy Corp.)
Dealer Inquiries Invited
SEND CHECK OR MONEY ORDER



Gosub Int'l., Inc.
501 E. Pawnee, Suite 430
Wichita, KS 67211
(316) 265-9858

GOSUB International (New England Division)
P.O. Box 2566
Framingham, MA 01701
(add 5% sales tax)

GOSUB
P.O. Box 275
Wichita, KS 67201
(316) 265-9992

Tree Analysis (Continued)

```

1470 PRINT
1480 : REM: USE A SPECIAL QUESTION IF USING THE TOP NODE FOR ELICITATION.
1490 IF IP=1 THEN PRINT W$NM$(1):GOTO 1540
1500 : REM: OTHERWISE PRINT THE STANDARD STEM
1510 PRINT "WHAT ELSE MIGHT
1520 PRINT RIGHTING (NM$(IP),LEN(NM$(IP))-1)W4NM$(1)
1530 : REM: SHOWING PRESENT LINKS, IF ANY.
1540 IO=LEN(L$(0)):FOR I=1 TO IO STEP 2
1550 IF FN(L(I)=IP OR (FN(L(I+1)=IP AND FN(L(I)<>I) THEN I=IO:NEXT I:GOTO 1570
1560 NEXT I:GOTO 1640
1570 PRINT "----ASIDE FROM
1580 BR$(0)=" ",LL=USR(BI+DOWN)
1590 FOR J=1 TO LL
1600 IF FNG(J)>1 AND FNG(J)<>IP THEN PRINT " NM$(FNG(J))
1610 NEXT J
1620 IF DI THEN DI=0:LL=USR(IP+UP):GOTO 1590
1630 : REM: GET USER'S INPUT.
1640 GOSUB 670:AS=LEFT$(WD$,1)
1650 : REM: ALLOW SKIPPING.
1660 IF AS=" " THEN RETURN
1670 : REM: ACCEPT A PRINT-GRID COMMAND.
1680 IF AS="*" THEN GOSUB 3160:GOTO 1450
1690 : REM: ACCEPT REVISE COMMAND.
1700 IF AS="!" THEN GOSUB 3700:GOTO 1450
1710 : REM: ACCEPT SAVE COMMAND.
1720 IF AS="." OR AS="@" THEN GOSUB 4620:GOTO 1450
1730 : REM: ACCEPT ANALYZE COMMAND.
1740 IF AS="0" THEN 5090:GOTO 1450
1750 : REM: THIS IS AN ENTRY!
1760 : REM: ADD AN ARTICLE TO THE WORD.
1770 GOSUB 1340
1780 : REM: ADD THE NODE TO NM$.
1790 H=L:GOSUB 3060:IF H=L THEN PRINT "<R>ALREADY EXISTS":GOTO 1450
1800 : REM: FIND THE NODE'S LINKAGES.
1810 GOSUB 1880
1820 : REM: GIVE AUTOMATIC PRESENTATION OF THE GRID OCCASIONALLY.
1830 IF RND(1)<.3 THEN WD$="*":GOSUB 3160
1840 RETURN
1850 :
1860 : REM: SUBROUTINE TO POSITION NEW NODE IN TREE.
1870 : REM: CLEAR SPACE FOR REMEMBERING YES-NO ANSWERS.
1880 FOR I=1 TO 65:ODD(I)=0:NEXT I
1890 : REM: GET A DOWN GRAPH FROM THE ORIGIN, WITHOUT REPEATS.
1900 BR$(0)=" ":LG=USR(I+DOWN)
1910 : REM: IS NEW NODE SUBORDINATE TO AN EXISTING NODE?
1920 FOR IB=1 TO LG
1930 : REM: STOP IF BEYOND THE LENGTH OF CURRENT GR$.
1940 IF IB>LEN(GR$(0)) THEN IB=LG:GOTO 2140
1950 : REM: AND SKIP THE QUESTION FOR THE ORIGIN.
1960 IF FNG(IB)=1 THEN NEXT IB
1970 : REM: IF QUESTION WAS ASKED FOR THIS NODE, RECALL THE ANSWER.
1980 IF ODX(FNG(IB))>0 THEN AS=CHR$(ODX(FNG(IB))):GOTO 2040
2000 W1$=NM$(ZZ):W2$=NM$(FNG(IB)):GOSUB 810
2010 : REM: (REMEMBER THE ANSWER.)
2020 ODX(FNG(IB))=ASC(AS)
2030 : REM: IF NEW NODE IS NOT SUBORDINATE TO A NODE,
2040 IF AS<>"N" THEN NEXT IB:GOTO 2160
2050 : REM: THEN REMOVE THAT NODE AND ITS SUBORDINATES FROM THE DOWN GRAPH.
2060 : REM: BY SETTING BR$(0) EQUAL TO A DUPLICATE OF GR$,
2070 BR$(0)=LEFT$(GR$(0)+",",LEN(GR$(0)))
2080 : REM: REMOVING UNWANTED NODES FROM BR$,
2090 LL=USR(FNG(IB)+DOWN)
2100 : REM: THEN SETTING GR$ EQUAL TO BR$.
2110 GR$(0)=LEFT$(BR$(0),LEN(BR$(0)))
2120 : REM: RELOOK AT THE REVISED GR$.
2130 IB=1
2140 NEXT IB
2150 : REM: CLEAR SPACE FOR REMEMBERING YES-NO ANSWERS.

```

Tree Analysis (Continued)

```

2160 FOR I=1 TO 65:ODD(I)=0:NEXT I
2170 : REM: CREATE A DUPLICATE OF THE CURRENT PRIMARY GRAPH.
2180 GR$=LEFT$(GR$(0),LEN(GR$(0)))
2190 : REM: WORK BACKWARD IN THE PRIMARY GRAPH UNTIL ALL NODES ARE DELETED.
2200 IB=LEN(GR$(0)):IF IB=0 THEN RETURN
2210 : REM: MAKE A LINK IN EDGE LIST FROM THIS NODE TO NEW NODE.
2220 L$(0)=L$(0)+RIGHT$(GR$(0),1)+CHR$(ZZ+64)
2230 : REM: GET THE DOWN-GRAPH FROM NODE IB.
2240 BI=FNG(IB)
2250 BR$(0)=" ",LL=USR(BI+DOWN)
2260 : REM: LOOK AT NEXT NODE IN THE DOWN GRAPH,
2270 FOR BB=1 TO LL
2280 : REM: UNTIL ALL ARE CONSIDERED.
2290 IF BB>LEN(GR$(0)) THEN NEXT BB:GOTO 2690
2300 : REM: SKIP THE NEW NODE OR NODE IB.
2310 IF FNG(BB)=ZZ OR FNG(BB)=BI THEN NEXT BB:GOTO 2690
2320 : REM: IF QUESTION WAS ASKED FOR THIS NODE, RECALL ANSWER.
2330 K=ODX(FNG(BB)):IF K<>0 THEN AS=CHR$(K):GOTO 2390
2340 : REM: OTHERWISE ASK IF THIS NODE IS SUBORDINATE TO THE NEW NODE.
2350 W1$=NM$(FNG(BB)):W2$=NM$(ZZ):GOSUB 810
2360 : REM: (REMEMBER THE ANSWER.)
2370 ODX(FNG(BB))=ASC(AS)
2380 : REM: WHEN THIS NODE, BB, IS SUBORDINATE TO NEW NODE,ZZ,
2390 IF AS="N" THEN NEXT BB:GOTO 2690
2400 : REM: SEE IF BB NODE IS DIRECTLY CONNECTED TO IB NODE.
2410 AS=LEFT$(GR$(0),1)+MID$(GR$(0),BB,1)
2420 FOR BC=1 TO LEN(L$(0)) STEP 2
2430 B$=MID$(L$(0),BC,2)
2440 : REM: IF IT IS NOT, THEN ADD EDGE FROM NEW NODE TO BB NODE
2450 IF AS=B$ THEN 2510
2460 NEXT BC
2470 : REM: (PROVIDING IT'S NOT ALREADY PRESENT).
2480 IF K<>ASC("Y") THEN L$(0)=L$(0)+CHR$(ZZ+64)+MID$(GR$(0),BB,1)
2490 GOTO 2610
2500 : REM: IF IT IS, SUBSTITUTE EDGE ZZ-BB FOR OLD EDGE IB-BB
2510 C$=CHR$(ZZ+64)+MID$(L$(0),BC,1)
2520 : REM: (PROVIDING IT'S NOT ALREADY THERE).
2530 FOR IO=1 TO LEN(L$(0)) STEP 2:IF MID$(L$(0),IO,2)=C$ THEN C$=""
2540 NEXT IO
2550 AS="" : IF BC>1 THEN AS=LEFT$(L$(0),BC-1)
2560 B$="" : IF BC<LEN(L$(0))-2 THEN B$=RIGHT$(L$(0),LEN(L$(0))-BC-1)
2570 : REM: (CLOSE THE BC LOOP.)
2580 BC=LEN(L$(0)):NEXT BC
2590 L$(0)=AS+C$+B$
2600 : REM: REMOVE BB AND ITS SUBORDINATE NODES FROM THE IB DOWN-GRAPH,
2610 BR$(0)=LEFT$(GR$(0),LEN(GR$(0)))
2620 LL=USR(FNG(BB)+DOWN)
2630 GR$(0)=LEFT$(BR$(0),LEN(BR$(0)))
2640 : REM: AND RELOOK AT IT.
2650 BB=BB-1
2660 NEXT BB
2670 : REM: IF ALL SUBORDINATES OF IB ARE DONE. RETURN TO PRIMARY GRAPH.
2680 : REM: REMOVE NODES UPWARD FROM IB.
2690 BR$(0)=LEFT$(GR$+" ",LEN(GR$)):LL=USR(BI+UP)
2700 : REM: AND CONTINUE.
2710 GR$(0)=LEFT$(BR$(0),LEN(BR$(0))):GOTO 2180
2720 :
2730 : REM: SUBROUTINE TO ELICIT GROUPING WORD.
2740 : REM: WHEN CURRENT NODE HAS 3+ IMMEDIATE SUBORDINATES,
2750 J=0:FOR I=1 TO LEN(L$(0)) STEP 2:IF CHR$(IP+64)=MID$(L$(0),I,1) THEN J=J+1
2760 NEXT I:IF J<3 THEN RETURN
2770 : REM: PRINT ELICITATION STIMULUS, ADJUSTING FOR KIND OF ANALYSIS.
2780 I=3:W1$="":JON TA GOTO 2790,2810,2830
2790 PRINT:PRINT "HERE ARE KINDS OF ";IF IP=1 THEN W1$="PARTICIPANTS":I=0
2800 GOTO 2840
2810 PRINT:PRINT "HERE ARE ASPECTS OF ";IF IP=1 THEN W1$="WHAT HAPPENS":I=0
2820 GOTO 2840
2830 PRINT:PRINT "HERE ARE SOME PARTS":W1$=" OF":IF IP=1 THEN W1$="OF":I=0

```

Tree Analysis (Continued)

```

3530 JJ=LL:NEXT JJ,JO
3540 : REM: WHEN LINE IS FINISHED, PRINT IT.
3550 FOR JO=1 TO IO:PRINT#1,CHR$(ODZ(JO));NEXT JO
3560 : REM: CONTINUE THROUGH ALL LINES.
3570 NEXT IO
3580 : REM: IF DOING HARD COPY, PRINT OUT NAMES.
3590 IF DV=4 THEN PRINT#1:FOR IO=1 TO IL:PRINT#1,NM$(FNG(IO));NEXT IO
3600 : REM: CLOSE THE PRINT FILE, AND PRINT A MESSAGE.
3610 CLOSE 1
3620 PRINT"<S>";PRINT" <R>TYPE RETURN";PRINT" <R> TO CONTINUE
3630 : REM: WAIT FOR USER RESPONSE.
3640 GET A$:IF A$="" THEN 3640
3650 : REM: CLEAR SCREEN AND RETURN TO ELICITATION.
3660 PRINT "<S>";:RETURN
3670 :
3680 : REM: SUBROUTINE TO REVISE A NODE.
3690 : REM: REMOVE : FROM FRONT OF WORD, ADD ARTICLE, AND FIND IN NM$ LIST.
3700 WD$=RIGHT$(WD$,LEN(WD$)-1);K=IL:GOSUB 1340:GOSUB 3060
3710 : REM: ABORT IF WORD WAS NOT IN NM$ LIST.
3720 IF K<IL THEN NM$(IL)="" : IL=IL-1:PRINT "<R>NOT FOUND":RETURN
3730 PRINT "<S>FOUND";PRINT "<R>NM$(ZZ)
3740 : REM: PRINT MENU.
3750 PRINT "YOU CAN <R><R>CHANGE THE NAME,
3760 PRINT " OR <R><R>DELETE THE WORD ENTIRELY,
3770 PRINT " <R><R>EXIT.
3780 GET A$:IF A$="" OR (A$<"C" AND A$>"E") THEN 3770
3790 : REM: GO TO SUBROUTINE FOR DELETION.
3800 IF A$="E" THEN RETURN
3810 IF A$="D" THEN GOSUB 3890:RETURN
3820 : REM: DO NAME CHANGE.
3830 PRINT "<S>";:GOSUB 670
3840 : REM: ADD ARTICLE AND REPLACE.
3850 GOSUB 1340:NM$(ZZ)=WD$
3860 : REM: SUBROUTINE TO DELETE A NODE.
3870 : REM: PUT NODE TO BE REMOVED ON 'STACK'.
3880 ODZ(0)=1:ODZ(1)=ZZ
3890 : REM: GET LAST NODE, X, ON STACK.
3900 X=ODZ(ODZ(0))
3910 X=ODZ(ODZ(0))
3920 : REM: DEFINE ITS DOWN-GRAPH,
3930 BR$(0)="" : LQ=USR(X+DOWN)
3940 : REM: WITHOUT THE X NODE.
3950 GR$(0)=RIGHT$(GR$(0),LG-1)
3960 : REM: SAVE GR$.
3970 GG$=LEFT$(GR$(0),LEN(GR$(0)))
3972 : REM: WHEN X HAS SUBORDINATES,
3974 IF LEN(GR$(0))=0 THEN 4270
3980 : REM: CONSIDER THE FIRST OF THEM.
3990 J=FNG(1)
4000 : REM: ASK IF NODE J IS TO BE KEPT.
4010 PRINT:PRINT "RETAIN <R>RIGHT$(NM$(J))-4)"<R>? ";:GOSUB 830
4020 : REM: IF NOT, DELETE J RECURSIVELY, THEN START OVER ON X.
4030 IF A$="N" THEN ODZ(0)=ODZ(0)+1:ODZ(ODZ(0))=J:GOSUB 3910:GOTO 3910
4040 BR$(0)="" : LQ=USR(J+UP+256)
4050 : REM: DELETE EDGE FROM X TO J.
4060 K=LEN(L$(0))
4070 FOR I=1 TO K STEP 2
4080 B$=MID$(L$(0),I,2)
4090 IF A$<>B$ THEN NEXT I:GOTO 4210
4100 A$="" : IF I>1 THEN A$=LEFT$(L$(0),I-1)
4110 B$="" : IF I<K-1 THEN B$=RIGHT$(L$(0),K-I-1)
4120 L$(0)=A$+B$
4130 I=I+1:NEXT I
4140 : REM: AND USE UP-LINKS OF X TO DEFINE UP-LINKS OF J.

```

Tree Analysis (Continued)

```

2840 PRINT W$RIGHT$(NM$(IP),LEN(NM$(IP))-1)" :
2850 : REM: PRINT IMMEDIATE SUBORDINATES OF NODE IP.
2860 FOR I=1 TO LEN(L$(0)) STEP 2
2870 IF CHR$(IP+64)=MID$(L$(0),I,1) THEN PRINT " NM$(FNG(I+1))
2880 NEXT I
2890 : REM: PRINT ELICITATION QUESTION AND GET ANSWER.
2900 PRINT:PRINT " WHAT IS A MORE GENERAL TERM FOR
2910 PRINT " SOME OF THESE (<R>=<R>):SKIP";:GOSUB 670
2920 : REM: PERMIT SKIP.
2930 IF WD$="" THEN RETURN
2940 : REM: ACCEPT PRINT-GRID COMMAND.
2950 IF LEFT$(WD$,1)="" THEN GOSUB 3160:GOTO 2750
2960 : REM: ADD AN ARTICLE TO NEW WORD, ENTER IN NM$, AND LINK IT INTO TREE.
2970 GOSUB 1340:H=L:GOSUB 3060:IF H=IL THEN PRINT"<R>ALREADY EXISTS":GOTO2750
2980 GOSUB 1880
2990 : REM: SHOW GRID.
3000 GOSUB 3160
3010 RETURN
3020 :
3030 : REM: SUBROUTINE TO ADD A WORD TO THE LIST OF WORDS, NM$.
3040 : REM: IF WORD ALREADY IS IN NM$
3050 : REM: THEN CLOSE LOOP AND RETURN.
3060 FOR JJ=2 TO IL
3070 IF WD$=LEFT$(NM$(JJ),LEN(WD$)) THEN ZZ=JJ:JJ=IL:NEXT JJ:RETURN
3080 NEXT JJ
3090 : REM: OTHERWISE ADD THE WORD TO NM$.
3100 ZZ=JJ:IL=JJ:NM$(JJ)=WD$
3110 RETURN
3120 :
3130 : REM: SUBROUTINE TO DISPLAY RESULTS ON GRID.
3140 : REM: ANSWER '##' GIVES HARD COPY.
3150 : REM: SET DEVICE # FOR SCREEN OR PRINTER.
3160 DV=3:IF MID$(WD$,2,1)="" THEN DV=4
3170 : REM: SCREEN OUTPUT LIMITED TO 22 NODES.
3180 IF DV=3 AND MS>0 AND IL>22 THEN RETURN
3190 : REM: IF ON THE 23RD NODE, PRINT ONE-TIME MESSAGE.
3200 MS$="" : TOO MANY ENTRIES TO DISPLAY GRID ON
3210 IF MS=0 AND IL>23 THEN PRINT MS$:MS=1:RETURN
3220 : REM: SET GRAPHIC CHARACTERS FOR SCREEN OR PRINTER.
3230 LN$=CHR$(219):MK$=CHR$(209):DG$=CHR$(95)
3240 IF DV=4 THEN LN$="" : MK$="" : DG$=""
3250 : REM: OPEN PRINTING FILE.
3260 OPEN 1,DV
3270 : REM: CLEAR SCREEN AND PRINT HEADER.
3280 PRINT"<S>";
3290 PRINT#1,MID$(NM$(1),2):PRINT#1
3300 : REM: SEARCH DOWN TO GET ORDERED LIST OF NODES (GR$) WITHOUT REPEATS.
3310 BR$(0)="" : LQ=USR(1+DOWN)
3320 : REM: FOR EACH WORD IN TOPOLOGICAL ORDER,
3330 FOR IO=2 TO IL
3340 : REM: REMOVE ARTICLE,
3350 W$=MID$(NM$(FNG(IO)),5)
3360 : REM: AND TRUNCATE IF TOO LONG.
3370 W$=MID$(W$,1,13)
3380 : REM: THEN PRINT ON NEXT ROW, FOLLOWED BY SPACES TO EDGE OF GRID.
3390 PRINT#1:PRINT#1,LEFT$(W$+SP$,15)
3400 : REM: GENERATE GRID CHARACTERS.
3410 FOR JO=1 TO IO-1:ODZ(JO)=ASC(LN$):NEXT JO:ODZ(IO)=ASC(DG$)
3420 : REM: SET SPACING INDEX.
3430 JL=1
3440 : REM: GO THROUGH RECEIVER NODES IN EDGE-LIST TO FIND THIS NODE.
3450 FOR JO=2 TO LEN(L$(0)) STEP 2
3460 IF MID$(L$(0),JO,1)<MID$(GR$(0),IO,1) THEN NEXT JO:GOTO 3550
3470 : REM: NOW FIND SENDER-NODE'S TOPOLOGICAL ORDER,
3480 FOR J=1 TO JL
3490 IF MID$(L$(0),JO-1,1)<MID$(GR$(0),JJ,1) THEN NEXT JJ
3500 : REM: INSERT A LINK MARK IN THE GRID LINE,
3510 ODZ(JJ)=ASC(MK$)
3520 : REM: AND CLOSE LOOPS.

```


Tree Analysis (Continued)

```

3530 JJ=LL:NEXT JJ,JO
3540 : REM: WHEN LINE IS FINISHED, PRINT IT.
3550 FOR JO=1 TO IO:PRINT#1,CHR$(ODZ(JO));NEXT JO
3560 : REM: CONTINUE THROUGH ALL LINES.
3570 NEXT IO
3580 : REM: IF DOING HARD COPY, PRINT OUT NAMES.
3590 IF DV=4 THEN PRINT#1:FOR IO=1 TO IL:PRINT#1,NM$(FNG(IO));NEXT IO
3600 CLOSE 1
3610 PRINT"<S>";PRINT" <R>TYPE RETURN";PRINT" <R> TO CONTINUE
3620 : REM: WAIT FOR USER RESPONSE.
3630 GET A$:IF A$="" THEN 3640
3640 : REM: CLEAR SCREEN AND RETURN TO ELICITATION.
3650 PRINT "<S>";RETURN
3660 : REM: SUBROUTINE TO REVISE A NODE.
3670 : REM: REMOVE : FROM FRONT OF WORD, AND FIND IN NM$ LIST.
3680 WD$=RIGHT$(WD$,LEN(WD$)-1);K=IL:GOSUB 1340:GOSUB 3060
3690 : REM: ABORT IF WORD WAS NOT IN NM$ LIST.
3700 IF K<IL THEN NM$(IL)="" : IL=IL-1:PRINT "<R>NOT FOUND":RETURN
3710 PRINT "<S>FOUND";PRINT "<R>NM$(ZZ)
3720 : REM: PRINT MENU.
3730 PRINT "YOU CAN <R><R>CHANGE THE NAME,
3740 PRINT " OR <R><R>DELETE THE WORD ENTIRELY,
3750 PRINT " OR <R><R>EXIT.
3760 PRINT "<Q>ENTER <R><R>, <R><R>, OR <R><R>".
3770 GET A$:IF A$="" OR (A$="C" AND A$="E") THEN 3770
3780 : REM: GO TO SUBROUTINE FOR DELETION.
3790 IF A$="E" THEN RETURN
3800 IF A$="D" THEN GOSUB 3890:RETURN
3810 : REM: DO NAME CHANGE.
3820 PRINT "<Q>NEW NAME FOR <R><R>RIGHT$(NM$(ZZ),LEN(NM$(ZZ))-3)
3830 : REM: ADD ARTICLE AND REPLACE.
3840 GOSUB 1340:NM$(ZZ)=WD$
3850 RETURN
3860 : REM: SUBROUTINE TO DELETE A NODE.
3870 : REM: PUT NODE TO BE REMOVED ON 'STACK'.
3880 ODZ(0)=1:ODZ(1)=ZZ
3890 : REM: GET LAST NODE, X, ON STACK.
3900 X=ODZ(0)
3910 X=ODZ(ODZ(0))
3920 : REM: DEFINE ITS DOWN-GRAPH,
3930 BR$(0)="" : LQ=USR(X+DOWN)
3940 : REM: WITHOUT THE X NODE.
3950 GR$(0)=RIGHT$(GR$(0),LG-1)
3960 : REM: SAVE GR$.
3970 GG$=LEFT$(GR$(0),LEN(GR$(0)))
3972 : REM: WHEN X HAS SUBORDINATES,
3974 IF LEN(GR$(0))=0 THEN 4270
3980 : REM: CONSIDER THE FIRST OF THEM.
3990 J=FNG(1)
4000 : REM: ASK IF NODE J IS TO BE KEPT.
4010 PRINT:PRINT "RETAIN <R>RIGHT$(NM$(J),LEN(NM$(J))-4)"<R>? ";GOSUB 830
4020 : REM: IF NOT, DELETE J RECURSIVELY, THEN START OVER ON X.
4030 IF A$="N" THEN ODZ(0)=ODZ(0)+1:ODZ(0)=J:GOSUB 3910:GOTO 3910
4040 BR$(0)="" : LQ=USR(J+UP+256)
4050 BR$(0)="" : LQ=USR(J+UP+256)
4060 : REM: DELETE EDGE FROM X TO J.
4060 K=LEN(L$(0))
4070 FOR I=1 TO K STEP 2
4080 B$=MID$(L$(0),I,2)
4090 IF A$<>B$ THEN NEXT I:GOTO 4210
4100 A$="";IF I>1 THEN A$=LEFT$(L$(0),I-1)
4110 B$="";IF I<K-1 THEN B$=RIGHT$(L$(0),K-I-1)
4120 L$(0)=A$+B$
4130 I=I+1:NEXT I
4140 : REM: AND USE UP-LINKS OF X TO DEFINE UP-LINKS OF J.

```

Tree Analysis (Continued)

```

2840 PRINT W$RIGHT$(NM$(IP),LEN(NM$(IP))-1)";
2850 : REM: PRINT IMMEDIATE SUBORDINATES OF NODE IP.
2860 FOR I=1 TO LEN(L$(0)) STEP 2
2870 IF CHR$(IP+64)=MID$(L$(0),I,1) THEN PRINT " NM$(FNG(I+1))
2880 NEXT I
2890 : REM: PRINT ELICITATION QUESTION AND GET ANSWER.
2900 PRINT:PRINT " WHAT IS A MORE GENERAL TERM FOR
2910 PRINT " SOME OF THESE (<R><R>):";GOSUB 670
2920 : REM: PERMIT SKIP.
2930 IF WD$="" THEN RETURN
2940 : REM: ACCEPT PRINT-GRID COMMAND.
2950 IF LEFT$(WD$,1)="" THEN GOSUB 3160:GOTO 2750
2960 : REM: ADD AN ARTICLE TO NEW WORD, ENTER IN NM$, AND LINK IT INTO TREE.
2970 GOSUB 1340:H=L:GOSUB 3060:IF H=L THEN PRINT"<R>ALREADY EXISTS":GOTO 2750
2980 GOSUB 1880
2990 : REM: SHOW GRID.
3000 GOSUB 3160
3010 RETURN
3020 : REM: SUBROUTINE TO ADD A WORD TO THE LIST OF WORDS, NM$.
3030 : REM: IF WORD ALREADY IS IN NM$.
3040 : REM: THEN CLOSE LOOP AND RETURN.
3050 FOR JJ=2 TO IL
3070 IF WD$=LEFT$(NM$(JJ),LEN(WD$)) THEN ZZ=JJ:JJ=IL:NEXT JJ:RETURN
3080 NEXT JJ
3090 : REM: OTHERWISE ADD THE WORD TO NM$.
3100 ZZ=JJ:IL=JJ:NM$(JJ)=WD$
3110 RETURN
3120 :
3130 : REM: SUBROUTINE TO DISPLAY RESULTS ON GRID.
3140 : REM: ANSWER '##' GIVES HARD COPY.
3150 : REM: SET DEVICE # FOR SCREEN OR PRINTER.
3160 DV=3:IF MID$(WD$,2,1)="" THEN DV=4
3170 : REM: SCREEN OUTPUT LIMITED TO 22 NODES.
3180 IF DV=3 AND MS>0 AND IL>22 THEN RETURN
3190 : REM: IF ON THE 23RD NODE, PRINT ONE-TIME MESSAGE.
3200 MS$="" : DO MANY ENTRIES TO DISPLAY GRID ON
3210 IF MS=0 AND IL>23 THEN PRINT MS$+MS=1:RETURN
3220 : REM: SET GRAPHIC CHARACTERS FOR SCREEN OR PRINTER.
3230 LN$=CHR$(219):MK$=CHR$(209):DG$=CHR$(95)
3240 IF DV=4 THEN LN$="" : MK$="" : DG$=""
3250 : REM: OPEN PRINTING FILE.
3260 OPEN 1,DV
3270 : REM: CLEAR SCREEN AND PRINT HEADER.
3280 PRINT"<S>";
3290 PRINT#1,MID$(NM$(1),2):PRINT#1
3300 : REM: SEARCH DOWN TO GET ORDERED LIST OF NODES (GR$) WITHOUT REPEATS.
3310 BR$(0)="" : LQ=USR(1+DOWN)
3320 : REM: FOR EACH WORD IN TOPOLOGICAL ORDER,
3330 FOR IO=2 TO IL
3340 : REM: REMOVE ARTICLE,
3350 W$=MID$(NM$(FNG(IO)),5)
3360 : REM: AND TRUNCATE IF TOO LONG.
3370 W$=MID$(W$,1,13)
3380 : REM: THEN PRINT ON NEXT ROW, FOLLOWED BY SPACES TO EDGE OF GRID.
3390 PRINT#1:PRINT#1,LEFT$(W$+SP$,15)
3400 : REM: GENERATE GRID CHARACTERS.
3410 FOR JO=1 TO IO-1:ODZ(JO)=ASC(LN$):NEXT JO:ODZ(IO)=ASC(DG$)
3420 : REM: SET SPACING INDEX.
3430 JL=1
3440 : REM: GO THROUGH RECEIVER NODES IN EDGE-LIST TO FIND THIS NODE.
3450 FOR JO=2 TO LEN(L$(0)) STEP 2
3460 IF MID$(L$(0),JO,1)<MID$(GR$(0),JO,1) THEN NEXT JO:GOTO 3550
3470 : REM: NOW FIND SENDER-NODE'S TOPOLOGICAL ORDER,
3480 FOR J=1 TO JL
3490 IF MID$(L$(0),JO-1,1)<MID$(GR$(0),J,1) THEN NEXT JJ
3500 : REM: INSERT A LINK MARK IN THE GRID LINE,
3510 ODZ(JJ)=ASC(MK$)
3520 : REM: AND CLOSE LOOPS.

```

Tree Analysis (continued)

```

4150 K=K-2
4160 FOR I=2 TO K STEP 2
4170 IF MID$(L$(0),I,1)<>CHR$(X+64) THEN NEXT I:GOTO 4210
4180 : REM: (DON'T LINK IF PATH ALREADY EXISTS.)
4190 KK=0
4173 FOR II=1 TO LL
4174 IF MID$(L$(0),I-1,1)=MID$(GR$(0),II,1) THEN KK=KK+1
4175 NEXT II
4176 IF KK>1 THEN 4190
4180 L$(0)=L$(0)+MID$(L$(0),I-1,1)+CHR$(J+64)
4190 NEXT I
4200 : REM: TAKE DOWN-GRAPH OF J OUT OF GR$.
4210 BR$(0)=LEFT$(GG$,LEN(BR$(0)))
4220 LD=USR(J+DOWN)
4230 GR$(0)=LEFT$(BR$(0),LEN(BR$(0)))
4240 : REM: AND CONTINUE WITH OTHER SUBORDINATES OF X.
4250 GOTO 3970
4260 I: REM: WHEN X HAS NO SUBORDINATES, REMOVE ITS UP LINKS FROM L$.
4270 K=LEN(L$(0))
4280 FOR I=2 TO K STEP 2
4290 IF MID$(L$(0),I,1)<>CHR$(X+64) THEN NEXT I:GOTO 4350
4300 AS="": IF I>2 THEN AS=LEFT$(L$(0),I-2)
4310 BS="": IF I<K THEN BS=RIGHT$(L$(0),K-I)
4320 L$(0)=AS+BS
4330 I=I:GOTO 4270
4340 : REM: THEN REMOVE THE WORD FROM THE NM$ LIST.
4350 FOR I=X TO 64
4360 NM$(I)=NM$(I+1)
4370 NEXT I:NM$(65)="
4380 IL=IL-1
4390 : REM: LOOK THROUGH L$, AND
4400 FOR I=1 TO K
4410 : REM: IF AN INDEX VALUE IS ABOVE X,
4420 IF FN(I)<X THEN NEXT I: GOTO 4490
4430 : REM: DECREMENT IT TO ADJUST FOR THE DELETED NODE.
4440 AS="": IF I>1 THEN AS=LEFT$(L$(0),I-1)
4450 BS="": IF I<K THEN BS=RIGHT$(L$(0),K-I)
4460 L$(0)=AS+CHR$(ASC(MID$(L$(0),I,1))-1)+BS
4470 NEXT I
4480 : REM: SIMILARLY, ADJUST INDEX VALUES IN THE 'STACK'.
4490 FOR I=1 TO ODZ(0)
4500 IF ODZ(I)>X THEN ODZ(I)=ODZ(I)-1
4510 NEXT I
4520 : REM: REMOVE THE X NODE FROM THE 'STACK'.
4530 ODZ(0)=ODZ(0)-1
4540 : REM: WHEREUPON REMOVAL OF NODE X IS FINISHED.
4550 : REM: RESET SCANNING INDEX.
4560 IP=2
4570 RETURN
4580 : REM: SUBROUTINE TO SAVE DATA,
4590 : REM: CALLED BY ENTERING .LABEL OR \LABEL AS A WORD.
4610 : REM: MAKE THE FILE NAME FROM NM$(1) AND THE LABEL.
4620 K=9-TA: IF TA=3 THEN K=4
4630 F$=MID$(WD$,2,7)+". "+MID$(NM$(1),K,8)
4640 : REM: SET UP TAPE SAVE,
4650 DV=1:CH=1
4660 : REM: BUT CHANGE TO DISK SAVE WHEN SLASH PRECEDES LABEL.
4670 IF LEFT$(WD$,1)="/" THEN DV=8:CH=S:F$="0:"+F$+"S,W"
4680 : REM: OPEN THE WRITE FILE.
4690 OPEN 1,DV,CH,F$
4700 : REM: SAVE THE TYPE OF ANALYSIS (FOLLOWED BY 'RETURN'),
4710 PRINT#1,TA:CHR$(13)
4720 : REM: THE NUMBER OF NODES,
4730 PRINT#1,IL:CHR$(13)
4740 : REM: THE NODE NAMES,
4750 FOR I=1 TO IL:PRINT#1,CHR$(34);NM$(I);CHR$(13);:NEXT I
4760 : REM: THE NUMBER OF EDGES,

```

```

4770 PRINT#1,LEN(L$(0))/2:CHR$(13)
4780 : REM: AND THE LIST OF EDGES.
4790 FOR I=1 TO LEN(L$(0)) STEP 2:PRINT#1,MID$(L$(0),I,2):CHR$(13);:NEXT I
4800 : REM: FINISH.
4810 CLOSE : RETURN
4820 :
4830 : REM: SUBROUTINE TO READ DATA,
4840 : REM: CALLED BY ENTERING T OR D AT FIRST QUESTION.
4850 : REM: GET THE FILE NAME.
4860 INPUT "<R>FILE NAME<R>:"; F$
4870 : REM: SET UP TAPE READ,
4880 DV=1:CH=0
4890 : REM: BUT CHANGE TO DISK READ WHEN AS=D.
4900 IF WD$="D" THEN DV=8:CH=S:F$="0:"+F$+"S,R"
4910 : REM: OPEN THE READ FILE.
4920 OPEN 1,DV,CH,F$
4930 : REM: GET AND DISPLAY THE TYPE OF ANALYSIS.
4940 INPUT#1,TA:PRINT "<Q><Q>TYPE OF ANALYSIS =";TA
4950 : REM: THE NUMBER OF NODES,
4960 INPUT#1,IL:PRINT IL;"NODES ARE:"
4970 : REM: THE NODE NAMES,
4980 FOR I=1 TO IL:INPUT#1,NM$(I):PRINT NM$(I);:NEXT I:PRINT
4990 : REM: THE NUMBER OF EDGES,
5000 INPUT#1,LL:PRINT LL;"EDGES:"
5010 : REM: AND THE LIST OF EDGES.
5020 FOR I=1 TO LL:INPUT#1,AS:PRINT AS;:L$(0)=L$(0)+AS:NEXT I:PRINT
5030 : REM: FINISH.
5040 CLOSE 1:ON TA GOSUB 980,1130,1280
5050 RETURN
5060 :
5070 : REM: SUBROUTINE TO DISPLAY COMMONALITIES.
5080 : REM: GET GENERAL DOWN-GRAPH.
5090 BR$(0)="":LL=USR(1+DOWN)
5100 : REM: SHOW NODES IN TYPOLOGICAL ORDER WITH REFERENCE NUMBERS.
5110 FOR I=2 TO IL:PRINT "<R>"FN(I)"<R>"MID$(NM$(FN(I)),5,13);:NEXT I
5120 : REM: PROMPT.
5130 GG$="":K=1:ODZ(0)=0:PRINT:PRINT"COMMONALITIES FOR WHICH ENTRIES"
5140 INPUT "(ENTER 0 TO END)"; ODZ(K):IF ODZ(K)=0 THEN S210
5150 ODZ(0)=ODZ(0)+1
5160 : REM: CONCATENATE THE UP-GRAPHS.
5170 LL=USR(ODZ(K)+UP)
5180 GG$=GG$+BR$(0)
5190 K=K+1:GOTO 5140
5200 : REM: CREATE A DUMMY SUBORDINATE OF FOCUS NODES.
5210 IL=IL+1:NM$(IL)=""
5220 FOR I=1 TO ODZ(0):L$(0)=L$(0)+CHR$(ODZ(I)+64)+CHR$(IL+64):NEXT I
5230 : REM: ELIMINATE IRRELEVANT NODES FROM GG$ BY TAKING UP-GRAPH FROM GG.
5240 FOR I=1 TO ODZ(0)-1
5250 BR$(0)=LEFT$(GG$+" ",LEN(GG$)):LL=USR(IL+UP)
5260 GG$=LEFT$(BR$(0),LEN(BR$(0)))
5270 : REM: REPEAT IF MORE THAN TWO FOCUS NODES.
5280 NEXT I
5290 : REM: LOWEST NODE OF GG$ IS A COMMONALITY.
5300 K=ASC(RIGHT$(GG$,1))-64
5310 PRINT:PRINT"COMMONALITY="NM$(K)
5320 : REM: ELIMINATE ITS UP-GRAPH.
5330 BR$(0)=LEFT$(BR$(0),LEN(GG$)):LL=USR(K+UP)
5340 GG$=LEFT$(BR$(0),LEN(BR$(0)))
5350 : REM: FIND SUBORDINATES LEADING TO EACH FOCUS NODE.
5360 FOR J=1 TO ODZ(0)
5370 : REM: PRINT RELEVANT FOCAL NODE AND ITS SUBCOMMONALITIES.
5380 PRINT "FOR"NM$(ODZ(J))
5390 : REM: GET UP-GRAPH FROM FOCUS NODE.
5400 BR$(0)="" : LL=USR(ODZ(J)+UP):AS=LEFT$(BR$(0),LEN(BR$(0)))
5410 : REM: GET DOWN-GRAPH FROM COMMON NODE.
5420 LL=USR(K+DOWN):BS=LEFT$(BR$(0),LEN(BR$(0)))
5430 : REM: SUBTRACT DOWN-GRAPH FROM SUBORDINATE OF COMMON NODE:

```

Tree Analysis (continued)

```

5440 IF LEN(B$)=1 THEN NEXT J:GOTO 5550
5450 REM: FROM A$:
5460 BR$(0)=LEFT$(A$+" ",LEN(A$)):LG=ASC(MID$(B$,2,1))-64:LL=USR(LG+DOWN)
5470 I=ASC(RIGHT$(BR$(0),1))-64
5480 REM: THEN FROM B$:
5490 BR$(0)=LEFT$(B$+" ",LEN(B$)):LL=USR(LG+DOWN)
5500 BR$(0)=LEFT$(BR$(0),LEN(BR$(0)))
5510 REM: FOUND A LINK IF FOCAL NODE WAS DELETED FROM A$.
5520 IF I<>ODZ(J) THEN PRINT NM$(LG)
5530 GOTO 5440
5540 REM: IF THERE ARE MORE COMMON NODES, THEN CONSIDER THEM.
5550 IF LEN(GG$)>0 THEN 5300
5560 REM: OTHERWISE ELIMINATE THE DUMMY NODE
5570 MD$=NM$(I):IZZ=IL:GOSUB 3970
5580 REM: THEN WAIT FOR USER RESPONSE AND RETURN.
5590 PRINT "<R>PRESS RETURN TO CONTINUE."
5600 GOTO 3640
5610 REM:
5620 "<S>": CURSOR HOME
5630 "<S>": CLEAR SCREEN
5640 "<Q>": CURSOR DOWN
5650 "<Q>": CURSOR UP
5660 "<I>": CURSOR RIGHT
5670 "<I>": CURSOR LEFT
5680 "<R>": REVERSE ON
5690 "<R>": REVERSE OFF
5700 "<I>": SHADED SQUARE
READY.
    
```

Listing 3: Trees

```

0010 REM:
0020 REM:
0030 REM: TREES
0040 REM:
0050 REM:
0060 REM:
0090 START
0095 REM:
0100 REM:
0110 REM:
0120 REM:
0130 REM:
0140 INHELP
0150 FLPTNT
0160 OUT
0170 REM:
0180 REM:
0190 PTR
0200 LIM
0210 RAM
0220 NAME
0230 L&POINTER
0240 FAC
0250 REM:
0270 GRAPH
0280 TLIST
0290 COUNT
0300 REM:
0310 MAX
0320 REM:
0350 REM:
0360 REM:
0370 INVOKE
0380 REM:
3DA8- A9 B9
3DA9- 85 01
    
```

Trees (Continued)

```

3DAC- A9 3D
3DAE- 85 02
3DB0- A9 A7
3DB2- 85 34
3DB4- A9 3D
3DB6- 85 35
3DB8- 60
0470
0480
0490
0500
0510
0520
0530
0540
0550
0560
0570
0580
0590
0600
0610
0620
0630
0640
0650
0660
0670
0680
0690
0700
0710
0720
0730
0740
0750
0760
0770
0780
0790
0800
0810
0820
0830
0840
0850
0860
0870
0880
0890
0900
0910
0920
0930
0940
0950
0960
0970
0980
0990
1000
1010
1020
1030
1040
3DC9- A9 4E
3DCB- 85 42
3DCD- A9 4D
3DCF- 09 80
3DD1- 85 43
3DD3- 20 1D 3E
3DD6- A0 00
3DD8- B1 54
3DDA- C9 48
3DDC- 90 05
3DDF- A2 46
3DE0- 4C 69 C3
3DE3- A4
3DE4- A9 FF
3DE6- 8D E0 3E
3DE9- A0 FE
3DEB- C8
3DEC- C8
3DED- C8
3DEE- B1 54
3DF0- F0 03
3DF2- EE E0 3E
3DF5- CA
3DF6- D0 F3
3DF8- A9 4C
3DFA- 85 42
3DFC- A9 80
3DFE- 85 43
3E00- 20 1D 3E
3E03- 20 9A D0
3E06- 45 62
3E08- 18
3E09- 69 40
3E0B- 8D 01 3F
3E0E- A9 01
3E10- 26 61
3E12- 6D 12 3F
3E15- A9 02
3E17- 25 61
LDA #H,TREE
STA #2
LDA #L,INVOKE-1
LDA #RAM
LDA #H,INVOKE-1
STA #RAM+1
RTS
*****
EXECUTIVE PROGRAM
;SET PARAMETERS FOR PROCEDURES
;GET TOPOLOGICAL ORDERING OF NODES
;GET LIST OF REACHABLE NODES -- GR$
;PUT GR$ IN TOPOLOGICAL ORDER
;REMOVE GR$ ENTRIES FROM BR$
*****
PRELIMINARY PROCEDURES IN TREE.
;COUNT NUMBER OF NODES IN TREE.
;FIND THE NM$ ARRAY.
LDA #N'
STA #NAME
LDA #M'
ORA #128
STA #NAME+1
JSR FIND
LDY #0
LDA (L&POINTER),Y
CMP #MAX+2
BC #MAX+2
LDX #46
;IF DIM>MAX,
;ABORT WITH OVERFLOW ERROR.
;SET X= DIM OF NM$ ARRAY.
;INITIALIZE COUNTER TO GIVE
;NUMBER OF NODES -1.
;TO START AT 1ST STRING IN NM$ ARRAY,
;MOVE UP 3 BYTES (NOT FOR OS 4.0)
;AND CHECK STRING LENGTH.
LDA (L&POINTER),Y
BEQ DIM
INC JL+1
;INCREMENT THE COUNT.
DEX
BNE DONE
;CONTINUE IF MORE IN NM$.
;FIND THE LIST OF EDGES--L$ ARRAY
LDA #L'
STA #NAME
LDA #128
STA #NAME+1
JSR FIND
;GETS L$ POINTER AND LENGTH.
;GET PARAMETERS PASSED BY USR.
;CONVERT # IN FAC TO 2 BYTES.
;GET START POINT FOR TREE SEARCH AND
;CONVERT TO CHARACTER.
ADC #64
LDA #1
;ALLOW REPEATS IN TRACING GRAPH?
AND #FACC+3
;(1=YES; 0=NO.)
STA RE+1
;TRACE UP OR DOWN?
LDA #2
AND #FACC+3
;(2=DOWN; 0=UP.)
    
```

Trees (Continued)

```
3E19- 8D F2 3E 1050 STA DI+1
3E1C- 60 1060 RTS
1070
1080
1090
1100 FIND
3E1D- 45 2C LDA #PTR
3E1F- 48 PHA
3E20- 45 2D LDA #PTR+1
3E22- 48 PHA
3E23- 40 00 LDA #0
3E25- B1 2C LDA (PTR),Y
3E27- C5 42 CMP #NAME
3E29- D0 08 BNE AGAIN2
3E2B- C8 1180 INY
3E2C- B1 2C LDA (PTR),Y
3E2E- C5 43 CMP #NAME+1
3E30- F0 23 BCP POINT
3E32- C8 1220 BNE AGAIN2
3E33- C8 1230 AGAIN2
3E34- C8 1240 BCP POINT
3E35- C8 1250 INY
3E36- B1 2C LDA (PTR),Y
3E38- 48 PHA
3E39- 88 1270 INY
3E3A- B1 2C LDA (PTR),Y
3E3C- 18 1290 CLC
3E3D- 65 2C ADC #PTR
3E3F- 85 2C STA #PTR
3E41- 68 1330 PLA
3E42- 65 2D ADC #PTR+1
3E44- 85 2D STA #PTR+1
3E46- C5 2F CMP #LJM+1
3E48- D0 D9 BNE CHECK
3E4A- A5 2E LDA #LJM
3E4C- C5 2C CMP #PTR
3E4E- D0 D3 BNE CHECK
3E50- A2 81 LDY #81
3E52- 4C 69 C3 JMP OUT
3E55- 40 07 LDY #7
3E57- B1 2C LDA (PTR),Y
3E59- 8D 8A 3E STA LL+1
3E5C- C8 1460 INY
3E5D- B1 2C LDA (PTR),Y
3E5F- 85 42 STA #NAME
3E61- C8 1490 INY
3E62- B1 2C LDA (PTR),Y
3E64- 85 43 STA #NAME+1
3E66- 18 1520 CLC
3E67- A9 06 LDA #6
3E69- 65 2C ADC #PTR
3E6B- 85 54 STA #L\A\POINT
3E6D- A9 00 LDA #0
3E6F- 65 2D STA #PTR+1
3E71- 85 55 PLA
3E73- 68 1590 STA #L\A\POINT+1
3E74- 85 2D STA #PTR+1
3E76- 68 1610 PLA
3E77- 85 2C STA #PTR
3E79- 60 1620 RTS
1630
1640
1650
1660
1670 TOPOL
3E7A- A2 45 LDA #MAX-1
3E7C- A9 00 LDA #0
3E7E- 9D 6D 03 STA COUNT,X
3E81- A9 80 LDA #128
3E83- 9D B3 03 STA TLIST,X
3E86- CA 1720 DEX
```

Trees (Continued)

```
3E87- 10 F3 1730 BPL INITIALIZ
1740 ;
1750 LL ;COUNT INCOMING EDGES FOR EACH NODE.
1760 LDI #0 ;LOAD L$ LENGTH-1.
1770 INVEDGES DEY ;
1780 LDA (NAME),Y ;GET A RECEIVER IN L$ (2ND IN PAIR).
1790 SEC ;
1800 SBC #65 ;CONVERT CHARACTER TO INDEX #.
1810 TAX
1820 3E92- FE 6D 03 INC COUNT,X
1830 3E95- DD 03 BNE INVEDGES
1840 LDA #A' ;CONTINUE WITH NEXT EDGE PAIR IN L$.
1850 STA TLIST ;START TLIST WITH TREE ORIGIN.
1860 LDY #0
1870 LDY LL+1 ;SET TLIST INDEX.
1880 LDY #0 ;GET CURRENT NODE FROM TLIST.
1890 STA #A' ;SET TLIST INDEX.
1900 LDA TLIST,X ;GET CURRENT NODE FROM TLIST.
1910 BPL NOLOOP
1920 LDA #8D5 ;(LOOP! ABOBUT WITH COMPLEX ERROR.)
1930 JMP OUT
1940 NOVALOOP LDI LL+1 ;LOOK FOR OUTGOING EDGE.
1950 DEY
1960 DEY
1970 SEARCH\OM1 CMP (NAME),Y
1980 BNE SEARCH\ON2 ;
1990 ; PHA ;
2000 TXA ;
2010 PHA ;
2020 TXA ;
2030 INY ;
2040 LDA (NAME),Y ;DETERMINE RECEIVING NODE,
2050 SEC ;
2060 SBC #65 ;CONVERT CHARACTER TO INDEX #, AND
2070 TAX
2080 DEC COUNT,X ;DECREMENT COUNT FOR THAT NODE.
2090 BNE RESTORE ;IF RECEIVER'S COUNT IS 0,
2100 CLC ;
2110 ADC #65 ;REMAKE INDEX # INTO CHARACTER,
2120 INC #A+1 ;INCREMENT TLIST INDEX,
2130 STA TLIST,X ;AND STORE THE NODE ON TLIST.
2140 PLA ;
2150 RESTORE TXA ;
2160 TAX ;
2170 PLA ;
2180 DEY ;
2190 SEARCH\ON2 CFI #0 ;
2200 BEQ NEXT ;
2210 DEY ;
2220 BEQ NEXT ;
2230 DEY ; ;CONTINUE THROUGH L$.
2240 JMP SEARCH\ON1 ;WHEN ALL EDGES HAVE BEEN CHECKED,
2250 NEXT INX ;REPEAT FOR NEXT ENTRY IN TLIST.
2260 IL CFI #0 ;STOPPING WHEN LAST NODE IS REACHED.
2270 BNE NXNODE ;
2280 RTS ;
2290 ;
2300 ;
2310 ;
2320 ;
2330 TRACE LDA #255 ;
2340 PHA ;
2350 PHA ;
2360 PHA ;
2370 PHA ;
2380 PHA ;
2390 PHA ;
2400 PHA ;
2410 PHA ;
2420 PHA ;
2430 PHA ;
2440 PHA ;
2450 PHA ;
2460 PHA ;
2470 PHA ;
2480 PHA ;
2490 PHA ;
2500 PHA ;
2510 PHA ;
2520 PHA ;
2530 PHA ;
2540 PHA ;
2550 PHA ;
2560 PHA ;
2570 PHA ;
2580 PHA ;
2590 PHA ;
2600 PHA ;
2610 PHA ;
2620 PHA ;
2630 PHA ;
2640 PHA ;
2650 PHA ;
2660 PHA ;
2670 PHA ;
2680 PHA ;
2690 PHA ;
2700 PHA ;
2710 PHA ;
2720 PHA ;
2730 PHA ;
2740 PHA ;
2750 PHA ;
2760 PHA ;
2770 PHA ;
2780 PHA ;
2790 PHA ;
2800 PHA ;
2810 PHA ;
2820 PHA ;
2830 PHA ;
2840 PHA ;
2850 PHA ;
2860 PHA ;
2870 PHA ;
2880 PHA ;
2890 PHA ;
2900 PHA ;
2910 PHA ;
2920 PHA ;
2930 PHA ;
2940 PHA ;
2950 PHA ;
2960 PHA ;
2970 PHA ;
2980 PHA ;
2990 PHA ;
3000 PHA ;
3010 PHA ;
3020 PHA ;
3030 PHA ;
3040 PHA ;
3050 PHA ;
3060 PHA ;
3070 PHA ;
3080 PHA ;
3090 PHA ;
3100 PHA ;
3110 PHA ;
3120 PHA ;
3130 PHA ;
3140 PHA ;
3150 PHA ;
3160 PHA ;
3170 PHA ;
3180 PHA ;
3190 PHA ;
3200 PHA ;
3210 PHA ;
3220 PHA ;
3230 PHA ;
3240 PHA ;
3250 PHA ;
3260 PHA ;
3270 PHA ;
3280 PHA ;
3290 PHA ;
3300 PHA ;
3310 PHA ;
3320 PHA ;
3330 PHA ;
3340 PHA ;
3350 PHA ;
3360 PHA ;
3370 PHA ;
3380 PHA ;
3390 PHA ;
3400 PHA ;
3410 PHA ;
3420 PHA ;
3430 PHA ;
3440 PHA ;
3450 PHA ;
3460 PHA ;
3470 PHA ;
3480 PHA ;
3490 PHA ;
3500 PHA ;
3510 PHA ;
3520 PHA ;
3530 PHA ;
3540 PHA ;
3550 PHA ;
3560 PHA ;
3570 PHA ;
3580 PHA ;
3590 PHA ;
3600 PHA ;
3610 PHA ;
3620 PHA ;
3630 PHA ;
3640 PHA ;
3650 PHA ;
3660 PHA ;
3670 PHA ;
3680 PHA ;
3690 PHA ;
3700 PHA ;
3710 PHA ;
3720 PHA ;
3730 PHA ;
3740 PHA ;
3750 PHA ;
3760 PHA ;
3770 PHA ;
3780 PHA ;
3790 PHA ;
3800 PHA ;
3810 PHA ;
3820 PHA ;
3830 PHA ;
3840 PHA ;
3850 PHA ;
3860 PHA ;
3870 PHA ;
3880 PHA ;
3890 PHA ;
3900 PHA ;
3910 PHA ;
3920 PHA ;
3930 PHA ;
3940 PHA ;
3950 PHA ;
3960 PHA ;
3970 PHA ;
3980 PHA ;
3990 PHA ;
4000 PHA ;
```

Trees (continued)

```

3F71- A9 00 2400 DI LDA #0 ;BUT IF SEARCHING DOWN,
3F72- 06 06 2410 BEQ STOREY ;INSERT AN INY COMMAND,
3F73- A9 08 2420 LDA #200 STA CMD
3F74- 80 0E 2430 DEY ; AND ADJUST Y.
3F75- 88 2440 ; AND ADJUST Y.
3F76- 8C 06 2450 STOREY STY *-1
3F77- A2 00 2460 LDA #0 ;START GRAPH WITH NODE
3F78- 90 0A 2470 SP ;GIVEN AS STARTING POINT.
3F79- 9D 7A 2480 BUILDGR LDA GRAPH,X
3F80- A0 00 2490 I LDI #0 ;STORE THE CURRENT NODE IN GR$.
3F81- D1 42 2500 CMP (NAME),Y ;INITIALIZE I.
3F82- D0 19 2510 SEARCHV LDA #0 ;IF CURRENT GR$ NODE = NEXT L$ NODE
3F83- 48 2520 BNE NEXTVEDE ;PUT NODE
3F84- 98 2530 TIA ;AND ITS L$ POSITION
3F85- 98 2540 PHA ;ON STACK.
3F86- 48 2550 DEY ;(DEY FOR UP SEARCH, INY FOR DOWN.)
3F87- 88 2560 CMD DEY ;GET OTHER NODE IN EDGE.
3F88- B1 42 2570 LDA (NAME),Y ;IF REPEATS ARE NOT WANTED
3F89- A0 00 2580 LDI #0 ;IF REPEATS ARE NOT WANTED
3F90- D0 0C 2590 RE BNE PUTVIN ;(TRANSFER X TO Y.)
3F91- 48 2600 PHA ;
3F92- 8A 2610 TXA ;
3F93- 88 2620 TAY ;
3F94- 68 2630 PLA ;
3F95- D9 7A 2640 CMP GRAPH,Y ;AND NODE ALREADY IS IN GR$
3F96- F0 11 2650 BEQ BACKUP ;THEN CONTINUE SEARCH.
3F97- 88 2660 DEY ;
3F98- 10 F8 2670 BPL UNIQUE ;OTHERWISE MAKE IT THE
3F99- 21 E8 2680 PUTVIN ;NEW CURRENT NODE IN GR$.
3F100- 10 F8 2690 PUTVIN ;
3F101- E8 2700 TXA ;
3F102- 10 DE 2710 BPL BUILDGR ;
3F103- C0 00 2720 NEXTVEDE ;
3F104- F0 07 2730 BEQ BACKUP ;
3F105- 88 2740 DEY ;
3F106- F0 04 2750 BEQ BACKUP ;
3F107- 88 2760 DEY ;
3F108- 4C 07 2770 JMP SEARCHV ;CONTINUE SEARCHING L$, OR
3F109- 68 2780 BACKUP PLA ;GET LAST ENTRIES ON STACK
3F110- A8 2790 TAY ;
3F111- 68 2800 PLA ;
3F112- 10 F0 2810 BPL NEXTVEDE ;AND CONTINUE THAT SEARCH.
3F113- 10 F0 2820 ;SEARCH IS DONE IF STACK IS EMPTY.
3F114- 88 2830 INX ;STORE GR$ LENGTH FOR RECALL.
3F115- 8E 4D 2840 STX GL+1 ;
3F116- 8A 2850 TXA ;
3F117- 88 2860 TAY ;
3F118- A9 00 2870 LDA #0 ;RETURN GR$ LENGTH IN USR.
3F119- 20 D2 2880 JSR INTPL ;
3F120- A9 47 2890 LDA #G' ;GET POINTER TO GR$ HEADER.
3F121- 85 42 2900 STA #NAME ;
3F122- A9 52 2910 LDA #R' ;
3F123- 06 80 2920 ORA #I28 ;
3F124- 8E 43 2930 STA #NAME+1 ;
3F125- 06 80 2940 JSR FIND ;
3F126- E0 1D 2950 LDA #0 ;STORE LENGTH OF GR$
3F127- A9 01 2960 LDI #1 ;
3F128- A0 01 2970 STA (L\POINTER),Y ;
3F129- 91 54 2980 INY ;AND POSITION OF GR$.
3F130- C8 2990 ;
3F131- A9 7A 3000 LDA #,GRAPH ;
3F132- 91 54 3010 STA (L\POINTER),Y ;
3F133- C8 3020 INY ;
3F134- A9 02 3030 LDA #,H.GRAPH ;
3F135- 91 54 3040 STA (L\POINTER),Y ;
3F136- C8 3050 ;
3F137- 60 3060 ; PUT GR$ IN TOPOLOGICAL ORDER.
3080 ;

```

Trees (continued)

```

3F5D- A0 00 3090 REORDER LDA #0 ;START WITH 1ST ELEMENT IN TLIST
3F5E- 6C 66 3F 3100 STY INGR+1 ;AND GR$.
3F5F- B9 B3 03 3110 TORRER LDA TLIST,Y ;GET NODE FROM TLIST.
3F60- A2 00 3120 INGR LDX #0 ;COMPARE TO UNORDERED NODE
3F61- DD 7A 02 3130 FINDANODE CMP GRAPH,X ;IN GR$.
3F62- DD 7A 02 3140 PHA ;IF IT MATCHES
3F63- 48 3150 BEQ NEXTAGR ;STORE SEARCH PARAMETERS.
3F64- 8E 88 3F 3160 STX \AGR+1 ;TRANSFER GR$ NODE FROM OLD POSITION
3F65- F0 09 3170 BEQ SHIFT ;TO NEXT ORDERED POSITION.
3F66- CA 3180 DO\SHIFT DEX ;
3F67- BD 7A 02 3190 LDA GRAPH,X ;
3F68- E8 3200 INX ;
3F69- 9D 7A 02 3210 STA GRAPH,X ;
3F70- CA 3220 DEX ;
3F71- EC 66 3F 3230 SHIF CPX INGR+1 ;
3F72- D0 E2 3240 BNE DO\SHIFT ;MOVE START OF UNORDERED PART OF GR$.
3F73- E8 66 3F 3250 INC INGR+1 ;MOVE THE ORIGINAL CHARACTER.
3F74- 68 3260 PLA ;
3F75- 9D 7A 02 3270 STA GRAPH,X ;CONTINUE SEARCH TO END OF GR$.
3F76- 9D 7A 02 3280 JNG LDX #0 ;(LENGTH OF GR$)
3F77- E8 3290 BEQ NEXTAGR ;
3F78- EC 4D 3F 3300 CPX GL+1 ;
3F79- D0 D8 3310 BNE FINDANODE ;ADVANCE IN TLIST.
3F80- C8 3320 INY ;
3F81- C8 3330 CPY IL+1 ;(# OF NODES)
3F82- D0 CD 3340 BNE TORRER ;
3F83- 60 3350 RTS ;
3F84- 60 3360 ;
3F85- A9 42 3370 LDA #B' ;
3F86- 85 42 3380 STA #NAME ;
3F87- A9 52 3390 LDA #R' ;
3F88- 09 80 3400 ORA #I28 ;
3F89- 85 43 3410 STA #NAME+1 ;
3F90- 20 1D 3E 3420 JSR FIND ;GET LENGTH OF GR$.
3F91- AE 4D 3F 3430 LDX CL+1 ;
3F92- CA 3440 DEX ;
3F93- AC 8A 3E 3450 LDY LL+1 ;SET CURRENT LENGTH OF BR$
3F94- C8 3460 TAY ;AS LOOP LIMITER.
3F95- 8C D5 3F 3470 STY BR\END+1 ;
3F96- A0 00 3480 LDY #0 ;SET INDEXES FOR REWRITING BR$.
3F97- 8C BA 3F 3490 STY NEW\BR+1 ;
3F98- C8 D3 3F 3500 STY OLD\BR+1 ;
3F99- B8 3510 CLV ;CLEAR FLAG FOR COPYING.
3F100- B8 3520 ;REAR CHARACTER FROM BR$
3F101- B8 3530 ;
3F102- B1 42 3540 LDA (NAME),Y ;AND REWRITE IT FOR NEW VERSION.
3F103- A0 00 3550 NEW\BR LDA #0 ;
3F104- 91 42 3560 BVS COPY ;IF CHARACTER IS NOT DELETED YET
3F105- 70 D0 3570 JSR COPY ;AND IS SAME AS THE GR$ CHARACTER,
3F106- DD 7A 02 3580 BEQ COPY ;DECREMENT BR$ LENGTH
3F107- D0 08 3590 BEQ COPY ;DEC LL+1
3F108- CE 8A 3E 3600 BEQ TRACE*1 ;AND SET THE FLAG FOR COPYING.
3F109- 2C E5 3E 3610 BVS BUMP\OLD ;OTHERWISE INCREMENT WRITE INDEX
3F110- 70 03 3620 LDA #NEW\BR+1 ;
3F111- EE BA 3F 3630 INC NEW\BR+1 ;
3F112- EE BA 3F 3640 ;ALWAYS INCREMENT READ INDEX,
3F113- EE D3 3F 3650 BUMP\OLD INC OLD\BR+1
3F114- D0 03 3660 LDY #0 ;AND CONTINUE THROUGH BR$.
3F115- C0 00 3670 BREVND CPY #0 ;
3F116- D0 DF 3680 BNE REWRITE ;REPEAT FOR NEXT GR$ CHARACTER.
3F117- E0 DF 3690 CPX #0 ;
3F118- E0 00 3700 BEQ START ;STORE FINAL LENGTH OF BR$
3F119- D0 CA 3710 LDA LL+1 ;IN BR$ HEADER.
3F120- AD 8A 3E 3720 BNE START ;
3F121- AD 8A 3E 3730 BNE START ;
3F122- AD 01 3740 LDI #1 ;
3F123- AD 01 3750 STA (L\POINTER),Y ;
3F124- 91 54 3760 ;
3F125- 60 3770 ;

```



sensational software



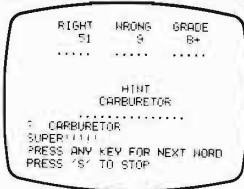
CAI Programs Vol I

Cassette CS-4201 \$11.95

Requires 16K Apple II or Apple II Plus



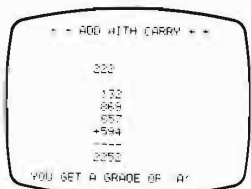
U.S. Map. Identify states and their capitals.



Spelling. Study aid with your list of troublesome words.



Math Drill. Arithmetic drill and practice with large or small display.



Add With Carry. Drill and practice on sums requiring numbers to be carried.

Ecology Simulations - I

Disk CS-4706, \$24.95

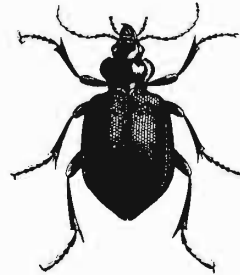
Requires 48K Applesoft in ROM or Apple II Plus

Steri
STERL allows you to investigate the effectiveness of two different methods of pest control—the use of pesticides and the release of sterile males into a screw-worm fly population. The concept of a more environmentally sound approach versus traditional chemical methods is introduced. In addition, STERL demonstrates the effectiveness of an integrated approach over either alternative by itself.

Pop
The POP series of models examines three different methods of population projection, including exponential, S-shaped or logistical, and logistical with low density effects. At the same time the programs introduce the concept of successive refinement of a model, since each POP model adds more details than the previous one.

Tag
TAG simulates the tagging and recovery method that is used by scientists to estimate animal populations. You attempt to estimate the bass population in a warm-water, bass-bluegill farm pond. Tagged fish are released in the pond and samples are recovered at timed intervals. By presenting a detailed simulation of real sampling by "tagging and recovery," TAG helps you to understand this process.

Buffalo
BUFFALO simulates the yearly cycle of buffalo population growth and decline, and allows you to investigate the effects of different herd management policies. Simulations such as BUFFALO allow you to explore "what if" questions and experiment with approaches that might be disastrous in real life.



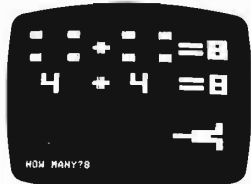
CAI Programs Vol II

Cassette CS-4202 \$11.95

Requires 16K Apple II or Apple II Plus



European Map. Identify countries and their capitals.



Meteor Math. Learn math skills by destroying menacing meteors.

Music Composing Aid. Make and play your own music on the Apple. No additional hardware required. Includes a sample from Bach's Tocatta & Fugue in D minor.



Ecology Simulations - II

Disk CS-4707 \$24.95

Requires 48K Applesoft in ROM or Apple II Plus

Pollute
POLLUTE focuses on one part of the water pollution problem; the accumulation of certain waste materials in waterways and their effect on dissolved oxygen levels in the water. You can use the computer to investigate the effects of different variables such as the body of water, temperature, and the rate of dumping waste material. Various types of primary and secondary waste treatment, as well as the impact of scientific and economic decisions can be examined.

Rats
In RATS, you play the role of a Health Department official devising an effective, practical plan to control rats. The plan may combine the use of sanitation and slow kill and quick kill poisons to eliminate a rat population. It is also possible to change the initial population size, growth rate, and whether the simulation will take place in an apartment building or an entire city.

Malaria
With MALARIA, you are a Health Official trying to control a malaria epidemic while taking into account financial considerations in setting up a program. The budgeted use of field hospitals, drugs for the ill, three types of pesticides, and preventative medication, must be properly combined for an effective control program.

Diet
DIET is designed to explore the effect of four basic substances, protein, lipids, calories and carbohydrates, on your diet. You enter a list of the types and amounts of food eaten in a typical day, as well as your age, weight, sex, health and a physical activity factor. DIET is particularly valuable in indicating how a diet can be changed to raise or lower body weights and provide proper nutrition.



CAI Programs I and II

Disk CS-4701, \$24.95
Requires 32K Integer Basic

This disk contains all 7 programs from cassettes CS-4201 and CS-4202.

Note: The ecology simulations programs are not available on cassette.

Stock & Options Analysis

Disk CS-4801, \$99.95
Requires 32K Applesoft or Apple II Plus

This is a comprehensive set of four programs for the investment strategy of hedging listed options against common stocks. A complete description is in the TRS-80 section. Available August 1981.

Order Today

To order any of these software packages, send payment plus \$2.00 postage and handling per order to Creative Computing, Morris Plains, NJ 07950. Attn: Cleo. Visa, MasterCard and American Express orders may be called in toll-free.

Order today at no risk! If you are not completely satisfied, your money will be promptly and courteously refunded.

Creative Computing Software
Morris Plains, NJ 07950
Toll-free 800-631-8112
In NJ 201-540-0445

creative computing software

Apple is the registered trademark of Apple Computer, Inc.

PET Menu and Tape Timer

by Dale De Priest

This article describes a menu program that allows rapid access to any program on either side of a cassette tape. In addition, a tape timer is presented that supplies the fast forward times for the menu program. These two programs feature advanced cassette control and use the WAIT command extensively.

MENU and TAPE TIMER
require:
PET
cassette

Every disk operating system has a way of maintaining a directory or menu on each disk. This directory allows you to find out the names of all the programs on the disk, and allows the disk operating system to locate these programs. A workable tape operating system should also provide a directory for its tapes.

The PET operating system is capable of locating and loading programs from tape but it is slow. We need some way to index the programs so that they can be located with the fast-forward mode. Unfortunately Commodore didn't provide for an index counter on their tape drive. So, I decided to develop a program to simulate an index counter and the directory or menu operation.

Menu Program

My original goal in creating the menu program was to provide a loader that would allow you to access any portion of the tape in as much time as it would take to get to the second program on a conventional tape. Since an average program takes about 90 seconds to load, I needed a program that would load in 30 seconds, thereby allowing 60 seconds for the fast-forward time.

Listing 1

```

50 REM DOCUMENTED SAMPLE MENU PROGRAM
52 REM WRITTEN BY DALE DE PRIEST
54 :
56 :
58 REM VARIABLE USAGE
60 REM A = INPUT NUMERIC VALUE
62 REM B = LOCATION OF INPUT BUFFER
64 REM C = CASSETTE #1 SWITCH FLAG
66 REM D = ADDRESS POINTING TO NUMBER OF CHARACTERS IN INPUT BUFFER
68 REM F = NUMBER OF PROGRAMS ON THIS ENTIRE TAPE
70 REM G = THE NUMBER OF THE FIRST PROGRAM ON THE OTHER SIDE
72 REM J = VARIABLE FOR 'FOR' LOOP
74 REM T1 = TIME AT THE BEGINNING OF THE SEARCH
76 REM A$ = INPUT VARIABLE
78 REM B$ = A MESSAGE
80 REM C$ = THE LOAD/RUN COMMAND STRING
82 REM A$(F) = NAMES OF ALL THE PROGRAMS ON THIS TAPE
84 REM ACF) = FAST FORWARD SEARCH TIMES FOR EACH PROGRAM
86 :           IN TENTHS OF SECONDS
88 :
90 REM INITIALIZE AND RANDOMIZE THE RANDOM NUMBER GENERATOR
100 F=36:G=20:A=RND(-T1):PRINT"MENU":DIMA$(F),ACF)
106 :
108 REM CORRECTS ADDRESSES FOR OLD AND NEW ROMS
110 B=527:C=519:D=525:IFPEEK(5E4)THEND=158:B=623:C=249
116 :
118 REM PRINT THE NAMES OF ALL THE PROGRAMS AND THE SELECTION NUMBER
120 FORJ=1TOF:IFJ=21THENPRINT"END"
130 READA$(J):READA(J):IFJ>20THENPRINTTAB(20);
140 PRINTJ:A$(J):NEXT:IFF>20THENFORJ=FTO39:PRINT:NEXT
150 PRINT"ENTER THE NUMBER BESIDE THEM"
156 :
158 REM THE 4 BLANKS AFTER 'WISH' ARE SHIFTED BLANKS
160 INPUT"PROGRAM YOU WISH   ****":A$=VAL(A$):IFAC<10RA<0THEN160
170 IFA<0>1THEN200
176 :
178 REM THIS SIMULATES THE RUN/STOP KEY
180 C$="LOAD"+CHR$(13)+"RUN"+CHR$(13):FORJ=0TO8:POKEB+J,ASC(MID$(C$,J+1))
190 NEXT:POKEB,J:END
198 :
200 B$="PLEASE PRESS STOP ON THE CASSETTE:"PRINTB$:WAIT59408.16
206 :
208 REM THE OPERATING SYSTEM AUTOMATICALLY GIVES MOTOR CONTROL TO THE USER
210 PRINT"PRESS THE FAST FWD BUTTON":WAIT59408.16.16
216 :
218 REM THE POKE RETURNS MOTOR CONTROL BACK TO THE COMPUTER
220 PRINT"LOOKING FOR "A$(A):POKEC,52:T1=T1
240 IFT1-T1<A<0)*6THEN240
246 :
248 REM THE COMPUTER ACTUALLY STOPS THE TAPE MOTION
250 POKES9411.61:PRINT" "MID$(B$,9):WAIT59408.16:IFAC<GTHEN180
260 PRINT"TURN THE CASSETTE OVER":FORJ=1TO3000:NEXTJ:GOTO180
496 :
498 REM SAMPLE NAMES AND SEARCH TIMES
500 DATACOVER16.1,NAB1.52,FIRE!,133,ALIENS!,195,BONZO!,246,CATCH!,309
510 DATACOVER17.349,POLICE!,393,SPOT,460,RULER,507,LETTER,539,MERGE,594
520 DATAPACK,630,COVER18,661,DROMEDA!,691,JOUST,750,WEATHER,797
530 DATAHI RES,856,SHEEP,900
536 :
538 REM THESE PROGRAMS ARE ON THE BACK SIDE OF THE TAPE
540 DATACOVER19,938,FROG!,896,GODZILLA!,847,MINER!,795,RAIL,746,GROOKA,703
550 DATAGBOOK8,663,COVER20,617,MUSIC!,584,BETS,528,CHECKERS,470
560 DATACURVES,420,EQUIP,375,COVER21!,384,CAPTURE!,239,DANCE!,178,
    BOSWAIN,93
564 :
566 REM THIS LINE IS NOT USED BY THE PROGRAM BUT PROVIDES DATA
    FOR THE USER IF
568 REM YOU WISH TO ADD AN ADDITIONAL PROGRAM TO THIS TAPE
570 DATAEND OF TAPE,948,1790
    
```

Listing 1 shows the result of this effort. It allows for up to 40 programs to be located on the same tape and will find any one of 30 programs in approximately 60 seconds. This program should be the first program on each of your cassettes, but it could be on a separate tape and contain the menu for several tapes. This would be desirable for tapes such as Cursor Magazine. Several interesting things can be learned from this program, so let me show you how it works.

Line 100: The statement `A=RND(-TI)` doesn't have anything to do with the rest of the program. It is simply an easy way to insure that the RND function is randomized for every program on the tape.

Line 110: The variable C points to the location of a flag that the PET uses to determine whether it or the user has control of the cassette motor. When you push one of the switches down, the PET turns on the motor for you. Since it believes that the operator should have control, it won't let the program stop the drive unless variable C is changed. A zero in this location means that the operator has control; any other value gives control to the PET.

Line 170: By convention the first program after the menu will contain a fast-forward value of one. This means no fast-forward is required.

Line 180-190: This is a special trick on the PET. This line stores the two commands LOAD and RUN in the PET's input buffer. You may wonder why the simpler method of

180 POKE B, 131:POKE D,1: END

is not used. This line forces the PET to respond as if the operator just hit the RUN/STOP key. The PET will load and run the next program. Unfortunately BASIC 4.0 directs this command to the disk, so you must put the command in the buffer yourself to make this program work with all versions of PET software. This technique is frequently useful when one program wants to turn control over to another program.

Line 200: This is the mysterious WAIT command. Here it is used to detect whether or not a key has been pressed. 59408 is the address whose contents will change and 16 is the decimal equivalent for the bit (bit 4) which will change to a 1 when the key is pressed. (If the bit were changing from 1 to 0, then the command would be `WAIT 59408,16,16`.)

Line 500-540: These lines contain the DATA statements that define the program names and the search times. The names recorded here do not need to be exactly the same as the name the program is stored under. They are only used by this program and not by the actual load routine. When setting up a tape for the first time you may not yet know the names of the programs that you are going to put on that tape. Be sure that you add enough filler to allow room to add these names later. I usually copy the menu from a different tape and then change the names as I need to. Setting F in line 100 will avoid confusion of the real contents of tape and the data that may be present simply to take up space. As each program is added I go back and update the menu program. This is easy since the long leaders prevent wiping out the second program.

This program works fine, but there is no way to easily determine what the fast-forward times should be. Now look at listing 2.

Tape Timer

This program supplies the fast-forward times for the menu program. In addition, it will provide a listing of your tape along with the load times for each program. The significant details are outlined below.

Line 120: This line finds headers and then measures the length of time between them. You should notice that although a program LOAD can distinguish between programs and data, an OPEN cannot. The last statement on the line initializes the program name variable.

Line 130: This line shows a method of getting the name of a program or data header into the program. The technique is to build a string right from the header buffer area. This is a useful line of code and shows you how to read beyond the sixteenth character of the header.

Line 150: H is the variable that is used to accumulate the time for all programs. Fast forward on a cassette tape is not linear. That is, there is no direct correlation between fast-forward times and normal play times. Although there is probably a mathematical formula that could be developed to calculate this relationship, the program uses a group of straight lines to approximate the curve. It then calculates a proportional relationship between these values. Empirically I have determined that the relationship between H and M for the first 200 seconds is .78.

Therefore if a load time of 100 seconds were measured, then this program would calculate the fast-forward time to be 7.8 seconds. All fast-forward times are calculated in tenths of seconds.

Lines 160-210: These lines continue the straight line approximations to the curve. Each value has been determined empirically to stop the tape 10 to 15 seconds in front of a header. This program should be fairly accurate for cassettes up to 60 minutes long. Longer cassettes use thinner tape which will invalidate the times.

Line 230: This line assumes that the menu program was stored with the name MENU as the first four characters. This is a convention that I always follow. Checking for this name allows the program to recalibrate itself to do all timings just after the menu load. The program will also work correctly when your program menu is on a different tape since the time is then from the beginning of tape.

Line 250: This program assumes that tapes have been properly ended with an end-of-tape header. The search times may then be copied to enter into the menu program. This program uses a feature of the cassette unit that causes it to shut off automatically when it reaches the end of a tape. When this happens, the program calculates the actual length of the tape. If you were to fast-forward the tape from the other end, it would then compute the fast-forward times. These times should be entered on a MENU program located at the beginning of the back side of the tape.

Putting It All Together

Now that you have seen the two programs, let's see how we can put them to work. First, you must build your tape by putting a MENU on the front of the tape followed by several programs. The tape must end with a file named "END OF TAPE." The easiest way is to type the following command in immediate mode.

```
OPEN1,1,2,"END OF TAPE":CLOSE1
```

Then load the TAPE TIMER program and time the tape. Write the front side search times as well as the names on a piece of paper. Hit the space bar and record the reverse side search times. Load in a copy of the MENU program and modify it with the correct names and reverse search times. Set the values of F and G to the correct values. Store

PET FEATURE

this new program as the first program on the back side.

Next, fast-forward the tape to the beginning of side one. Change the search times to the correct ones for the front side. Change variable G, and save this new update over the top of the old MENU program. You now have a complete tape that can be searched from either direction. Of course you could also add programs to the back of the tape.

To use this program, rewind the tape to either end. Then simply use the SHIFT/RUN keys to get the MENU program in and running. (This will not work for BASIC 4.0 users.) The MENU program will then do the rest. This way, someone who is not familiar with programming will be able to use your tapes.

Manual Updating

There will be times when you want to add a program to the end of a tape. Simply position your tape just past the current last program, either by loading or verifying it, and then save your program. Be sure to add a new "END OF

TAPE" header. You could then load in the tape timer and retape your tape. Then go back and update your MENUs. There is an easier way. It takes a little bit of advance planning but it works like a charm.

When the tape timer program finds the end of tape it outputs two things

about that point: the fast-forward search time and the total load time. When you create your menu program for that tape you should add these two numbers to the program. They can be added as additional data statements that are never read or they can be added in a remark. Now when you wish to add a program you will be ready.

Listing 2

```

10 REM TAPE TIMER
12 REM WRITTEN BY DALE DE PRIEST
14 :
14 :
16 REM VARIABLE USAGE
20 REM C = CASSETTE #1 SWITCH FLAG
22 REM H = ACCUMULATED TOTAL LOAD TIME
24 REM I = VARIABLE FOR 'FOR' LOOP
26 REM J = COUNTER FOR NUMBER OF PROGRAMS
28 REM M = COMPUTED SEARCH TIME
30 REM T1 = INITIAL TIME OF TIMER
32 REM A$(24) = NAMES OF EACH OF THE PROGRAMS
34 REM R(24) = LOAD TIMES FOR EACH OF THE PROGRAMS
36 REM LR = LENGTH OF RECORD IN BYTES
38 :
48 :
50 PRINT"PLEASE INSERT AND START THE TAPE YOU":PRINTTAB(13)
50 PRINT"WISH TO TIME IN TAPE # 1
52 :
58 REM DEFINE FUNCTION FOR ADDRESS SEEKING AND CORRECT FOR OLD AND NEW FORMS
70 DEFFNAD(C)=PEEK(C)+PEEK(C+1)*256:C=519:IFPEEK(5E4)THENC=249
74 :
76 REM WAIT UNTIL TAPE STOP IS DEPRESSED THEN WAIT FOR THE PLAY BUTTON
78 REM THEN GIVE MOTOR CONTROL BACK TO THE COMPUTER
80 WAITS9408,16:WAITS9408,16,16:POKEC,52
90 DIMA(30),R$(30):A$(0)="BEGINNING OF TAPE "
100 PRINT"NAME SEARCH TIME LOAD TIME
106 :
108 REM ALL TAPES MUST END WITH A FILE NAMED "END OF TAPE"
110 PRINTA$(J):INT(M):IFLEFT$(A$(J),11)="END OF TAPE"THEN250
114 REM INITIALIZE TIME & START THE TAPE MOVING
115 T1=TI:POKE9411,53
116 :
118 REM SPACE OVER THE PROGRAM FILE WITHOUT READING
120 IFT1-T1<L9GOTO120
122 :
124 REM OPEN WILL FIND BOTH PROGRAM AND DATA FILES
125 J=J+1:OPEN1,1:A(J)=(T1-T1)/60:A$(J)="
126 :
128 REM THIS LINE FINDS THE PROGRAM NAME FROM THE TAPE HEADER
130 FORI=639TOD59:A$(J)=A$(J)+CHR$(PEEK(I)):NEXT:CLOSE1:PRINTINT(A(J))
140 GOSUB150:LR=1:IFPEEK(634)=1THENLR=.0182*60*(FNAD(637)-FNAD(635))+200
145 GOTO110
146 :
148 REM THESE LINES ACTUALLY COMPUTE THE FAST FORWARD TIMES
150 M=H+A(J):M=M:IFM<200THENM=.78*M:GOTO230:REM <200
160 M=M-200:IFM<200THENM=153+.67*M:RETURN:REM 200-400
170 M=M-200:IFM<200THENM=285+.53*M:RETURN:REM 400-600
180 M=M-200:IFM<200THENM=401+.54*M:RETURN:REM 600-800
190 M=M-200:IFM<200THENM=507+.49*M:RETURN:REM 800-1000
200 M=M-200:IFM<200THENM=605+.46*M:RETURN:REM 1000-1200
210 M=M-200:M=740+.42*M:RETURN:REM >1200
220 IFLEFT$(A$(J-1),4)="MENU"THENH=0:M=1
240 RETURN
244 :
246 REM TURN THE MOTOR ON AND WAIT FOR THE AUTO SHUT OFF FEATURE
248 REM "H" NOW REPRESENTS THE TOTAL ACCUMULATED LOAD TIMES
250 T1=TI:POKE9411,53:PRINTINT(H)
260 PRINT"WAIT UNTIL THE PHYSICAL END":WAITS9408,16:H=H+(T1-T1)/60-10
270 PRINT"PRESS SPACE FOR REVERSE SEARCH TIMES":WAITS9410,4,4
276 :
278 REM TOTAL TAPE LENGTH = LOAD TIMES + BLANK TAPE AT END + MENU LOAD TIME
280 IFLEFT$(A$(1),4)="MENU"THENH=H+A(2)
290 PRINT"CORRECTED TAPE LENGTH"INT(H)
296 :
298 REM THE REVERSE SIDE SEARCH TIMES ARE NOW CALCULATED
300 FORI=1TOJ:H=H-R(I):M=M:IFM<200THENM=.78*M:GOTO320
310 GOSUB160:IFM>740THENM=M+3
320 PRINTA$(I):INT(M)
330 IFI/22=INT(I/22)THENPRINT"PRESS SPACE TO CONTINUE":WAITS9410,4,4
340 NEXT

```

AT LAST... ...FOR OSI

For investors
and financial managers

Stock portfolio analysis

\$150.00

- in your office - instant valuations
- compound growth measurement
- pertinent company operating statistics
- input data electronically

Stock financial statement analysis

\$250.00

- input your interpretation of financial data
- analyze up to 10 years of data
- see mean, trend and stability
- data stored on DBM system
- input data electronically

On-line data retrieval

\$50.00

Accounting package

\$150.00

DBM system

\$200.00

for 8" floppy/hard disc
under OS65U

write for details

Genesis Information Systems, Inc.

P.O. Box 3001 • Duluth, MN • 55803
Phone 218-724-3944

First position the tape to the correct point and then type in the following line to save your program.

TI = TI:SAVE "NAME" :?(TI - T)/60

The program will save in the normal fashion, but it will also tell you how long it took. Note this time, rewind the tape and load in your menu. The menu can be updated by adding your new program name, changing variables F and G and using the search time number that you had previously entered. Then calculate a new search time for the next addition, add the save time to the accumulated time, and change your note in the program. Use this time and the listing for the tape timer to calculate your new time.

Simply pick the line you need from the choices beginning at line 150. The remarks at the end of each line should make your choice easy, and the PET calculator mode will make the calculation easy. The value of M in the equation is the length of the tape minus the left number in the remark statement. For example, suppose your time to that point is 682 and you just saved an 87-second program. Adding those two numbers gives you 769 seconds. That

places us on line 180 of the listing so we must subtract 600 to get M (169). The new search value is $401 + .54 \cdot 169$ or 492 (49.2 seconds).

By the way, if the record and play buttons weren't already pressed when you typed in the save command, you'll have to be very careful to get the times to come out right. Type the line in but do not hit the return key. Then push the record and play keys at the same time that you push the return key.

Final Thoughts

When typing in the menu program be sure to remove all remarks. Your program will load about twice as fast that way. I hope that you get as much use out of this as I have. These programs have greatly enhanced my use of the PET Tape Operating System.

Dale De Priest is the manager of Circuit Development and Document Control at ISS Sperry Univac. He has an associate degree in Electrical Engineering Technology from Central Technical Institute in Kansas City. He can be contacted at: 611 Galen Drive, San Jose, CA 95123.

MICRObits

(Continued from page 21)

OSI Screen Editor

Never retype a line with Edit-All. Dynamic insert/delete corrects any line on the screen. With user-definable scroll window and full cursor control and screen clear. For OS-65D systems - \$19.95. Universe (by *Galaxian* author) plays like *Cobra Copter* - \$14.95 tape/disk.

DMP Systems
319 Hampton Blvd.
Rochester, N.Y. 14612

OSI Superboard II, C1P

Star Chase: avoid and shoot the oncoming asteroids. 4K version - \$7.95; 8K version - \$9.95. *Star Fire*: Plays like *Defender*. 4K - \$9.95; 8K - \$12.95. *Spanish I*: Computer dictionary of Spanish words. 8K - \$9.95 cassette only. Send for free catalog.

Swany's OSI Software
2652 37th West
Seattle, WA 98199

(Continued on page 105)



EDITRIX + GRAPHTRIX = THE MOST POWERFUL WORD PROCESSOR
THIS SIDE OF A NEWSPAPER COMPOSITION ROOM

<p>EDITRIX™ TEXT EDITOR</p> <p>EASY TO USE</p> <ul style="list-style-type: none"> • HELP! Key • Friendly, COMPLETE instructions that you or your secretary can understand. • Easy to remember 1 or 2 keystroke commands. • See your document formatted on the screen AS YOU EDIT IT. <p>POWERFUL</p> <ul style="list-style-type: none"> • 250 Column Horizontal Scrolling. • Automatic Graphic Insertion and Formatting. • Automatic Footnote Insertion. • Underline - Superscript - Subscript - Search - Replace - Block Move. • Full Printsize, Emphasis, Justify, Margin and Cursor Control. <p>FLEXIBLE</p> <ul style="list-style-type: none"> • Capital letters with ESC or Shift Key modification. • To be supported by Data Transforms new headline generator coming soon. • Printout through GRAPHTRIX to 11 different Printers WITHOUT CHANGING YOUR TEXT FILE! <p>REQUIRES: Apple II with 48K, Applesoft in ROM, DOS 3.3 and the GRAPHTRIX Matrix Graphics System</p>	<p>GRAPHTRIX™ TEXT PRINTER AND GRAPHICS SCREEN DUMP</p> <p>EASY TO USE</p> <ul style="list-style-type: none"> • Complete READABLE documentation. • Fully Menu Driven. • Self-running Introduction and Demonstration. <p>POWERFUL</p> <ul style="list-style-type: none"> • Graphic Magnification, Normal/Inverse, Page Centering, Hi and Low Crop Marks, Title String. • Automatic Formatting of Graphics in your Document. • Print Size, Emphasis, Underline, Superscript, Footnotes, Chapters, controlled from your text file. <p>FLEXIBLE</p> <ul style="list-style-type: none"> • Prints ANY HI-RES Graphic your Apple II can create. • Formats Text files from Applewriter OR EDITRIX. • Use as a Menu Driven Screen Dump OR from in YOUR OWN Applesoft Program. • Compatible with 11 different Matrix Line Printers AND 7 different Parallel Interface Cards. <p>REQUIRES: Apple II with 48K, Applesoft in ROM, DOS 3.3 and one of the following line printers: EPSON MX-70/MX-80/MX-100, ANADEX 9500/9501, IDS 440G/445G/460G/560G, CENTRONICS 739, MPI 88G, SILENTYPE.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

FROM DATA TRANSFORMS, INC., THE GRAPHICS LEADER


EDITRIX and GRAPHTRIX are the trademarks of Data Transforms Inc., a division of Solarstatics Inc. Apple II and Applewriter are trademarks of Apple Computer Inc. © Copyright 1981 Data Transforms, Inc. 616 Washington, Suite 106, Denver, CO 80203 (303) 832-1501 All Rights Reserved

DR. DOBB'S JOURNAL

For Users of Small Computer Systems

with topics like:

- Small-C
- FORTH
- Lisp
- CP/M
- S-100
- Cross-assemblers
- 8086/8088
- 16-bit tech.
- Pascal
- Assembler
- Compilers
- Algorithms
- 8080/Z80
- 6502
- etc.



Each issue includes:

- valuable software tools
- algorithms & problem solving
- industry news
- important product reviews

With in depth coverage of:

- telecommunications
- systems programming
- language development
- machine independent programs

and much, much more!

Yes! Please enter my subscription for T4

1 yr. (12 issues) \$25 2 yrs. \$47 (save \$13 off newsstand)

Please bill me I enclose check/money order

Name _____

Address _____

City _____ State _____ Zip _____

Are you ready?

DDJ, the world's foremost micro-computer publication, has been working for years to prepare its readers to be innovators, to lead the wave of breakthroughs in our changing technology.

Every issue of *Dr. Dobb's Journal* helps one to understand the nuts and bolts of small computer systems. We offer entire listings of valuable software: our pages have included compilers, cross-assemblers, editors, new languages, hardware interfaces and more – usually before anyone else thinks of them!

Even more important!

As valuable and significant as all those things are, there is an even greater reason for you to join forces with *DDJ*. That is the keen, responsive readership. Our subscribers share insights, correspond, and contribute to one another's work, more than any other group we know. They treat *Dr. Dobb's Journal* as a "hands-on" publication.

This warm cooperation has done more to refine software products, and generally to advance the state of microcomputer technology, than perhaps any other resource. And it is available to you through our pages!

For the straight Facts . . .

If you are a serious computing professional or enthusiast, then you should take a very close look at what *DDJ* offers you. We've been on the cutting edge since 1976.

MICRO™

Hardware Catalog

Name: **16 Plus Board**
System: Radio Shack Color Computer TRS-80
Memory: 16K

Description: The *16 Plus Board* expands the Color Computer memory from 16K to 32K quickly, easily and inexpensively. There is no soldering. It plugs in neatly under the RF shield cover. There are no software modifications to existing software and it is completely compatible with anticipated disk systems. Plus it allows the graphics display to reside anywhere within the 32K of memory.

Price: \$84.95
Includes memory board and instructions

Available:
Computerware
P.O. Box 668
Encinitas, CA 92024
(714) 436-3512

includes a bookshelf for manuals, a top support for a printer or CRT terminal and a slide out stabilizer for servicing heavy units in extended position. The hinged rear door can be lifted off entirely for service access. Computer units are placed on sliding shelves, eliminating the need to mount rack slices on each chassis. An AC outlet strip with 6 grounded outlets, power switch indicator and circuit breakers are included.

Price: \$795 list, OEM/Dealer discounts available
Includes Shepard casters.
Options - Drawers which fit on any shelf in place of a computer unit. Standard finish gunstock walnut laminated plastic. Other colors/woodgrains and finishes and smoked tempered glass doors available.

Available:
Micro Solutions, Inc.
6305 Arizona Ave.
Los Angeles, CA 90045

Name: **CBM 8032P (Bi-Directional Printer)**

Memory: 1K RAM
Description: Low cost, high speed bi-directional printer. This new printer connects directly to a Commodore computer and is a 136-column printer with tractor and friction feed. The 8032P is dot-matrix, and prints 150 characters per second, prints all alphabetic and graphic characters available with a Commodore computer, as well as user-defined characters.

Price: \$995.00

Available:
Commodore Business Machines, Inc.
Computer Systems Division
[through Commodore authorized dealers.]

Name: **Model 5000/D Computer Console Cabinet**

Description: Handsome, functional and portable cabinet, specially designed to house most of the components of a microcomputer system. In addition to providing ample space for the computer and disk drive units, the console

Name: **AMS8000 Megabyte**
System: 8" floppy disk drive system
Memory: 2 megabytes, expandable to 4 megabytes
Language: DOS 3.3, CP/M, Pascal
Hardware: Shugart 850 drives, SVA ZVX4 Controller Card

Description: Dual-density, double-sided drive packages support single-density and single-sided disks.

Price: \$2995
Includes cable, power supply fan in Apple-matching cabinet

Available:
All Apple Dealers

Name: **DATASAVERTM AC Power Backup Unit**

Hardware: 90 Watt Micro-UPS for Microcomputers

Description: The *DATASAVERTM* continuously conditions and monitors the micro-

computer line voltage while charging its own battery. Upon a brownout or loss of line voltage, the *DATASAVERTM* automatically transfers to battery operation within 1/2 cycle to assure uninterrupted AC power. Hold-up time is at least 15 minutes for Apple II, TRS-80 III, and OSBORNE 1 size units, and five minutes for Apple III.

Price: \$395 (120VAC@60Hz)
\$424 (220VAC@50Hz)
Includes proprietary voltage surge and transient suppressor

Available:
Microcomputer System Dealers
[substantial wholesale discounts available] or factory direct:
Cuesta Systems, Inc.
3440 Roberto Court
San Luis Obispo, CA 93401
(805) 541-4160

data acquisition system. It consists of an 8-bit successive approximation A/D converter, an 8-channel multiplexer and 8x8 random access memory. The analog to digital conversion takes place on a continuous, channel sequencing basis. Data is automatically transferred to on-board memory at the end of each conversion. No A/D converter could be easier to use.

Price: \$129 (Texas Residents add 5% sales tax.)

Available:
Applied Engineering
P.O. Box 470301
Dallas, Texas 75247
(214) 492-2027

Name: **Prism 80 and Prism 132 Column Printers**

Language: Full 96 ASCII characters/ additional resident sets

Description: Up to 200 CPS data speed (optional), 150 CPS proportional spacing, correspondence print quality; 9-wire staggered print head; cut sheet and pinfeed fanfold paper types; programmable features; print mode, line spacing, forms length, intercharacter spacing, margins, tabs; 10/12/16.8 CPI plus double width characters; fixed and proportional spacing; text justification; forms length control; line spacing; slew rate 7.5 IPS; parallel centronics compatible serial, RS-232 (DTR or XON/XOFF); baud rate up to 9600; graphics optional 84x84 DPI; cartridge ribbon. Options: cut sheet feeder, graphics, data speed, color printing, 9-wire in line head, special character set. [Above subject to change without notice.]

Price: \$1995
Includes 132-column with color, sheet feeding and data speed. (\$899 for 80-column without options.)

Available:
Representatives and Distributors;
over 900 retail outlets
(800) 258-1386

Name: **TNW 1000, 2000, 232-D Dual Port and TNW 103 Low Speed Modems**

Description: TNW's serial interfaces are designed to interface RS-232 devices to the Commodore PET/CBM and other IEEE-488 bus-capable computers. Each unit can share the IEEE bus with other peripheral devices and each provides a connector for daisy-chaining. TNW's 103 is a full-service, auto answer, auto dial modem.

Price: \$129 TNW-1000;
\$229 TNS-2000;
\$389 TNS-103;
\$369 TNW-232D.

Available:
TNW Corporation
3444 Hancock Street
San Diego, CA 92110
(714) 296-2115
TWX 910-335-1194

Name: **8-Bit, 8-Channel A/D System**

System: Apple
Hardware: A/D System
Description: The Applied Engineering A/D board is an 8-bit, 8-channel, memory-buffered,

MICRO

Write for MICRO!

MICRO Magazine is actively seeking articles to present to our unique audience of sophisticated computerists. We welcome contributions on any aspect of 6502/6809 hardware and software. And send us your latest ready-to-key-in program!

Each issue of **MICRO** presents articles on a wide range of systems and topics. In addition, each issue includes a special feature. Our Editorial Schedule for the rest of 1982 is shown below. Please send

material for:		to:	no later than:
September	68000	Marjorie Morse	June 11
October	Education	Marjorie Morse	July 9
November	Atari & Games	Mary Ann Curtis	August 6
December	PET	Loren Wright	September 10

Standard **MICRO** categories that may be included in any issue are:

I/O • BASIC Aids • Machine Language Aids • Graphics
Games • Applications • Programming Techniques • Utilities
Programming Languages • Hardware

Anyone who is interested in writing about networking or online database searching should contact Laurence Kepple, Senior Editor, either at **MICRO** (617/256-5515) or at 71535,231 on the CompuServe network.

Special Note to all MICRO authors: please begin sending your text as well as your programs in machine-readable format. This will enable us to edit the text with the help of word processing. Continue to include a printed text version as you include a printed program listing.

VIC PET APPLE SOFTWARE

NEW GRAPHICS - full screen super graphics package for the VIC. Adds 18 commands to BASIC. Draw detailed pictures using 152 X 160 points. Hires & Multicolor graphic modes on same screen! Switch between text and graphics screens with function keys. Save/load pictures to/from tape or disk. Requires 3K or 8K expander. With sample programs and user's manual. Price \$25; manual only \$5.

VIC/PET VIGIL - Interactive Games Language - Program your own or play the 9 games included. With 60+ powerful commands. Easy to learn. VIC version has color (requires 3K or 8K expander). Complete with user's manual. Price \$35; manual only \$10.

VIC/PET PIPER - the MUSIC MACHINE - Simplest way yet to compose, conduct and play music. Complete music control of notes, rests, volume, repeats, tempo, etc. Written in fast machine language. With sample compositions and user's manual. Price \$25; manual only \$5.

VIC HIRES/MULTICOLOR GRAPHICS UTILITIES - Add graphics to VIC BASIC. Requires no extra memory. Plot points, lines and boxes in fine detail - 104 X 152 points. With sample programs and user's manual. Price \$15; manual only \$5.

PET TINY Pascal PLUS - structured language alternative to BASIC. With Editor, Compiler and Interpreter. All programming constructs supported. With user's manual. Graphics or non-graphics versions. For NEW/4.0/8032. W/graphics(32K) - disk \$50; cassette \$55. W/O graphics(16K+) - disk \$35; cassette \$40; manual only \$10.

NEW APPLE II DYNASOFT Pascal - complete Pascal development system with Editor, Compiler, Interpreter and Supervisor. All programming constructs and data types: scalars, char, string, pointer, integer. Hires, Lores and machine language interface. With sample programs and user's manual. Disk \$50; Disk with complete machine readable source code \$85; manual only \$10.

PET TINY BASIC COMPILER - Produces true 6502 code. Supports all floating point operations. Subset of the full PET BASIC. Compiler listing optional with 16k version (included). With user's manual. For OLD/NEW/4.0/8032 and 8K+. Price \$25; manual only \$5.

PET MACHINE LANGUAGE GUIDE - Now in its ninth printing. Learn the hidden talents of your Old, New or 4.0 Rom PET/CBM. 30+ routines fully detailed. Price \$9 (\$11 foreign).

NEW ABACUS - Beautifully crafted Chinese abacus. Teach children basics of math or use as lovely decoration. With free 50+ page manual explaining use. Price \$14.95 + \$2.00 postage (\$47.00 foreign).



ABACUS SOFTWARE
P. O. Box 7211
Grand Rapids, Michigan 49510
616 / 241-5510



Unless noted, prices are for cassette. Add \$3.00 per disk package. Foreign orders add \$5.00 per package or \$7.00 for manual only. All orders must be prepaid in US Dollars, VISA, MC, ACCESS, Eurocard.

ATARI® OHIO SCIENTIFIC

COMING SOON! AFFORDABLE ATARI SOFTWARE FROM PRETZELLAND. IF YOU'RE TIRED OF PAYING \$30- \$50 FOR QUALITY GAMES, SEND FOR OUR ATARI SOFTWARE LIST TODAY!

--MACHINE CODE GAMES FOR OHIO SCIENTIFIC--
TWO NEW ORIGINAL ARCADE GAMES FROM DAVE EDSON

OUTPOST TAKE THE SHIP MOVEMENT FROM ASTEROIDS, THE SHAMING ALIENS FROM SPACE INVADERS, AND THE RUBBER, BOUNCING SHIP FROM OMEGA SYSTEM AND YOU'RE BEGINNING TO GET AN IDEA OF WHAT THIS ONE IS LIKE! THE FUEL CELL FROM YOUR OUTPOST HAS BEEN STOLEN BY ALIENS AND IS DRIFTING IN SPACE. YOU HAVE TO RECOVER IT AND BRING IT BACK INSIDE. MEANWHILE, SPACE AROUND YOU IS FILLED WITH ALL SORTS OF NASTY THINGS LIKE REPRODUCERS AND BLOCKS, NOT TO MENTION THE ALIENS- UP TO 20 OF THEM BOUNCING AROUND AT A TIME! TO MAKE MATTERS WORSE, THE OUTPOST ITSELF IS CONSTANTLY ROTATING LIKE A STAR CASTLE, SO YOU'VE GOT TO RE-ENTER CAREFULLY, OR YOU'LL GO BOUNCING OFF ACROSS THE GALAXY! 1 OR 2 PLAYER, 8K CIP CASSETTE ONLY! \$11.95

Diamond Run THIS TIME, IT'S "RIP-OFF" IN REVERSE! UP TO FOUR RADAR CONTROLLED GUNS ARE CONSTANTLY CIRCLING AROUND AND AROUND, GUARDING A HUGE VALUABLE DIAMOND INSIDE A FORTRESS. YOU HAVE TO BLAST YOUR WAY IN, STEAL THE DIAMOND, THEN BLAST YOUR WAY BACK TO SANCTUARY. IF YOU MAKE IT, THE DIFFICULTY LEVEL GOES UP AND YOU CAN TRY IT AGAIN, BUT KEEP A LOOK-OUT FOR THE KILLER PLANE! IT WILL FOLLOW YOU OUTSIDE THE FORTRESS AND TRY TO CATCH YOU BEFORE YOU CAN ESCAPE WITH ITS DIAMOND! DIAMOND RUN IS AVAILABLE FOR 8K CIP CASSETTE MACHINES ONLY! \$11.95

Introductory Offer: EXPIRES 5/30/82

HERE'S A SPECIAL DEAL FOR MACHINE CODE LOVERS- TAKE BOTH OF THESE NEW DAVE EDSON SOFT PRETZELS FOR ONLY \$20.00!

PLEASE ADD 10% FOR SHIPPING ON TAPE ORDERS ONLY.

SEND \$1.00 FOR OUR PHOTO-ILLUSTRATED CATALOG OF OHIO SCIENTIFIC SOFTWARE AND GET A \$1.50 CREDIT ON YOUR FIRST ORDER.

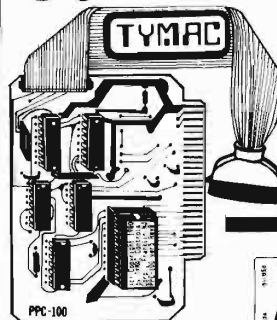
PROGRAMS AVAILABLE ON CASSETTES ONLY!



Pretzelland Software
2005 D WHITTAKER RD.
YPSILANTI, MI. 48197



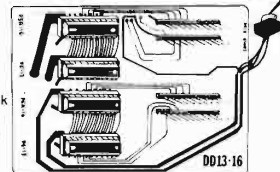
201 839-3478 hardware



PARALLEL PRINTER CARD

A Universal Centronics type parallel printer board complete with cable and connector. This unique board allows you to turn on and off the high bit so that you can access additional features in many printers. Use with EPSON, ANADEX, STARWRITER, NEC, OKI, and other with standard Centronics configuration. \$59.00

DOUBLE DOS Plus



DOUBLE DOS Plus - a piggyback board that plugs into the disk controller card so that you can switch select between DOS 3.2 and DOS 3.3.

Nothing needs to be soldered, just plug in and go. Since all four ROMS are used, all software will work, the ROMS fit on the back of the board allowing full use of slot #7. One set of ROMS is powered up at a time.

DOUBLE DOS Plus requires APPLE DOS ROMS

\$39.00

201-839-3478 software



AGAIN! Ahead of all others.

- AUTO-LOAD PARAMETERS
- EXPANDED USER MANUAL
- TRACK/SECTOR EDITOR
- DISK DIAGNOSTICS
- PROGRAM MODIFICATION
- FAST & MORE ACCURATE

NIBBLES AWAY II

THE ULTIMATE NEW AND IMPROVED DISK BACKUP SYSTEM

NIBBLES AWAY II, second edition, is a greatly enhanced and improved version of our earlier product **NIBBLES AWAY**. Many new and exciting features have been implemented in **NIBBLES AWAY II** to insure State of the Art reliability. Other similar systems on the market today (regardless of price) can't begin to compare with all the features built into **NIBBLES AWAY II**.



Updates will be made available to all earlier versions through COMPUTER applications for a modest charge of Ten Dollars, to cover manual, Diskette and Postage & Handling include NAME and SERIAL # of ORIGINAL



ONLY \$69

Dealer Inquiries Invited.

3D Robot Tank

Fast Machine Language. 3D Hires Animation Arcade game with sound. Look out of your turret and try to blast the enemy tanks and saucers.

3D Animation adds to the effects and use of the game paddles or joystick give you command of your tank. ONLY \$29.95

Super PIX HIRES SCREEN DUMP

The Software package that will allow your printer to dump page 1 or page 2 of the Apple Hires screen horizontally or vertically. Use with EPSON® MX-80 with or without GRAFTRAX® Roms, MX-70 OKI® Microline 80, 82, 83, 82A, 83A - C ITOH® 8510 and NEC. Requires Tymac Parallel Printer Board PPC-100. MX-80 Version \$39 all others \$24.95

THE APPLE CARD—Two sided 100% plastic reference card Loaded with information of interest to all Apple owners \$3.98



MICRO-WARE DIST. INC.

P.O. BOX 113
POMPTON PLAINS, N.J.
07444

MICRO™

Software Catalog

Name: **Color Editor**
System: TRS-80C
Memory: 32K
Language: Assembly
Language

Description: *Color Editor* is used for program development and text processing. It allows use of the upper and lower case features of the Color Computer and can print letters or programs on a printer attached to the RS-232 port. It has change and search commands that work on one or all lines and can copy or move sentences or paragraphs anywhere in the file. Lines can be inserted, deleted, or moved. Your work can be saved and later retrieved on cassette.

Price: \$29.95 includes cassette and instructions

Available:
Computerware
P.O. Box 668
Encinitas, CA 92024
(714) 436-3512

Name: **The Merger
VisiBlend**
System: Apple II or Apple II Plus
Memory: 48K
Language: Applesoft in ROM
Hardware: Disk II

Description: *The Merger* is a utility that aids users of *The Data Factory* and *The Invoice Factory*. Merge data from fields in either program into those of another file. *VisiBlend* allows users of *VisiCalc™* to combine the data in multiple *VisiCalc™* files, merging data across files. (*VisiCalc* is a trademark of Personal Software, Inc.)

Price: \$50 each
Available:
Micro Lab
2310 Skokie Valley Rd.
Highland Park, IL 60035

Name: **PET Library Card
Maker**
System: PET
Memory: 16K
Language: BASIC
Hardware: IDS 460 printer
Description: Prints a full set of library cards on tear-off card stock. The information is

typed once to the PET screen. You can preview the card, make corrections, then with one keystroke, print a set of up to 7 library cards. The program does all the formatting. A tape file can be made. For all small libraries.

Price: \$80 - Canadian
Includes cassette tape
Author: J. Horemans

Available:
M&W Computer Stores
Sheridan Corporate Centre
2155 Leanne Blvd., Unit 3
Mississauga, Ontario
Canada L5K 2K8

Name: **Inventory System**
System: OS65U
Memory: 48K
Language: BASIC
Hardware: OSI C-2 or C-3 series

Description: *Inventory System* is an integrated portion of EIS General Accounting Systems. It has perpetual inventory, sales invoicing, accounts receivable, bills of materials, and interrelated purchase orders; information on availability, cost ordering of low or out-of-stock inventory items.

Price: \$1,200.00
Includes three program disks and a step-by-step user's manual.

Available:
Electronic Information
Systems, Inc.
P.O. Box 5893
Athens, GA 30604
(404) 353-2858

Name: **Turf Management**
System: OSI C4P MF
Language: BASIC under OSI 65D
Hardware: Disk drive, optional printer

Description: A program that provides the characteristics of eight common grass species, giving optimum growing conditions, use, techniques for establishment, lime and fertilizer requirements, and pest management and control for general insect and weed problems. Rates of seed required for establishment and amounts of

chemicals needed for each required application can be calculated given the dimensions of the area. The program can be customized to suit the area of the country and availability of chemicals used for fertilizer and pest control.

Price: \$100.00
Includes 5¼" disk and documentation ppd.
Modification to operate with other systems can be requested.

Author: J. Benton Jones, Jr.
Available:
Benton Laboratories, Inc.
P.O. Box 5455
Athens, GA 30604

Name: **SCORE: The
Academic
Assistant**
System: Apple II Plus (or Apple II with Applesoft on firmware card)

Memory: 48K
Language: Applesoft and machine
Hardware: 80-column printer; optical mark reader (Chatsworth, HEI or Scan-tron) highly recommended

Description: *SCORE* is a comprehensive set of programs which will score multiple choice tests, conduct comprehensive item analyses, maintain academic records, prepare frequency distributions, and individualized student feedback, and much more. This package interfaces the Apple with the Chatsworth, HEI, or Scan-tron optical mark readers.

Price: \$395.00
Includes program disk, backup disk, data disk, comprehensive manual, ongoing support

Author: Bryan Hendricks, and Bob Bermant
Available:
Scientific Software Assoc., Ltd.
P.O. Box 208
Wausau, WI 54401
(715) 845-2066

Name: **Napoleon's
Campaigns: 1813
& 1815**

System: Apple II
Memory: 48K
Language: Applesoft in ROM
Hardware: Disk Drive
Description: Corps-level game simulating the last campaigns of Napoleon: Leipzig and Waterloo. Displayed on 18 x 21 hex grid maps in hi-res graphics. Computer acts as corps commander.

Price: \$59.95
Includes one diskette, rulebook, player aid card, two-sided map boards, 100 counters.

Available:
Strategic Simulations, Inc.
465 Fairchild Drive
Suite 108
Mountain View, CA 94043

Name: **TRS-80 Color
Computer
Learning Lab
(26-3153)**

System: TRS-80 Color Computer
Memory: 4K, 16K, 32K
Language: Color BASIC
Hardware: Cassette

Description: A new self-instruction system that teaches user how to program in Color BASIC. Allows student to develop gradually through writing and editing longer, more complex programs. Example programs are practical and can be used for educational, family and personal purposes. The lab is divided into three sections: introduction to the computer; programming the computer; programming guides and tools to make programming easier, faster and more fun. The lessons take full advantage of color graphics and sound available from the TRS-80 Color Computer.

Price: \$49.95
Includes eight program cassettes and 30-lesson text
Author: Radio Shack
Available:
Radio Shack


(Continued on page 94)

FUNTASTIC!

Habit forming

SNACK ATTACK

Beche can't play just one game!



DATAMOST

A trademark of Apple Computer, Inc.

\$29.95. For Apple II*

"THIEF"

The most exciting game ever!



It's your quest for treasure, riches, a great level of achievement and excitement to become the galaxy's greatest Thief!

\$29.95 For Apple II*


DATAMOST
1927 North St.
Menlo Park, CA 94025

© 1983 L&L Software, Inc. All rights reserved.

COUNTY FAIR

IS COUNTY TERRIFIC

Pick up your gun and blow away in the most colorful, exciting shooting gallery in the world. This is the game everyone's crazy about.



\$29.95
For Apple II*

Casino

5 great casino games. The favorite of Monte Carlo, Vegas, Atlantic City. In your luck of the draw, Reno, roulette, Poker and Blackjack betting! You can bet on it!

\$39.95.
For Apple II*



DATAMOST
1927 North St.
Menlo Park, CA 94025

© 1983 L&L Software, Inc. All rights reserved.

DATAMOST
9748 Cozycroft Ave.,
Chatsworth, CA 91311
(213) 709-1202

Software Catalog (Continued)

Name: The Terrapin Logo Language

System: Apple II or Apple II Plus

Memory: 64K, 48K Apple with 16K memory extension

Language: Terrapin Logo Language

Hardware: 1 disk drive

Description: The Logo language is the most powerful interpretive language ever devised for the Apple II. Yet, it is probably the easiest to use as well. It is designed so that young children can easily control the power of computers without having to know how to program. However, advanced programmers will enjoy the many features common to artificial intelligence research languages permitting programs of great power to be written quickly and easily. (Language is licensed by Mass. Institute of Technology.)

Price: \$149.95

Includes language disk, utilities disk and documentation including tutorial and technical manual

Author: Leigh Klotz, Pat Sobalvarro, Steve Hain

Available:

Terrapin, Inc.
678 Massachusetts Ave.
Cambridge, MA 02139
(617) 492-8816

Name: Alkemstone™

System: Apple II, Apple II Plus, Apple III

Memory: 48K

Language: Machine

Hardware: DOS 3.3

Description: *Alkemstone*, is a computer adventure which offers a \$7500 cash reward to the first person who can recover the missing Alkemstone. The quest for the Alkemstone will lead the player through underground paths of the lair of the original owner. There are unusual messages, fragments of words, sketches and other clues written on the walls. Some items are distributed randomly, so that one may be visible numerous times while some will only appear once in several trips. Each trip will result in a different combination of possible hints. If all bits of information are pieced

together correctly, then the location of the Alkemstone will be obvious.

Price: \$39.95

Includes 1 disk, 24-page booklet

Author: Level-10

Available:

Level-10, a division of Dakin5 Corp.
7475 Dakin St.
Denver, CO 80221
or local Apple dealer

Name: Chem Lab Simulations #3 and #4

System: Apple II Atari 800

Memory: 48K

Language: Applesoft or Atari BASIC

Hardware: 48K Apple II with disk drive or 40K Atari 800 with disk drive

Description: High Technology Software Products, Inc., is pleased to announce the third and fourth additions to its series of chemistry laboratory simulations. *Chem Lab Simulations #3* contains four calorimetry experiments through which Hess' Law is demonstrated. *Chem Lab Simulations #4* utilizes two capillary tube experiments to illustrate principles of thermodynamics. Designed for college-level introductory chemistry courses, these simulations are also well suited for advanced high school students.

Price: \$100 each

Includes program diskette, 3-ring binder with complete documentation.

Author: J.I. Gelder

Available:

High Technology Software Products, Inc.
2201 N.E. 63rd St.
P.O. Box 14665
Oklahoma City, OK 73113
or computer retailers

Name: K-RAZY Shoot-Out

System: Atari 400/800

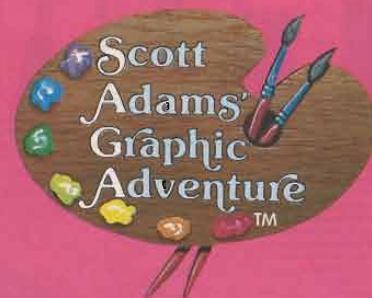
Memory: 8K ROM

Language: 6502 machine

Description: Fast action game.

(Continued on page 96)

Have you experienced a S.A.G.A. recently?



The Scott Adams Adventure series for the Apple II is now better than ever. The new Scott Adams Graphic Adventure series features voice, lowercase, and full color, high resolution graphics.

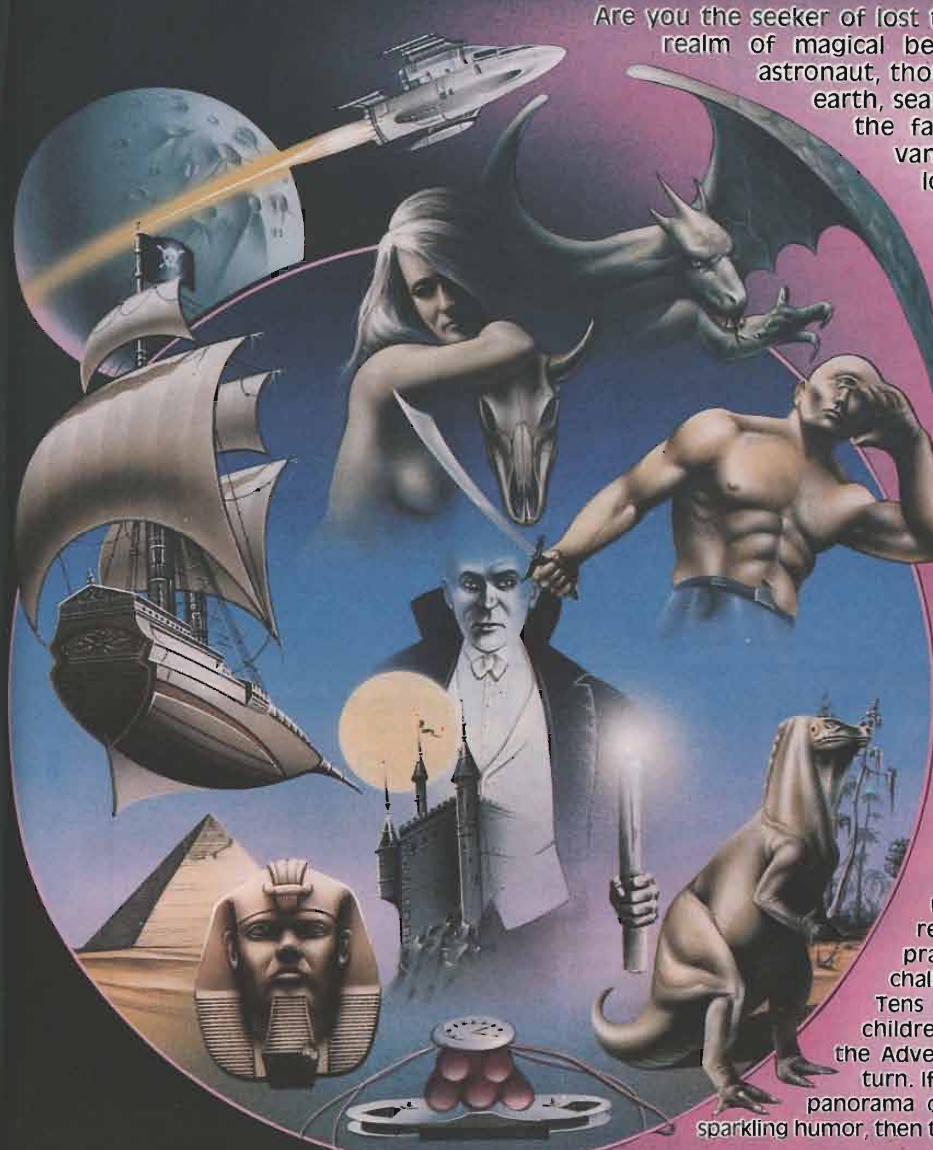
- ★ A jam-packed disk, with both sides used, contains the Adventure and the full color hi-res pictures which are stored using new, special state-of-the-art techniques.
- ★ Each screen was drawn using a palette of over 100 colors and textures. Each work of art was designed and entered by a professional artist.
- ★ Both the VOTRAX Type-N-Talk™ voice synthesizer and lower case hardware are supported.
- ★ Your printer is supported, too! With a single keystroke you can print your Adventure as you play to create a log of your journey.



The 12 Scott Adams Adventures

Adventureland • Pirate Adventure • Mission Impossible • Voodoo Castle • The Count • Strange Odyssey • Mystery Fun House • Pyramid of Doom • Ghost Town • Savage Island-Part 1 • Savage Island-Part 2 • Golden Voyage

WHO ARE YOU THIS TIME?



Are you the seeker of lost treasures in an enchanted realm of magical beings? Perhaps you're an astronaut, thousands of light years from earth, searching the galaxy's rim for the fabulous treasures and advanced technologies of a long-dead civilization. Maybe you're the plunderer of ancient pyramids in a maddeningly dangerous land of crumbling ruins and trackless desert wastes.

Imagine.

You wish to soar to other worlds, to behold wonders never beheld by mortal eyes, to dream unrestrained to the furthest limits of your imagination.

Imagine now that you can.

The ADVENTURE SERIES by Scott Adams has been reviewed by every major microcomputing magazine, receiving only the highest praise for its mind-puzzling challenge & refreshing originality. Tens of thousands of adults and children have matched wits with the Adventure Master. Now, it's your turn. If you crave the challenge and panorama of the exotic, touched with sparkling humor, then the Adventure Series is for you.

With the Incredible Adventure Series by Scott Adams
**YOU ARE WHOMEVER YOU WANT TO BE,
 WHENEVER YOU WANT TO BE!**

Adventure
 INTERNATIONAL
 A DIVISION OF SCOTT ADAMS, INC.

1982
 BOX 3435, LONGWOOD FL 32750
 (305) 862-6917 (QUESTIONS)
ORDER FROM YOUR FAVORITE DEALER
 or CALL TOLL FREE (800) 327-7172
 (ORDERS ONLY PLEASE)

	SINGLE TAPE	12 PACK TAPE	SINGLE DISK	TRIPLE DISK	12 PACK DISK	
APPLE	\$12.95 24K	\$129.95† 24K	\$29.95* 48K	\$69.95* 48K	—	Adventure Hint Book (All-12) . . . \$ 7.95
ATARI	\$19.95 24K	\$129.95† 24K	—	\$39.95 48K	\$129.95† 48K	Adventure Hint Sheet (Single) . . . 1.00
CP/M (Z-80)	—	—	—	—	\$129.95†	Commodore VIC (Cartidge) . . . 39.95
TRS-80 MOD 1&3	\$19.95 16K	\$129.95† 16K	—	\$39.95 32K	\$129.95† 32K	T-1-9914 (Tape or Disk) . . . 29.95
TRS-80 MOD 2	—	—	—	—	\$129.95	Requires Adventure Command Module

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

*Scott Adams' Graphic Adventure Series Apple Disks: Full Color Hi-Res Graphics: T12 Adventure Hint Book Incl.

SHIPPING & HANDLING ARE EXTRA

Software Catalog *(Continued)*

You're trapped in an alien fortress and you must clear your sector of alien droids and proceed to the next sector. Each sector is more challenging and demanding with faster and more aggressive droids. Available in solid state ROM pack.

Price: \$49.95

Includes ROM pack and multi-color 12 page instruction booklet.

Author: K-BYTE

Available:

K-Byte
1705 Austin
Troy, Michigan 48099
or your dealer

Name: **Debug**
System: OSI C1P/MF,
C4/MF

Memory: No Additional

Language: Machine

Description: This machine-code program is used with OSI's Extended Monitor to provide single stepping of your computer to trace machine-code programs one instruction at a time. As you step through the program the display will

show the mnemonic instruction before it is executed in the same form as the disassembler shows it. You can set high and low trace limits; this lets the program execute without displaying the instructions until the address is within these limits. The third feature is the ability to set a breakpoint. This will let you stop execution at any address in memory. This breakpoint, and tracing, can be done in ROM.

Price: \$12.95

Includes 5 1/4" disk, documentation.

Author: Dave Pompea

Available:

DMP Systems
319 Hampton Blvd.
Rochester, N.Y. 14612

Name: **UTIL1**
System: AIM 65
Memory: 16K
Language: AIM Assembler
Hardware: Standard AIM
Description: *UTIL1* is a 2K extension of the AIM 65 monitor. It interfaces to AIM via the user I/O ports and the user

function key 3. It adds 18 commands to AIM. Eight of these are associated with a Buffer Manager that gives AIM a virtual I/O capability. Up to 8 I/O devices are emulated in RAM. This gives the AIM editor move and copy capability. These I/O devices can be used with any AIM firmware or software that uses the AIM Active Output Device [AOD] and the Active Input Device [AID]. An additional 10 commands provide utilities such as memory display, search and move, and an offset loader for AIM object files.

Price: \$25 object on cassette

\$5 16-page manual

\$25 commented assembly listing

Includes object assembled to your specified address.

Should reside at top of RAM. Specify Address.

Author: Joel Swank

Available:

Nehalem Bay Software
P.O. Box 2006
Beaverton, OR 97075

Name: **Creature of the Maze**

System: Ohio Scientific

Memory: 8K

Language: BASIC-in-ROM

Hardware: Challenger C1P or Superboard Series I or II

Description: Incredibly realistic 3-D graphics lock you into combat with the "Creature of the Maze." Each game starts with a new and different maze, created and displayed for you to ponder, but only for a short moment. Then the screen clears and you find yourself looking down long corridors, peeking around corners and searching for your enemy. The hallways explode with your lazer blasts as the message across the screen spells out, "The monster is near." Tremendous fun with ten skill levels and hundreds of maze sizes to choose from.

Price: \$14.95

Includes cassette, user's manual with objectives, options, and suggestions for modification

Author: John H. DeRosa

Available:

Dee Products
150 Birchwood Road
Lake Marion, IL 60110

MICRO

**MORE
MORE
MORE
MORE
FUN!
FUN!**

6 Ports & a Joy of a stick

... to expand your
Apple's* fun & convenience.

Our exclusive EXPANDAPORT gives outside-of-the-Apple convenience! No more opening the cover. Just install EXPANDAPORT...and plug in joy sticks, printer, light pen etc. Comes complete with speaker for greater sound. Only **\$69.95**.

This JOY STICK's a joy to use. Its heavier metal case doesn't slip or slide like plastic. And the very sensitive switch's guaranteed for 1,000,000 pushes. Better control and reliability, only **\$59.95**.

At computer stores, or:
DATAMOST
9748 Cozycroft Ave.
Chatsworth, Ca 91311
(213) 709-1202

VISA/MASTERCARD accepted. 3% shipping/handling charge. (California residents add 6% tax)
*Apple is a trademark of Apple Computer, Inc.



SoftSide™

CAN'T AFFORD SOFTWARE? READ THIS:

If the high price of commercial software and the lack of clear information about your microcomputer has got you down, here's the solution you've been waiting for!

SoftSide Magazine

SoftSide is a favorite of computer users and hobbyists alike. They rely on it as a prime source of programs, reviews and articles for the Apple™, ATARI®, and TRS-80® microcomputers.

SoftSide is the magazine for the microcomputer owner who wants to learn BASIC programming, learn MORE about BASIC programming, or just wants to have FUN!

SoftSide gives you the BASIC code listings of several programs — adventures, utilities, games, simulations, you name it — for your computer EVERY MONTH.

There's more:

- **Reviews** — of the software and hardware products you want to know about.
- **Articles** — about all aspects of BASIC programming.
- **Tutorials** — on graphics, use of important commands, and more.
- **Programs** — each month SoftSide publishes a variety of program for the Apple, ATARI® and TRS-80®.

- **Columns** — which discuss different topics including: computer graphics, picking the right modem for you and marketing your software — just to name a few.
- **Input from our readers** — each month we devote a space in the magazine to let our readers give us some feedback about SoftSide.
- **Hints & Enhancements** — programmers and readers provide us with enhancements to our programs, and programming tips.

As you can see, you'll receive pages and pages of information and entertainment from SoftSide. But here's the best part:

A subscription to SoftSide is just \$24 a year. That's 12 issues for only \$2 each! What a value!

YES! Send me the first copy of my SoftSide subscription right away!

\$24/yr for USA and Canada only. For orders to APO/FPO or Mexico — \$40/yr. Other foreign orders — \$62/yr.

I own a Apple ATARI® TRS-80®

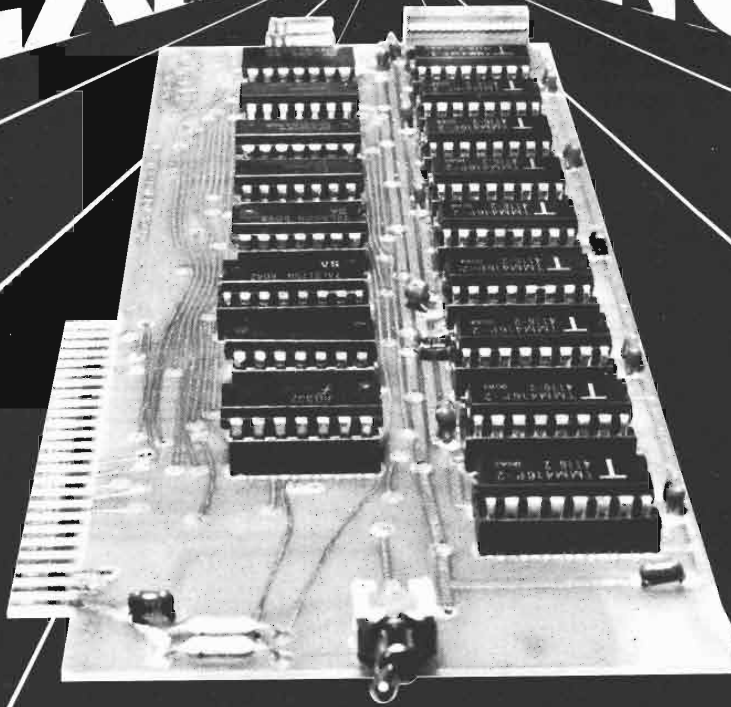
Name _____
Address _____
City/State _____ Zip _____

Check is enclosed
 MasterCard VISA

Name of Cardholder _____
MC# and Interbank#/VISA# _____
Exp. Date _____
Signature _____

**NEW LOW PRICE
DIRECT FROM MANUFACTURER
\$120.00**

EXPAND * ENHANCE



16K RAM EXPANSION BOARD FOR THE APPLE II* \$120.00

The Andromeda 16K RAM Expansion Board allows your Apple to use RAM memory in place of the BASIC Language ROMs giving you up to 64K of programmable memory. Separate Applesoft* or Integer BASIC ROM cards are no longer needed. The 16K RAM Expansion Board works with the Microsoft Z-80 card, Visicalc, DOS 3-3, Pascal, Fortran, Pilot, and other software. A switch on the card selects either the RAM language or the mainboard ROMs when you reset your Apple.

The Andromeda 16K RAM Expansion Board has a proven record for reliability with thousands of satisfied customers.

Now with One Year Warranty.

ANDROMEDA



INCORPORATED
Greensboro, NC. 27410
P.O. Box 19144

919 852-1482



Price for Andromeda 16K RAM expansion board now only \$120.00. Please add \$5 for shipping and handling. North Carolina residents add 4% sales tax.

*DEALER INQUIRIES WELCOME.

6809 Bibliography

18. Commodore Interface (July, 1981)

Anon., "Super PET," pg. 18-20.
Questions and answers on the new Super PET based on a 6502 plus 6809.

19. MICRO No. 41 (October, 1981)

Puckett, Dale L., "The 6809 and the S-50 Bus," pg. 68-73.
The 6809 is much easier to work with than the 6800, and the programs are about 30 percent shorter and run about 30-40 percent faster.

Steiner, John, "The Radio Shack Color Computer: A 6809-based System," pg. 9-10.
The TRS-80 Color Computer is one of the most popular and versatile 6809-based systems to date. An outline of the features is given.

20. Apple Assembly Line 2, Issue 1 (October, 1981)

Wiggs, Chris and Sander-Cederlof, Robert, "6809 Cross Assembler," pg. 12.
Patches for the S-C Assembler Version 4.0 are available to give a brand new assembler for the 6809.

21. MICRO No. 42 (November, 1981)

Steiner, John, "Lunar Lander," pg. 41-44.
Animated graphics in BASIC for the 6809-based TRS-80 Color Computer.
Capouch, Brian, "OS-9 and the 6809: Revolutionary Tools," pg. 81-86.
The Microware OS-9 operating system is an advanced software package for the 6809. Also described is BASIC09, a high-level programming system (alternative to a BASIC interpreter or compiler). Illustrative listings of a pair of BASIC09 procedures in source code are given.

22. FWAUG Newsletter 2, No. 6 (October, 1981)

Hardenburg, Hal and McVay, Ray, "Concerning the 6502 and 6800," pg. 25-26.
Notes comparing the 6502, 6809 and the 68000 microprocessors, including a report that the Apple in the future may use the new 68000 microprocessor.

23. BYTE 6, No. 11 (November, 1981)

Walker, Gregory, "Toward a Structured 6809 Assembly Language, Part 1," pg. 370-382.
An introduction to structured assembly language for the 6809.

24. Stems From Apple 4, Issue 9 (September, 1981)

Hardenbergh, Hal W., "To Persons Interested in Both the 6502 and the 6800," pg. 5-14.
A comparison of the 6502, 6809 and the 68000 microprocessors, including an assembly language program for the 68000 multiply routine.

25. Compute! 3, No. 10, Issue 18 (November, 1981)

MacLean, Bill, "SuperPET: A Preview," pg. 38-40.
A rundown on the SuperPET micro which incorporates both the 6502 and the 6809.

26. BYTE 6, No. 12 (December, 1981)

Barden, William, "Color Computer from A to D," pg. 134-160.
A detailed look at the Radio Shack TRS-80 Color Computer, based on the 6809 microprocessor.
Walker, Gregory, "Toward a Structured 6809 Assembly Language," pg. 198-228.
Part 2 discusses implementing a structured assembler.

27. KB Microcomputing 5, No. 12, Issue No. 160 (December, 1981)

Stark, Peter A., "68XX Secrets," pg. 116-130.
A review of Dynamite, a good disassembler running on a 6809 FLEX 9 disk operating system. Notes on building a 6809 48K system.

28. Compute! 3, No. 12 (December, 1981)

Anon., "A Look at SuperPET," pg. 130-132.
Features of the CBM SuperPET and several useful utilities for this 6809-based system.

29. Compute! 4, No. 1, Issue 20 (January, 1982)

Mansfield, Richard, "BRANCH NEVER and QUIF Assembling on SuperPET," pg. 146-149.
Discussion of using some of the special 6809 statements available when assembling on the SuperPET.

30. MICRO No. 44 (January, 1982)

Tenny, Ralph, "Experimenters and the Color Computer," pg. 18-22.
A summary of the normal capabilities of the TRS-80 Color Computer and an examination of the unit's I/O capability. Also information on hardware for I/O use.

31. Apple Assembly Line 2, Issue 3 (December, 1981)

Sander-Cederlof, Bob, "EXCEL-9: A 6809 Card with FLEX," pg. 1.
A board with a 6809E CPU, 8K of ROM and an interval timer with built-in linkage routines for calling 6809 routines from Applesoft, Integer BASIC, or from 6502 machine language.

32. KB Microcomputing 6, No. 1, Issue 61 (January, 1982)

Wolf, Michael A., "Changing Chips in Midstream," pg. 96-100.
Discussion of the use of the 6809 microprocessor in the Radio Shack Color Computer.

33. KB Microcomputing 6, No. 2, Issue 62 (February, 1982)

Stark, Peter A., "6800 Secrets," pg. 84-98.
More bench tests on various microprocessor-equipped systems. Includes several related to the 6809 chip.

34. MICRO No. 45 (February, 1982)

Garrett, Leo E., "Utilities for the Color Computer," pg. 9-15.
A versatile routine allowing TRS-80 Color Computer users to dump or disassemble the 6809 or ASCII code in any section of memory, including the BASIC or expansion ROMs.
Staff, "MICRO Software Catalog," pg. 117-121.
Includes items of software for 6809 systems.

6502 Bibliography

1. MICRO No. 43 (December, 1981)

Traeger, John C., "Data Collection with Your Micro," pg. 9-11.

How to construct and implement an interface which enables high-speed sampling and recording of experimental data. Written for an AIM 65, it is readily adapted to any 6502 microprocessor with either a 6502 or 6222 interface adapter.

2. PEEK(65) 2, No. 12 (December, 1981)

Cook, William H., "Add an 8-Inch Floppy to the C2-4F/C4P," pg. 2-4.

A hardware article for OSI users. Modifications for the 502 CPU board and detailed information on the interconnections of the 470 Floppy Disk Controller board with the disk drive are given. Thirteen signal lines are run to the Siemens FDD 400-8 drive.

3. BYTE 6, No. 12 (December, 1981)

Jacobs, Jacob R., "Generating Programs Automatically," pg. 352-362.

Let your Apple II do the programming. Three programs are written in Applesoft BASIC. These utilities help set up your desired program with data entry, data output, instructions, etc. Sample dialogs in running the program are given in the article.

4. BYTE 6, No. 12 (December, 1981)

Kopp, Gregory L., "Discovering Atari's 'Hidden' Graphics," pg. 98-102.

Improper graphics commands on the Atari often leads to unexpected results. Some of these undocumented commands may be used to advantage. A chart of useful 'hidden' commands are given and example listings demonstrate the effect.

5. Apple-Dayton 2, No. 12 (December, 1981)

Brungart, David L., "Organizing Applesoft," pg. 19-25.

Listing and discussion of a program utility package used to set up temporary utility routines to ease the task of writing Applesoft programs in an orderly manner.

6. KB Microcomputing 5, No. 12, Issue No. 60 (December, 1981)

Young, John E., "Poor Man's Memory Expansion for the OSI," pg. 56-60.

An inexpensive way to expand the memory of the Superboard II or Challenger C1P. Instead of using a \$300 OSI 610 expansion board, a method to implement a \$30 16K static RAM board is described.

7. Call —A.P.P.L.E. 4, No. 9 (November/December, 1981)

Anon., "Puffin," pg. 13-42.

A DOS to Pascal File Converter for the Apple. A menu of four commands is presented: Catalog, Display, Transfer and Quit. Earlier a program called Huffin to convert Pascal files back to DOS was published (Call —A.P.P.L.E. Oct., 1982).

8. Softalk 2, No. 4 (December, 1981)

Coats, Douglas E. and Waldman, Cy H., "FORTRAN," pg. 160-172.

Comparisons of Apple FORTRAN and Microsoft FORTRAN for the Apple. Includes versions using "The Mill" and the Softcard accessories for the Apple.

9. Nibble 2, No. 8 (December, 1981)

Exner, Chris; Guy, Rudy; and Harvery, Mike, "Trend Reporting, Analysis, and Control," pg. 7-29.

A group of three extensions to the Apple-based TRAC system. Budget TRAC allows you to be aware of where your money goes, TRAC Spending Graph will graph the data in hi-res, and TRAC Plus shows how to get the best use of the system.

10. Creative Computing 7, No. 12 (December, 1981)

Brewster, Keith, "Who's Afraid of the Big Bad Matrix?," pg. 168-173.

Arrays and Matrix operations on the Atari are discussed and illustrated with numerous listings.

"YOUR SALVATION IN THE SEA OF INFLATION"

ARK COMPUTING

80 Column Card & 16K RAM Board Sale

The best price available on a 80-Column for your APPLE Wizard-80 by Wesper-Micro
RP \$345.00 ARK \$225.00

16K Ram Board ARK \$ 89.00
32K Ram Board ARK \$139.95
- with 16-K ram installed -
(add \$ 20.00 for full 32-K ram installed)

LOWER CASE PLUS II by Lazer Microsystems.
The Lazer MicroSystems LCP II is the LCA value for the budget minded APPLE II owner. Works with all Revision 7 and Later APPLE II's. Includes Basic and Pascal software on disk. Works with many popular word processors.

ARK's LOW PRICE of \$ 19.95

For people who demand the best

The best Lower Case Adapter available for the APPLE II. LOWER CASE PLUS by Lazer Microsystems. This feature packed board has twice the features of competing boards. The only LCA that works with VISICALC and is recommended by Stoneware for DB MASTER.

FEATURES:
-Expandable to 4 character sets (2 on board)
-Inverse only mode
-Includes graphics & ASCII
-Compatible w/most word processors available.

AND WITH ARK's LOW PRICE THERE'S NO REASON TO PURCHASE ANY OTHER.

RP \$ 64.95 ARK \$44.95

Remarkable value in Keyboard Enhancers. The Lazer MicroSystems Keyboard +Plus with these features:
-64 character type ahead buffer
-Buffer can be cleared & disabled
-Works on all APPLE II's
-Lets you use the shift-key as a typewriter shift-key, allowing you to enter directly the 128 ASCII character set from the APPLE keyboard.

KEYBOARD +PLUS
RP\$ 99.95 ARK \$ 69.95

THREE EXCITING PRODUCTS. A ONCE IN A LIFETIME DEAL. LOWER CASE PLUS+, KEYBOARD PLUS+, & AN 80-COLUMN BOARD ALL FOR LESS THAN THE NORMAL COST OF THE 80-COLUMN BOARD ALONE.

ARK \$299.95

software

By LAZER

- ANIX 1.0: A set of incredible disk utilities w/ UNIX like Operating System.
ARK \$ 34.95
- LAZER PASCAL: A unique systems programming language with many features of the 'C' programming language.
ARK \$ 29.95
- DISASM/65: Most popular disassembler for the APPLE II.
ARK \$ 19.95

By DATAMOST

- SNACK ATTACK: RP\$ 29.95 ARK\$ 19.95
- THEIF: RP\$ 29.95 ARK\$ 19.95
- CONEY ISLAND: RP\$ 29.95 ARK\$ 19.95
- 6502 ASSEMBLY by RANDY HYDE:
RP\$ 19.95 ARK\$ 15.00

By BRODERBUND

- MIDNIGHT MAGIC: RP\$ 34.95 ARK\$ 24.95
(Best Pinball game we've seen yet)
- APPLE PANIC : RP\$ 29.95 ARK\$ 19.95
- SPACE QUARK : RP\$ 29.95 ARK\$ 19.95
- RED ALERT : RP\$ 29.95 ARK\$ 19.95
- TRACK ATTACK : RP\$ 29.95 ARK\$ 19.95
- GENETIC DRIFT : RP\$ 29.95 ARK\$ 19.95
- SPACE WARRIOR : RP\$ 24.95 ARK\$ 17.95

By ON-LINE SYSTEMS

- MOUSKATTACK : RP\$ 34.95 ARK\$ 24.95
- CROSS FIRE : RP\$ 29.95 ARK\$ 21.95
- CRANSTON MANOR: RP\$ 34.95 ARK\$ 24.95
- ULYSSES : RP\$ 34.95 ARK\$ 24.95
- SOFTPORN ADV : RP\$ 29.95 ARK\$ 21.95
- TIME ZONE : RP\$ 99.95 ARK\$ 75.95
- LISA 2.5 : RP\$ 79.95 ARK\$ 59.95
- SPEED/ASM : RP\$ 39.95 ARK\$ 29.95
- THRESHOLD : RP\$ 39.95 ARK\$ 29.95
- PEGASUS II : RP\$ 29.95 ARK\$ 21.95

We accept: MASTERCARD, VISA (Include card # and expiration date), CASHIER or CERTIFIED CHECKS, MONEY ORDERS, or PERSONAL CHECKS (Please allow 10 days to clear). Please add 3% for shipping & handling (minimum \$2.00). Foreign orders please add 10% for shipping & handling. We accept COD's (Please include \$2.00 COD charge) California residents add 6% sales tax. All equipment is subject to price change and availability without notice. All equipment is new & complete with manufacturer's warranty.

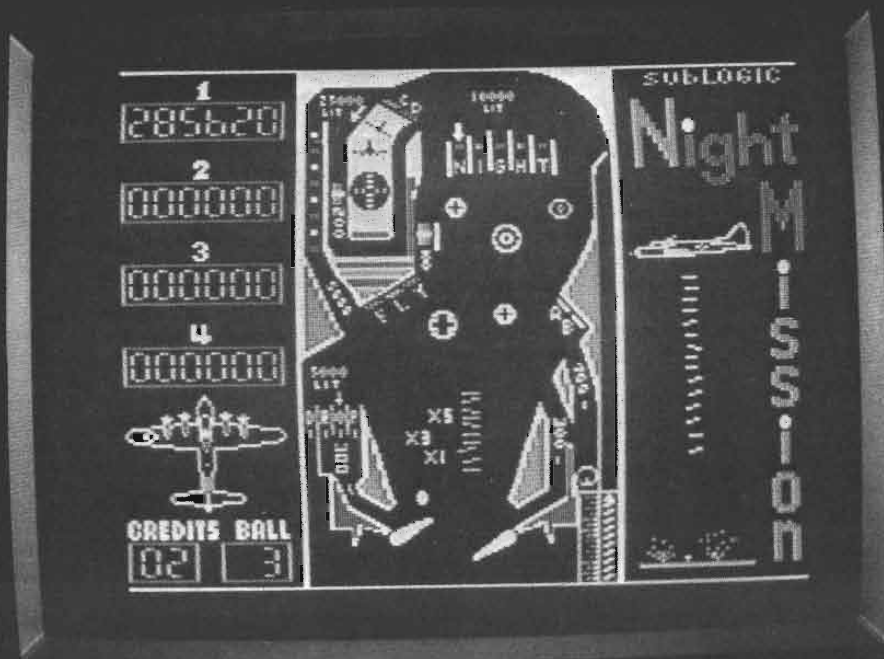
Call us at (714) 735-2250

ARK COMPUTING • P.O. Box 2025 • Corona, CA 91720

Introducing...

A2-PB1

PINBALL



This beautiful and sophisticated simulation offers superior ball action, superior flipper control, and many subtle perfections which will result in your superior enjoyment.

Please don't compare A2-PB1 Pinball with other pinball simulations. Nothing else even comes close. \$29.95

The engineering and graphics experts opening a new era in computer simulation.

subLOGIC

Communications Corp.
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex: 206995

See your dealer . . .

or for direct orders, add \$1.50 and specify UPS or first class mail.
Illinois residents add 5% sales tax.
Visa and MasterCard accepted.

Annual Index

June 1981 — May 1982

(Issues 37 - 48)

Articles

APPLE		
Musical Duets on the Apple II <i>Rick Brown</i>	37:11	42:89
Macros for Micros <i>John Figueras</i>	37:45	42:94
Create a Data Disk for DOS 3.2 and 3.2.1 <i>Glenn R. Sogge</i>	37:49	42:100
Apple Color Filter <i>Stephen R. Berggren</i>	37:53	42:105
Serial Line Printer for the Apple <i>Wes Huntress</i>	37:59	43:85
Integer Flash for the Apple <i>Richard C. Vile, Jr.</i>	37:83	43:96
Function Input Routine for Applesoft <i>Roy E. Myers and William Miller III</i>	38:13	43:101
Phone Search <i>Horst K. Schnelder</i>	38:21	43:106
Double Barrelled Disassembler <i>David L. Rosenberg</i>	38:33	44:25
Single-Drive Disk Back-Ups for Apple <i>Steve Emmett</i>	38:37	44:75
Enhanced Input Routine <i>Bruce A. Robertson</i>	38:41	44:95
Binary File Parameter List <i>Clyde R. Camp</i>	38:45	44:100
Electric Typing Program for the Apple <i>Thomas D. Brock</i>	38:61	45:59
Common Array Names in Applesoft II <i>Steve Cochard</i>	39:77	45:67
The Extended Parser for the Apple <i>Paul R. Wilson</i>	39:81	45:71
SEARCH <i>R.C. Merten</i>	39:83	47:87
Applesoft Error Messages from Machine Language <i>Steve Cochard</i>	39:86	47:97
Trick DOS <i>Sanford M. Mossberg</i>	39:88	48:9
Sorting with Applesoft <i>Norman P. Herzberg</i>	39:92	48:37
SHAPER: A Utility Program for Managing Shape Tables <i>Clement D. Osborne</i>	40:50	48:48
List Controller <i>Preston R. Black</i>	40:59	
Lo-Res Graphics and Pascal <i>C. Donald Heth</i>	40:62	
Dollars and Sense Revisited <i>David Dell'Quadri</i>	40:66	
Paddle Hi-Res Graphics <i>Kim G. Woodward</i>	40:68	
True 3-D Images on Apple II <i>Art Radcliffe</i>	40:71	
Apple Bits, Part 1 <i>Richard C. Vile, Jr.</i>	40:75	
Tracemark: An Apple II Debugging Aid <i>Raymond Weisling</i>	40:79	
Applesoft Mystery Parameter <i>Sherm Ostrowski</i>	41:12	
Taming the Wild Reset <i>Michael M. Seiy</i>	41:83	
Apple Byte Table <i>Kim G. Woodward</i>	41:88	
Apple Bits, Part 2 <i>Richard C. Vile, Jr.</i>	41:94	
Solar System Simulation, Part 2 <i>Dave Partyka</i>	41:108	
Galacti-Cube <i>Bob Bishop</i>	42:47	
Othello <i>Charles F. Taylor, Jr.</i>	42:63	
Apple II Digital Storage Oscilloscope <i>Ellis Cooper</i>		42:89
Function Generator and Library Manager <i>Ray Cadmus</i>		42:94
ASCII Dump for the Apple <i>Robert F. Zant</i>		42:100
Apple Bits, Part 3 <i>Richard C. Vile, Jr.</i>		42:105
Applesoft Variable Lister <i>Richard Albright</i>		43:85
Applesoft Memory Map Display <i>N.D. Greene</i>		43:96
Applesoft Line Finder Routine <i>Peter J.G. Meyer</i>		43:101
Applesoft and Matrices <i>Cornelis Bongers</i>		43:106
Sweet-16 Revisited <i>Charles F. Taylor, Jr.</i>		44:25
A Disk Menu Program <i>David C. Oshel</i>		44:75
RELOC <i>Robert D. Walker</i>		44:95
Apple Pascal Textfile Lister <i>Robert D. Walker</i>		44:100
Credit Box Creator <i>Sandy Mossberg</i>		45:59
RUNZMENU <i>Frank Shyjka</i>		45:67
Shape Manipulate <i>John R. Raines</i>		45:71
Applesoft Array Dump <i>Phillipe Francois</i>		47:87
Integer Cross Reference Utilities <i>Lee Reynolds</i>		47:97
Memory Map Relocator <i>Preston R. Black</i>		48:9
LISZT with Strings <i>Leonard H. Anderson, Donald Cohen, Richard F. Searle</i>		48:37
Apple Graphics for Okidata Microline 80 <i>Charles F. Taylor, Jr.</i>		48:48
OSI		
A C1P Dump Utility <i>Francois Faguy</i>		37:27
Memory Expansion for the Superboard <i>Fred Boness</i>		37:79
Polled Keyboard for C1P/Superboard <i>Michael J. Alport</i>		37:88
Real Time Clock for Superboard <i>James Mason</i>		37:99
A Typewriter Bell for Your Microcomputer <i>Charles L. Sanford</i>		38:65
Monobyte Checksum Dumper for C1P <i>Peter D. H. Broers</i>		38:67
Line Editor for OSI 540 Board <i>E.D. Morris, Jr.</i>		38:72
LIFE in a Wrap-Around Universe <i>Pauli Krieger</i>		38:75
STEP and TRACE for C1P <i>M. Pilot</i>		38:79
MICROCRUNCH: An Ultra-Fast Arithmetic Computing System Part 1 <i>John E. Hart</i>		39:7
A \$200 Printer for C1P and Superboard <i>Louis A. Beer</i>		39:40
C1P to Epson MX-80 Printer Interface <i>Gary E. Wolf</i>		39:42

(Continued on next page)

Expanding the Superboard <i>Jack McDonald</i>	39:97	PET Listener <i>Louis F. Sander, Victor H. Pitre</i>	43:14
The Disk Switch <i>Nelson E. Ingersoll</i>	40:15	KEYSORT for BASIC 4.0 <i>Gordon Campbell</i>	44:75
MICROCRUNCH: An Ultra-fast Arithmetic Computing System Part 2 <i>John E. Hart</i>	40:83	Auto-Run Wedge for the PET <i>Werner Kolbe</i>	44:76
Auto Line Numbers for OSI Disk BASIC <i>Lester Cain</i>	42:23	PET Audible Disk Alarm <i>John E. Girard</i>	46:107
OSI Symbolic Disassembler <i>David E. Pitts</i>	43:53	A Real Tape Operating System <i>Dale DePriest</i>	47:16
List Scroller <i>Colin Macauley</i>	44:57	7SEG PET Giant Character Set <i>John Girard</i>	47:83
Superboard Expansion System <i>D.W. Kammer</i>	45:31	PET Memory Protector <i>Louis F. Sander and Victor H. Pitre</i>	48:63
Speedy Routine for C1P Joystick <i>John Krout</i>	45:43	Growing Knowledge Trees <i>David Heise</i>	48:67
A Cross Reference Generator for OSI BASIC <i>John Krout</i>	46:37	PET Menu and Tape Timer <i>Dale De Priest</i>	48:84
More Hooks Into OSI BASIC <i>Richard L. Tretheway</i>	46:43		
Microsoft BASIC-In-ROM Extensions <i>Michael M. Mahoney</i>	46:51		
Machine Language to DATA Statement Generator <i>Yasuo Morishita</i>	46:61	KIM	
Autonumber Plus for Cursor Control <i>Kerry Lourash</i>	46:67	Improved KIM Communication Capabilities <i>Ralph Tenny</i>	37:65
General Purpose Tape I/O for OSI <i>Jerry D. Boucher</i>	47:11	Step Up to Programmable Motion <i>David S. Liscinsky</i>	41:29
A Single Drive Copier In BASIC <i>Peter Kleijnjan</i>	47:21	Some Help for KIM, Part 1 <i>Wayne D. Smith</i>	42:27
Programmable Reverse Video for the C1P <i>Charles L. Stanford</i>	47:39	Some Help for KIM, Part 2 <i>Wayne D. Smith</i>	43:49
		Some Help for KIM, Part 3 <i>Wayne D. Smith</i>	44:69
		KIM Bouncy Keypad Cure <i>Jody Nelis</i>	46:77
AIM			
Telephone Directory/Dialer for the AIM <i>Rodney A. Kreuter</i>	37:35		
AIM 65 RS-232 Interface <i>James Gullbeau</i>	37:97	GENERAL 6502	
AIM Memory Maps <i>Greg Paris</i>	38:7	Machine Language to DATA Statement Conversion <i>Les Cain</i>	37:33
Bit Pad Routines for AIM 65 <i>Ralph O. Ericson</i>	38:87	Vector Calculations with a Microcomputer <i>Peter Koski</i>	38:15
Disassembling to Memory with AIM 65 <i>Larry P. Gonzalez</i>	39:25	Expressions Revealed, Part 1 <i>Richard C. Vile, Jr.</i>	38:49
Using a TTY Printer with the AIM 65 <i>Larry P. Gonzalez</i>	39:36	Improved <i>n</i> th Precision <i>Glenn R. Sogge</i>	39:25
ASMLST: Full-sized Assembler Listing for AIM <i>Joel Swank</i>	40:37	Sorting <i>William R. Reese</i>	39:29
Interfacing Two 12-Bit A/D Converters to an AIM <i>G. Roger Heal and J. Derek Openshaw</i>	41:100	An Inexpensive Printer for Your Computer <i>Michael J. Keryan</i>	39:57
Formatting AIM Assembler Listings <i>Christopher J. Flynn</i>	45:19	Expressions Revealed, Part 2 <i>Richard C. Vile, Jr.</i>	39:66
I/O Expansion for AIM <i>Gary Finley</i>	46:111	Jumps and the 6502 <i>Mark Bernstein</i>	40:8
AIM User Device Arbiter <i>Joel Swank</i>	47:6	6522-Based Pot Position Digitizer <i>Kenny Winograd</i>	41:18
BASIC to Machine Language Interface <i>Christer Engstrom</i>	48:13	6502 Frequency Counter <i>Phil Lindquist</i>	41:24
		Handling Analog Signals with a Micro <i>Richard Saltero and Arthur Poulos</i>	41:76
		Precision Programming <i>Al Hamilton</i>	42:6
		Pascal Tutorial, Part 1 <i>Victor Fricke</i>	42:13
PET		Data Collection with Your Micro <i>John C. Traeger</i>	43:9
Programmable Character Generator for the CBM 2022 Printer <i>Roger C. Crites</i>	37:11	Watch that Ground Connection <i>Raymond Weisling</i>	43:21
Horizontal Screen Scrolling on the CBM/PET <i>John Girard</i>	37:81	Pascal Tutorial, Part 2 <i>Victor R. Fricke</i>	43:57
PET Interface to Bit Pad <i>Peter Coyle</i>	38:83	Flags and Boolean Algebra in Microsoft BASICs <i>M. Guzdial</i>	43:64
PET/CBM IEEE 448 to Parallel Printer Interface <i>Alan Hawthorne</i>	39:53	Recursive Use of GOSUB in Microsoft BASIC <i>R.B. Johannesen</i>	43:68
Commodore ROM Genealogy <i>Albert I. Reuss</i>	41:50	Pascal Tutorial, Part 3 <i>Victor R. Fricke</i>	44:85
VIC Light Pen-manship <i>David Malmberg</i>	41:54	Elementary Pascal Internals <i>Arnle Lee</i>	44:103
The PET from A to D <i>John Sherburne</i>	41:60	A FORTHword <i>Loren Wright</i>	45:83
Substitute Characters <i>Loren Wright</i>	41:64	Using FORTH with the 6502 <i>Raymond Weisling</i>	45:85
Ultimate Ping Pong for PET <i>Werner Kolbe</i>	42:67		

Stepper Motor Control: A FORTH Approach <i>Mark Bernstein</i>	45:95	OS-9 and the 6809: Revolutionary Tools <i>Brian Capouch</i>	42:81
LIFE in FORTH and BASIC <i>Nicholas J. Vrtls</i>	45:103	Experimenters and the Color Computer <i>Ralph Tenny</i>	44:18
VisiCalc Formulas for Depreciation <i>Kim G. Woodward</i>	46:9	Utilities for the Color Computer <i>Leo E. Garrett</i>	45:9
Numerical Solution of Differential Equations <i>Robert D. Waiker</i>	46:13	A Disassembler for the 6809 <i>Mark T. Borgerson</i>	46:89
Le Grange Interpolating Polynomial <i>Paul H. Muller</i>	46:21	Structured Programming in BASIC09 <i>Brian Capouch</i>	47:45
SIN(X) The Hard Way <i>Earl Morris</i>	46:24	Extensions to the CBUG Monitor <i>Ralph Tenny</i>	47:51
Binary Storage and Array Retrieval <i>Hank Blakely</i>	46:83	Multiprecision Addition — A Comparison of 6809 and 6502 Programming <i>Gregory Walker and Tom Whiteside</i>	47:57
An Enhanced 6502 <i>Ralph Tenny</i>	47:31	Memory Moves with the 6502 and the MC6809 <i>Gregory Walker and Tom Whiteside</i>	48:19
Time of Day The Easy Way <i>Martin DeGeorge</i>	47:35		
6809			
It's Time to Stop Dreaming, Part 1 <i>Robert M. Tripp</i>	37:3	Atari Journal It: Screen Print Utility for Atari <i>John Elliott</i>	40:27
It's Time to Stop Dreaming, Part 2 <i>Robert M. Tripp</i>	38:27	Proportional Joystick for Atari <i>Mike Dougherty</i>	43:27
It's Time to Stop Dreaming, Part 3 <i>Robert M. Tripp</i>	39:16	Atari 800 Player/Missile Graphics <i>Mike Dougherty</i>	44:9
It's Time to Stop Dreaming, Part 4 <i>Robert M. Tripp</i>	40:20	Using Atari's Countdown Timers <i>Mike Dougherty</i>	45:38
The Radio Shack Color Computer <i>John Steiner</i>	41:9		
The 6809 and the S-50 Bus <i>Dale Puckett</i>	41:68	SYM Epson MX80 Interface for SYM-1 <i>Richard H. Turpin</i>	44:45
Lunar Lander <i>John Steiner</i>	42:41		

ACRO

MICRObits

(Continued from page 87)

HTS OSI Products

Manual documents circuitry and provides software to implement Real-Time Clock. User-friendly system prompts and displays date/time. Add 4K CMOS RAM, 4K EPROM, and PORTS using designs documented in *General Expansion Board Construction Manual*. Each manual - \$15.

Hunter Technical Services
P.O. Box 359
Elm Grove, WI 53122

PET/CBM Operating Systems 3.0

Matric expands Commodore BASIC with 14 new commands for handling arrays. Algebraic-style syntax. Checks for conformability. Extended error messages. Machine language (5K) program lets you display a matrix on screen and change its values, transfer data between matrices or fill a matrix with a constant, transpose, transfer diagonals between matrices, eigenvalues and eigenvectors. Specify size and ROM set, tape or disk. 32-page manual. Price: \$125.00.

Cognitive Products
P.O. Box 1658
Bloomington, IN 47402

RTTY Without Hardware

Receive/Transmit RTTY with Egbert Disc Program. Apple generates and demodulates tones with screen display and other features. Apple II, 3.2/3.3, 48K, disk drive. State S.W.L. or HAM call. Send \$39.95 plus \$2.50 shipping (California residents add 6% tax).

W.H. Nail Co.
275 Lodgeview Drive
Oroville, CA 95965

COMPendium

COMPendium each month abstracts all articles in 20-plus microcomputer publications. Indispensable listings of software, hardware and book reviews. Find *all* information for your machine. Annual Index each December. Keeps track of published programs you half-remember. Sample - \$1.50; yearly - \$18.

Epicurious
P.O. Box 129
Lincolndale, NY 10540

Computer Covers

Keep dust and dirt out of your equipment with a computer cover. Cover for full Apple \$11.50, disk drive \$4.25, stack disk (2) \$6.25, monitor (state size) \$10.25. Send check plus \$1.50 postage.

M.P. Computer Services
P.O. Box 2396 Encinal Stations
Sunnyvale, CA 94087

For Sale

System utilities, terminal emulator software, data encryption systems for secure communications. For OSI Superboard, Apple, other 6502 systems.
D. Wolf, Ph.D.
Box 565
Port Hueneme, CA 93041

Target - An AIM 65 Newsletter

Need information for your AIM 65 computer? News, software, and hardware are examples of items covered in the newsletter. Yearly subscription rate is \$6.00 in the US and Canada, \$12.00 elsewhere. Back issues are available beginning with 1979 at the same per year rate.

Target
C/O Donald Clem
RR#2
Spencerville, OH 45887

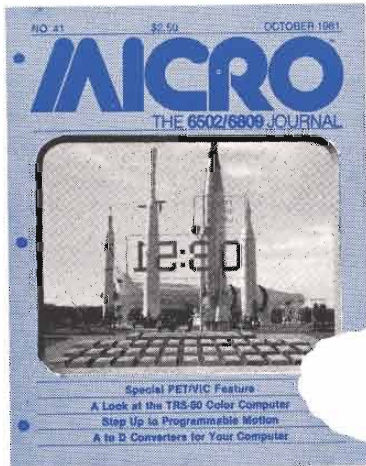
Joystick Interface

For PET, AIM 65, SYM, KIM or other 6502-based computer. Uses five VIA ports to give eight-bit conversion of up to eight resistance devices. Requires 64 bytes of memory [software included]. Assembled, tested — \$29.95; bare board - \$12.95.

Sydney S. Koegler
Micro-K Computer Products
2339 Carriage Ave.
Richland, WA 99352

ACRO

A feast of computing ideas.



You'll love every byte.

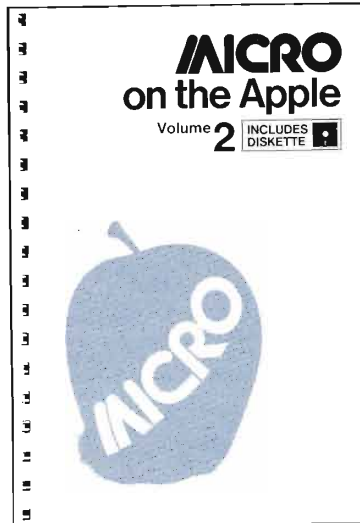
If you work with a 6502 or 6809 based system, you're probably hungry for the facts and ideas that will help you understand the inner workings of your computer. You want to go beyond canned software—use your computer for more than games—learn the advanced programming techniques that enable you to get the most out of your 6502/6809 system.

MICRO, The 6502/6809 Journal, gives you page after page, month after month, of solid information to sink your teeth into. **MICRO** is the premier how-to magazine for serious users of the Apple, PET/CBM, OSI, Atari, AIM, SYM, KIM, and all 6809 based systems including the TRS-80 Color Computer. It's a resource journal internationally respected by professionals in business, industry, and education.

SUBSCRIPTION RATES (U.S. dollars)
 Yearly subscription (ISSN 027-9002) saves 20% off the single-issue price. U.S., \$24 (SPECIAL OFFER: Save 30% off single-issue price: 2 years, \$42)
 Other countries, \$27 (via surface mail. Foreign air rates available on request.)

Get more out of your Apple . . .

with the **MICRO ON THE APPLE** series



VOLUME 2 just released!

More than 40 new programs on diskette to help you get more from your Apple:

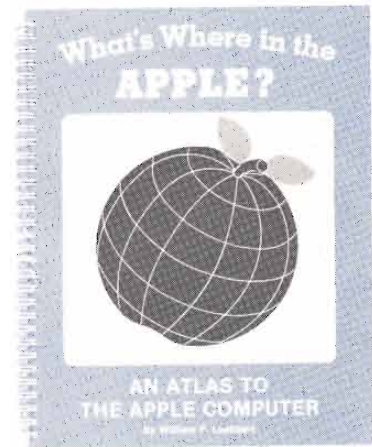
- Machine Language Aids
- I/O Enhancements
- Runtime Utilities
- Graphics and Games
- Hardware and Reference Information

31 choice articles
46 tested programs on diskette
 (13 sector DOS 3.2 format)

Volume 1 also available at \$24.95. Together **MICRO on the Apple 1 & 2** provide more than 70 programs on diskette for less than \$1.00 each. No need to type in hundreds of lines of code.

- ★ Look for all these **MICRO INK** publications at your local computer store, or
- ★ Call our **toll-free** number: **1-800-227-1617, ext. 564** (In California, 1-800-772-3545, ext. 564) and charge it to your VISA or MasterCard, or
- ★ Use the order form below. Send your check (payable to MICRO) and the form to: MICRO, Dept. OA, P.O. Box 6502, Chelmsford, MA 01824.

with the most important book ever published for the Apple



The most comprehensive description of Apple II firmware and hardware ever published—all in one place.

What's Where in the Apple?

- Guides you—with a numerical Atlas and an alphabetical Gazetteer—to over **2,000** memory locations of **PEEKs**, **POKEs** and **CALLs**.
 - Gives names and locations of various **Monitor**, **DOS**, **Integer BASIC**, and **Applesoft** routines—and tells you what they're used for.
 - Helps BASIC users to speed up their programs.
 - Enables assembly language programmers to simplify coding and interfacing.
- All Apple users will find this book helpful in understanding their machine, and essential for mastering it!

QTY	ITEM	PRICE EACH	COST	<input type="checkbox"/> Check enclosed	<input type="checkbox"/> Charge my credit card below
	MICRO on the Apple 2	@ \$24.95			
	MICRO on the Apple 1	@ \$24.95			
	What's Where in the Apple?	@ \$14.95			
	MICRO (U.S.) 1 yr @ \$24; 2 yrs @ \$42				
	MICRO (Foreign) 1 yr @ \$27				
	Subtotal				
	Massachusetts residents add 5% sales tax				
	Add \$2 per book for shipping				
	MICRO Journal excluded. Allow 4-6 weeks for delivery				
	TOTAL				

Name _____
 Company: _____
 Street _____
 City _____ State _____ ZIP _____
 Signature _____
 VISA MasterCard
 Card Number _____ Expiration Date _____

PET/CBM

PET — "Personal Electronic Transactor"
 CBM — "Commodore Business Machine"

6502-based computer, manufactured by Commodore Business Machines

PET models include graphic keyboard

CBM models include business keyboard

Available in 4K, 8K, 16K, 32K, and 96K configurations

All models, except 8000 series, include 25-row by 40-column screen. 8016, 8032, and 8096 have 25 by 80 screen.

Two 256-character sets — one for graphics, one (with lower case) for business

Memory expansion bus, parallel interface, IEEE-488 instrumentation bus standard

Reliable cassette operating system, powerful screen editing and character-programmable cursor control are characteristic of PET/CBM.

VIC and SuperPET have many features in common with PET/CBM.

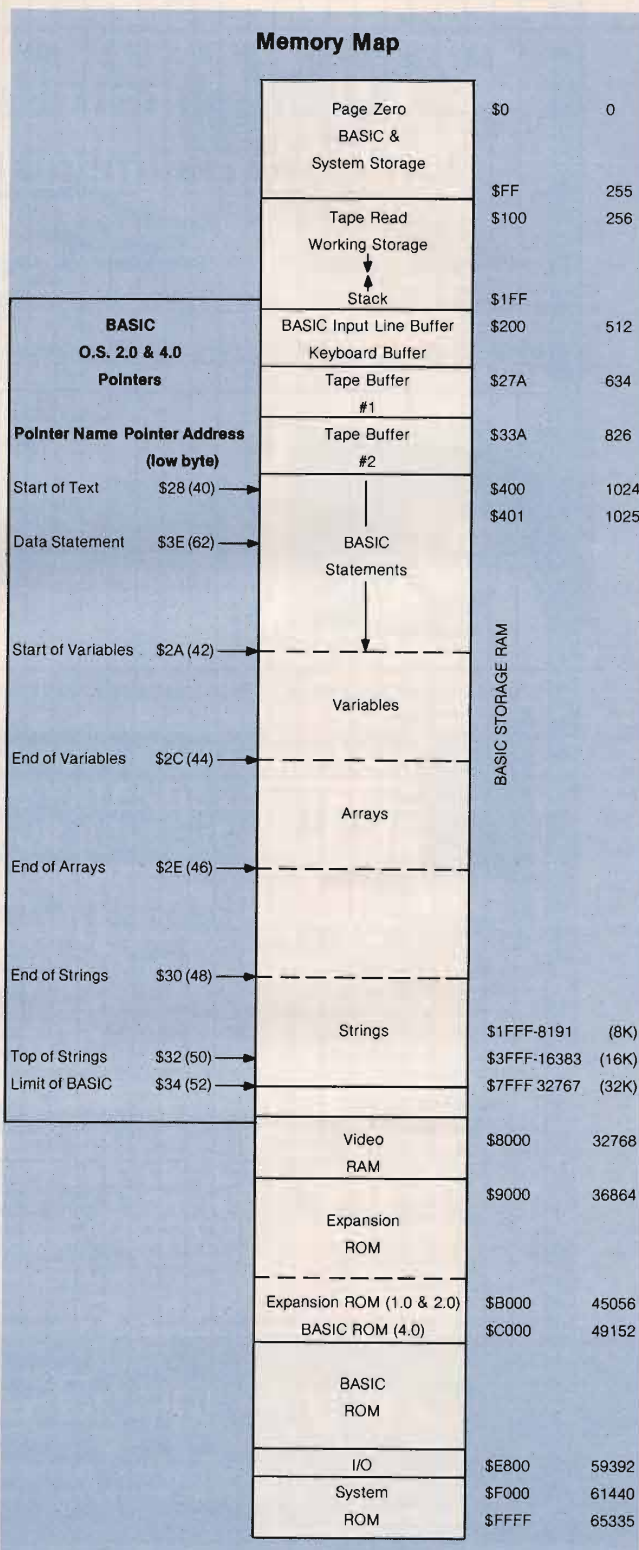
Parallel Port Connector

Upper Pin Identification Character	Signal Label	Lower Pin Identification Character	Signal Label
1	Ground	A	GND
2	T.V. Video	B	CA1
3	IEEE-SRQ	C	PA0
4	IEEE-EOI	D	PA1
5	Diagnostic Sense	E	PA2
6	Tape #1 READ	F	PA3
7	Tape #2 READ	H	PA4
8	Tape Write T.V.	J	PA5
9	Vertical T.V.	K	PA6
10	Horizontal T.V.	L	PA7
11	GND	M	CB2
12	GND	N	GND

IEEE-488 Connector

PET Edge-card Pin Numbers	Standard IEEE Connector Pin Numbers	IEEE Signal Mnemonic
Upper Pins		
1	1	DI01
2	2	DI02
3	3	DI03
4	4	DI04
5	5	EOI
6	6	DAV
7	7	NRFD
8	8	NDAC
9	9	IFC
10	10	SRQ
11	11	ATN
12	12	GND
Lower Pins		
A	13	DI05
B	14	DI06
C	15	DI07
D	16	DI08
E	17	REN
F	18	GND
H	19	GND
J	20	GND
K	21	GND
L	22	GND
M	23	GND
N	24	GND

Memory Map



MICRO Data Sheet #4

PET/CBM

Decimal	Hex	ASCII	Screen	BASIC	6502	Decimal	Hex	ASCII	Screen	BASIC	6502	Decimal	Hex	ASCII	Screen	BASIC	6502
0	00		@	end line	BRK ORA(L,X)	64	40	@				192	C0			TAN	CPY #
1	01	A	A			65	41	A				193	C1	a		AIN	CMPI(Y)
2	02	B	B			66	42	B				194	C2	b		PEEK	
3	03	C	C			67	43	C				195	C3	c		LEAK	
4	04	D	D			68	44	D				196	C4	d		VIA	CPY Z
5	05	E	E	white	ORA Z	69	45	E				197	C5	e		VIA	CMPI Z
6	06	F	F		ASL Z	70	46	F				198	C6	f		ASC	DEC Z
7	07			bell		71	47					199	C7	g		CHR\$	INY
8	08	lock				72	48					200	C8			LEFT\$	CMPI
9	09	unlock				73	49					201	C9			RIGHT\$	DEX
10	0A					74	4A					202	CA			MID\$	
11	0B					75	4B					203	CB			CONCAT	CPY
12	0C					76	4C					204	CC			OPEN	CMPI
13	0D	carr. ret.			ORA	77	4D					205	CD			DOOPEN	CMPI
14	0E	top-left			ASL	78	4E					206	CE			DOCLOSE	DEC
15	0F					79	4F					207	CF			RECORD	
16	10	cur. down			BRL	80	50					208	D0			HEADER	BNE
17	11	cur. up			ORA(W,Y)	81	51					209	D1			DOSELECT	CMPI(Y)
18	12	cur. home				82	52					210	D2			DOCOPY	
19	13	delete				83	53					211	D3			APPEND	CMPI Z
20	14	del. line			ORA Z,X	84	54					212	D4			DSAVE	DEC Z,X
21	15	erase begin			ASL Z,X	85	55					213	D5			CATALOG	
22	16					86	56					214	D6			RECALL	CMPI Y
23	17					87	57					215	D7			SEARCH	
24	18				CIC	88	58					216	D8			DIRECTORY	
25	19				ORA Y	89	59					217	D9				
26	1A					90	5A					218	DA				
27	1B	red				91	5B					219	DB				
28	1C	cur. right			ORA X	92	5C					220	DC				CMPI X
29	1D	green			ASL X	93	5D					221	DD				DEC X
30	1E	space				94	5E					222	DE				
31	1F	space				95	5F					223	DF				CPY #
32	20	space			JSR	96	60					224	E0				SBC(I,Y)
33	21				AND(L,X)	97	61					225	E1				
34	22					98	62					226	E2				
35	23					99	63					227	E3				
36	24				BIT Z	100	64					228	E4				CPY Z
37	25				AND Z	101	65					229	E5				SBC Z
38	26				ROL Z	102	66					230	E6				INC Z
39	27					103	67					231	E7				
40	28				PLP	104	68					232	E8				INX
41	29				AND #	105	69					233	E9				SBC #
42	2A				ROL A	106	6A					234	EA				NOP
43	2B				BIT	107	6B					235	EB				CPY
44	2C				AND	108	6C					236	EC				SBC
45	2D				ROL	109	6D					237	ED				INC
46	2E					110	6E					238	EE				INC
47	2F					111	6F					239	EF				BEQ
48	30				BMI	112	70					240	F0				SBC(I,Y)
49	31				AND(I,Y)	113	71					241	F1				
50	32					114	72					242	F2				
51	33					115	73					243	F3				SBC Z,X
52	34				AND Z,X	116	74					244	F4				INC Z,X
53	35				ROL Z,X	117	75					245	F5				
54	36					118	76					246	F6				SED
55	37					119	77					247	F7				SBC Y
56	38				SEC	120	78					248	F8				
57	39				AND Y	121	79					249	F9				
58	3A				CLI	122	7A					250	FA				SBC Y
59	3B					123	7B					251	FB				
60	3C					124	7C					252	FC				SBC X
61	3D				AND X	125	7D					253	FD				INC X
62	3E				ROL X	126	7E					254	FE				
63	3F					127	7F					255	FF				

VersaWriter & APPLE II: The Keys to Unlimited Graphics

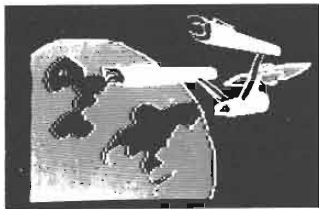
DRAWING TABLET

Although VersaWriter operates on a simple principle, it produces graphics which match or exceed those of other digitizers. Rugged construction, translucent base, easy to use — plugs directly into APPLE II.



GRAPHICS SOFTWARE

Easily the most capable and complete graphics software for the home computer available. Fast fill drawings in 100 colors. All text in five sizes, compile and display shapes, edit, move and much more!



UNIQUE OFFER

See VersaWriter at your local dealer and pick up a copy of our demonstration disk. The complete VersaWriter hardware and software package is a real bargain at \$299. For more information call or write:



EZ Port Will Solve Your Game I/O Problem!

How many times have you gone through the hassle of changing from game paddles to joystick, VersaWriter, or any other device using the game I/O? First, you have to remove whatever is sitting on top of the Apple—a video terminal, disk drives, printer, etc.

Next you remove the computer cover and try to see what you're doing as you switch plugs to the I/O. Then you replace the computer cover and whatever was on top of the Apple.

After all this, you find that you can't run the program because the I/O device is plugged in backwards or is 'off by a pin'.

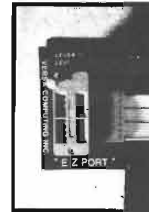
Sound familiar?

EZ PORT GAME I/O EXTENDER FOR APPLE II

WHAT IS EZ PORT?

EZ Port is a specially designed extension unit for the Apple game I/O port. It's a board with a socket and a two foot long cable which plugs into the internal I/O port. You attach EZ Port wherever you prefer on the outside—on the side, the back, or on top.

EZ Port has a ZIP DIP II socket (ZIP=zero insertion force). These sockets are meant to be plugged into many times and will not wear out like ordinary sockets. All you do is plug in the appropriate device (joystick, paddles, etc.) and flip the switch to the ON position. No pressure is exerted on the 16-pin plug until you switch, so all the connectors will last longer, too!



EZ PORT MAKES GAME I/O CHANGES CONVENIENT, QUICK & SAFE.

Suggested price **\$24.95**
Ask your local computer retailer for EZ Port, or contact:

VersaComputing, Inc.
3541 Old Conejo Rd. Suite 104
Newbury Park, CA 91320
(805) 498-1956

Dealers inquiries welcome.

Apple and Apple II are registered trademarks of Apple Computer, Inc.

ZIP DIP is a trademark of Textool Products, Inc.



Versa Computing, Inc.

3541 Old Conejo Road, Suite 104
Newbury Park, CA. 91320 (805)498-1956

PERRY PERIPHERALS REPAIRS KIMs!! (SYM_s AND AIM_s TOO)

- We will Diagnose, Repair, and Completely Test your Single Board Computer
- We Socket all replaced Integrated Circuits
- You receive a 30-day Parts and Labor Warranty
- Labor is \$38.00 if 40-pin ICs are socketed (\$40.00 otherwise) — Parts Extra
- Your repaired S.B.C. returned via U.P.S. — C.O.D., Cash

Don't delay! Send us your S.B.C. for repair today
Ship To: (Preferably via U.P.S.)

PERRY PERIPHERALS
6 Brookhaven Drive
Rocky Point, NY 11778

Perry Peripherals carries a full line of the acclaimed HDE expansion components for your KIM, SYM, and AIM, including RAM boards, Disk Systems and Software like HDE Disk BASIC V1.1. Yes, we also have diskettes. For more information, write to: P.O. Box 924, Miller Place, NY 11764 or Phone (516) 744-6462.



EXTEND THE POWER OF APPLESOFT WITH FOUR NEW & POWERFUL COMMANDS

WITH THE SORT AND FORMAT ROMS IN YOUR MOUNTAIN COMPUTER'S ROMPLUS* OR ANDROMEDA'S ROMBOARD* YOU WILL MAKE APPLESOFT* BASIC ONE STEP MORE POWERFUL THAN THE MOST ADVANCED LANGUAGE AVAILABLE. AND - BECAUSE READ ONLY MEMORY NEVER FORGETS, THESE NEW COMMANDS BECOME A PERMANENT PART OF APPLESOFT'S* BASIC LANGUAGE.

**SORT
ROM™**

ARRAY SORTING - Sort any single dimensioned (or multi-dimensioned-indirectly) integer, real, or string array. Unparalleled sorting speeds are accomplished by a unique machine code and does not use any scratch arrays or additional memory. A typical sort time for a 1000 item array is 3.5 seconds. Sort options include: create a record array, reverse sort, subrange sort, and a sort activity indicator.

ARRAY DELETION - Unused arrays, which can consume large amounts of memory space, can be deleted to free up memory or allow a previously dimensioned array to be redimensioned **\$39.95**

**FORMAT
ROM™**

WORD PROCESSING - A powerful and versatile system which will properly format your program's output on your monitor or printer. You get all the commands of a dedicated word processor plus the power of Applesoft's commands. Define or change printing margins, page width, page length, text centering, right and left justification, indenting and outdenting, etc., with simple basic commands. By combining the commands of the format ROM and Applesoft you can print more than one copy with a for-next loop or insert different information with input commands or take it from your existing arrays.

PRINT...USING will tabulate, right justify, line up all decimal points, pad the right and left side of Alpha/Numerical data with any predefined character, insert commas, and can be used within formulas or equations which will then format the mathematical result **\$49.95**

FORMAT ROM AND SORT ROM will support all printers. 80 column boards, lower case adapters, and requires 48K, FP in ROM, DOS 3.2 or 3.3, M.C.'s ROMPLUS* or Andromeda's ROMboard* **SAVE MONEY BY ORDERING THE FORMAT ROM AND SORT ROM AS A SET \$84.95**

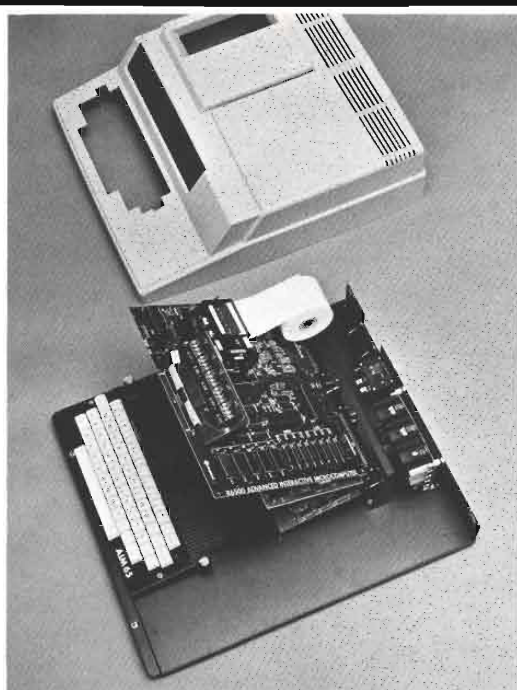
OTHER ROMS AVAILABLE: All ROMS are compatible with MC's Romplus or Andromeda's ROM Board.

- DUAL DOS ROMS - Switch from one DOS (3.2 or 3.3) to the other without booting	\$54.95
- FP RENUMBER/MERGE ROM - Apple Computer's infamous renumber program	\$39.95
- BASICS ROM - Will boot standard, special, and dedicated 13 sector disks	\$39.95
- FP EDITROM - Global search, change, and remove. (Works jointly with PLE)	\$39.95
- COMMAND ROM - Catalog Command Menu and Disk Map	\$39.95
- DISK COPY/SPACE ROM - Duplicates 13 or 16 Sector Disks	\$39.95
- FP UTILITY ROM - Auto Line Numbering, LIST CTRL, RESURRECT KILLED PROG, Alphabetize DISK CAT., Expung DOS	\$39.95

(Check or M.O.) Visa or MasterCard Accepted

SOFT CTRL SYSTEMS, BOX 599, WEST MILFORD, NJ 07480 201-728-8750

*REGISTERED TRADEMARK
ALL FIRMWARE IS COPYRIGHTED



AIM HIGH

Let Unique Data Systems help you raise your sights on AIM 65 applications with our versatile family of AIM support products.

- Go for high quality with our ACE-100 Enclosure. It accommodates the AIM 65 perfectly, without modification, and features easy access two board add-on space, plus a 3" x 5" x 17" and a 4" x 5" x 15.5" area for power supplies and other components. \$186.00.
 - Get high capability with Unique Data System's add-on boards. The UDS-100 Series Memory-I/O boards add up to 16K bytes of RAM memory or up to 48K bytes ROM/PROM/EPROM to your Rockwell AIM 65. You also get 20 independently programmable parallel I/O lines with an additional user-dedicated 6522 VIA, two independent RS-232 channels with 16 switch-selectable baud rates (50 to 19.2K baud), and a large on-board prototyping area. Prices start at \$259.00.
 - If you need to protect against RAM data loss, the UDS-100B offers an on-board battery and charger/switchover circuit. \$296.00.
 - Heighten your AIM 65's communications range by adding the UDS-200 Modem board. It features full compatibility with Bell System 103 type modems and can be plugged directly into a home telephone jack via a permissive mode DAA. No need for a data jack or acoustic coupler. The UDS-200 also has software-selectable Autoanswer and Autodial capability with dial tone detector. The modem interfaces via the AIM 65 expansion bus, with the on-board UART and baud rate generator eliminating the need for an RS-232 channel. \$278.00.
 - The UDS-300 Wire Wrap board accepts all .300/600/900 IC sockets from 8 to 64 pins. Its features include an intermeshed power distribution system and dual 44-pin card edge connectors for bus and I/O signal connections. \$45.00.
 - Get high performance with the ACE-100-07 compact 4" x 5" x 1.7" switching power supply, delivering +5V @ 6A, +12V @ 1A, and +24V for the AIM printer. \$118.00.
- Installation kits and other related accessories are also available to implement your AIM expansion plans. Custom hardware design, programming, and assembled systems are also available. High quality, high capability, high performance, with high reliability... all from Unique Data Systems. Call or write for additional information.

Unique Data Systems Inc.
1600 Miraloma Avenue, Placentia, CA 92670
(714) 630-1430

Advertiser's Index

Aardvark Technical Services, Ltd.	35
Abacus Software	91
Adventure International	94-95
Andromeda, Inc.	98
Anthro-Digital Software	69
Apple Tree Electronics	22
Ark Computing	101
Arrow Data Systems	20
Astar International Co.	7
Aurora Software Associates	59
Comp-U-Gamer	59
CompuTech	16
Computer Case Co.	59
The Computerist, Inc.	IBC
Computer Mail Order	24
Computer Mat	60
Computer Science Engineering	59
Computer Systems Design	72
Connecticut Information Systems Co.	47
Creative Computing	83
Datamost	93, 96
Data Transforms, Inc.	87
Decision Systems	50
D&N Micro Products, Inc.	72
Dr. Dobbs Journal	88
Eastern House Software	26
ESD Labs Co., Ltd.	23
Excert, Inc.	28
Execom Corp.	60
Genesis Information Systems Inc.	86
Gimix, Inc.	1
GOSUB International, Inc.	75
Hogg Laboratory Inc.	59
Hudson Digital Electronics Inc.	12
Huntington Computing	BC
Interesting Software	65
Keystone Data Consultants	17
MICRObits (Classifieds)	21, 87, 105
Micro Business World Inc.	56
MICRO INK, Inc.	90, 106
Microsoft Consumer Products	IFC
Micro Ware Distributing Inc.	91
Modular Systems	60
Nibble	73
Nikrom Technical Products	11
Olympic Sales Co.	14
Orion Software	32
Passage Research	23
Peek (65)	75
Perry Peripherals	110
Pretzelland Software	91
Progressive Computing	66
RC Electronics Inc.	45
Sensible Software	2, 31
SGC	46
S&H Software	30
Skyles Electric Works	60
Smartware	60
Smoke Signal Broadcasting	4
Softape Inc.	18
Soft CTRL	110
Softside	36, 97
Southwestern Data Systems	8
Stellation Two	66
Sublogic Communications Corp.	55, 102
Talbot Microsystems	53
Unique Data	111
Versa Computing, Inc.	27, 109
Voicetek	54
Word's Worth	23

MICRO INK, Inc is not responsible for claims made by its advertisers. Any complaint should be submitted directly to the advertiser. Please also send written notification to MICRO.

Next Month in MICRO

June Applications Feature

- **A Low-Cost Digitizer for the Apple II** — This article tells you how to use a sheet of half-reflecting plastic to build a digitizer for your high-resolution graphics computer.
- **AIM Logic Trainer** — Test your logic circuits with this program. It allows you to control the inputs and send the outputs to LEDs or printer.
- **Face Synthesizer** — This program creates an animated face on a PET screen that changes expression on keyboard command. Animation can also be controlled from BASIC programs, so the PET face can be used in any application — education, marketing, games, etc.
- **Program for Inverting a Matrix** — Include this short BASIC routine in your mathematical and statistical programs.

Plus...

*AIM User Function Dispatcher
Additional Output Ports on the KIM
Tiny Pilot for the PET
Disk to Tape Backup Utility for the Apple
Sequential File Handler for C1P*

1982 Editorial Schedule

July — Apple
August — Programming Techniques
September — 68000
October — Education
November — Atari/Games
December — PET

20% OFF

Your money goes farther when you subscribe. During the course of a year, when you subscribe, you save 20% (in the U.S.).

Pay only \$24.00 (\$2.00 a copy) for 12 monthly issues of MICRO sent directly to your home or office in the U.S.

More MICRO for Less Money When You Subscribe

But on the newsstand — if you can locate the issue you want — you pay \$30.00 a year (\$2.50 a copy).

Special Offer — Subscribe for 2 years (\$42.00) and get 30% off the single issue price.

Subscribe to MICRO today.

MICRO
34 Chelmsford Street
P.O. Box 6502
Chelmsford, MA 01824

Please send me MICRO for 1 year 2 years
NOTE: Airmail subscriptions accepted for 1 year only.

Check enclosed \$ _____
Charge my VISA account
 Mastercard account

No. _____

Expiration date _____

Name _____

Address _____

City/State _____ Zip _____

Subscription Rates Effective January 1, 1982

Country	Rate
United States	\$24.00 1 yr. 42.00 2 yr.
Foreign surface mail	27.00
Europe (air)	42.00
Mexico, Central America, Mid East, N. & C. Africa	48.00
South Am., S. Afr., Far East, Australasia, New Zealand	72.00

* Airmail subscriptions accepted for only 1 year.
For U.S. and Canadian 2-year rates, multiply by 2.

Job Title: _____

Type of Business/Industry: _____

WE MAY HAVE ALREADY BUILT THE BOARD YOU ARE STARTING TO DESIGN!

While you may not have heard of us before, you certainly know our customers: Fortune 500 companies, Universities and Government Agencies. Since 1976 we have been providing high quality microcomputer products, ranging from expansion boards, to stand-alone controllers, to complete systems. Before you start your next project, consider how easy it might be to use some of our products.

FLEXI PLUS

A multi-function controller which handles:

- 8" and mini diskettes, double-sided, double-density
- RS-232 Communications with programmable features
- IEEE-488 Instrumentation Bus fully implemented
- 6809E Microprocessor, up to 56K RAM, ROM, EPROM
- Parallel/Serial I/O, Cassettes and TTY Interfaces

This versatile controller may be used as an expansion board for any 6502 or 6809 system; as a stand-alone controller; or, as the basis of a complete microcomputer system.

MICRO PLUS

A video-oriented controller which includes:

- Programmable display up to 132 columns by 30 rows
- Programmable character sets in EPROM and RAM
- Character and Bit-Mapped Graphics
- ASCII Keyboard and Light Pen Interfaces
- RS-232 Programmable Communications Interface
- 6502 Microprocessor, up to 7K RAM, 2K EPROM

This video controller may be used to expand almost any 6502- or 6809-based system; or as a stand-alone intelligent terminal; or, as the basis of a complete 6502-based computer system.

DRAM PLUS

A multi-purpose expansion board which features:

- Up to 40K RAM memory with a memory manager
- Up to 16K ROM or EPROM memory
- EPROM Programmer for 2516, 2716, 2532 and 2732
- Multiple parallel/serial I/O ports and timers/counters
- Prototyping area for custom circuits

This memory-oriented expansion board permits addressing of memory on 4K boundaries, supports swapping of sections of memory, and works with most 6502- or 6809-based systems.

FOCUS

An Industrial quality system which features:

- Two mini disk drives, double-sided, double-density for over 640K bytes of on-line storage
- Commercial quality keyboard with numeric pad
- Upper/lower case ASCII with programmable characters and display formats, plus bit-mapped graphics
- High-resolution video monitor with green phosphor
- Heavy-duty aluminum case for desk or rack mounting
- RS-232 Communications built in; second optional
- IEEE-488 may be added to existing boards
- Includes 48K RAM, 4K EPROM, 6809 microprocessor

This extremely versatile system may be used for system development, for developing stand-alone products, as an end-user system, as an in-house business computer, as a word processor, and much more. Software is available from a number of suppliers and includes compiled BASIC, PASCAL, FORTH and many complete application packages.



\$3495.00

If you have a requirement which involves 6502- or 6809-based products, join the growing number of OEMs and System Integration Houses who look to us first. For additional information and our current product literature, please contact us at 617/256-3649 or TELEX 955318 INTL DIV.

**THE
COMPUTERIST®**

34 Chelmsford Street
Chelmsford, MA 01824

HUNTINGTON COMPUTING

ONE OF THE WORLD'S LARGEST INVENTORIES!

Great Grandma
Huntington said:

"Always buy your computer
media at a discount."
...And boy, do we ever let you
do that!



DISKS for APPLE[®]

Elephant	\$24.99 box
Maxell	\$36.99 box
Verbatim Datalife ...	\$28.99 box
Dysan	\$49.99 box

Software for Apple[®]

Bug Attack	\$26.29	The Home Accountant	\$63.69	Roach Hotel	\$30.69
Castle of Darkness	\$31.39	LISA	\$67.89	Soft 70	\$44.99
David's Midnight Magic	\$30.69	Neutrons	\$26.29	Tax Preparer '82	\$131.99
The Dictionary	\$87.89	Photar	\$26.89	Time Zone	\$87.89
The Game Show	\$34.29	Star Blazer	\$25.39	Torax	\$21.89
The General Manager	\$127.39	Suicide	\$26.29	Utility City	\$25.89
Hodgepodge	\$17.49	Raster Blaster	\$25.39	Volcanoes	\$43.49

NEW

Understand Yourself

Six great programs to test yourself. Packed with information. An outstanding buy. (Apple[®]) on disk.

\$24.99

Apple[®] Pie

Word Processor	\$116.89
12" Green Monitor	\$199.00
Lower Case Adaptor 2 Revision 7	\$29.69
VisiCalc	\$199.00
Z-Card (ALS)	\$239.00

NEW

Computer Almanac

- Biorhythms
- Loan Amortization
- Weather Predictor
- Calorie Counter
- Much more on disk for Apple[®]

\$24.99

We Also Carry PET[®], Atari[®], and TRS-80[®]

Call Toll-Free 800-344-5106 (outside California)

HUNTINGTON COMPUTING

Post Office Box 1297
Corcoran, California 93212

Foreign Orders 209-992-4481
In California 800-692-4146

Apple[®] is a registered trademark of Apple Computer, Inc.
Pet[®] is a registered trademark of Commodore.
TRS-80[®] is a registered trademark of Tandy Corp.
Atari[®] is a registered trademark of Atari, Inc.

Outside Calif. 800-344-5106

We take MasterCard, American Express or VISA (include card # and expiration date). California residents add 6% tax. Include \$2.00 for postage. Foreign and hardware extra. Foreign: remit U.S. currency, use listed charge cards, or make direct wire transfers through Security Pacific Bank, Corcoran, for a \$6.00 charge. All overseas orders shipped by air. Send for free catalog. Prices subject to change without notice.